



DeMa: dual-path delay-aware Mamba for efficient multivariate time series analysis

Rui AN^{1,2*}, Haohao QU^{2*}, Wenqi FAN²✉, Xuequn SHANG¹✉, Qing LI²

1. School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China

2. Department of Computing, The Hong Kong Polytechnic University, Hong Kong 999077, China

Received December 31, 2025; accepted April 22, 2026

E-mail: wenqifan03@gmail.com; shang@nwpu.edu.cn. * These authors contributed equally to this work.

© The Author(s) 2026. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract

Accurate and efficient multivariate time series (MTS) analysis is increasingly critical for a wide range of intelligent applications, including traffic forecasting, anomaly detection for industrial maintenance, and trajectory classification for health monitoring. Within this realm, Transformers have emerged as the predominant architecture due to their strong ability to capture pairwise dependencies. However, Transformer-based models suffer from quadratic computational complexity and high memory overhead, limiting their scalability and practical deployment for long-term, large-scale MTS modeling. Recently, Mamba has emerged as a promising linear-time alternative with high expressiveness. Nevertheless, directly applying vanilla Mamba to MTS remains suboptimal due to three key limitations: (i) the lack of explicit cross-variate modeling, (ii) difficulty in disentangling the entangled intra-series temporal dynamics and inter-series interactions, and (iii) insufficient modeling of latent time-lag interaction effects. These issues constrain its effectiveness across diverse MTS tasks. To address these challenges, we propose **DeMa**, a dual-path **Delay-Aware Mamba** backbone for efficient and effective MTS analysis. DeMa preserves Mamba's linear-complexity advantage while substantially improving its suitability for multivariate settings. Specifically, DeMa introduces three key innovations: (i) it decomposes the MTS context into intra-series temporal dynamics and inter-series interactions and learns them via two dedicated paths; (ii) it develops a temporal path with a *Mamba-SSD* module to capture long-range dynamics within each series, accommodating variable-length inputs and enabling series-independent, parallel computation while maintaining linear complexity; and (iii) it designs a variate path with a *Mamba-DALA* module that integrates delay-aware linear attention to model cross-variate dependencies, enhancing fine-grained, delay-sensitive dependency learning. Extensive experiments on five representative tasks, long- and short-term forecasting, data imputation, anomaly detection, and series classification, demonstrate that DeMa achieves state-of-the-art performance while delivering remarkable computational efficiency.

Keywords

time series analysis; state space model; mamba; attention mechanism

1 Introduction

Multivariate time series (MTS) are a crucial and fundamental data modality in both industrial and daily scenarios. They are collected at scale by physical and virtual sensors, which continuously record system measurements and encapsulate valuable information about the evolving dynamics of real-world systems [1,2]. Consequently, time series analysis serves as a cornerstone for understanding and predicting the behaviors of complex systems, enabling a wide range of intelligent applications, such as traffic flow forecasting for transportation scheduling [3,4], missing data imputation for web stream processing [5], anomaly detection for maintenance [6,7], and trajectory classification for health monitoring [8].

Due to the complex and non-stationary nature of real-world systems, observed MTS often exhibit intricate and entangled

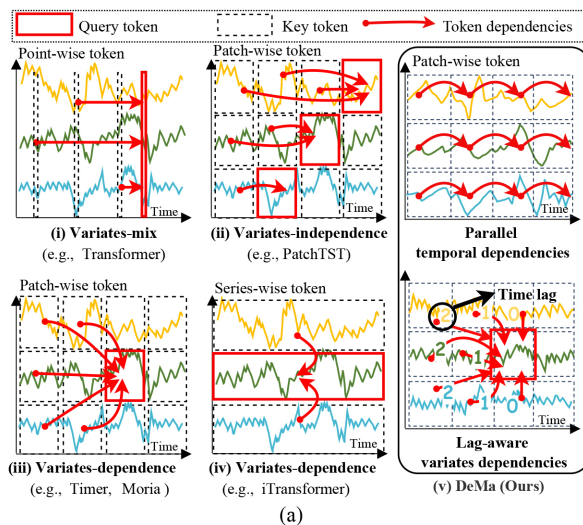
patterns. Along the temporal axis, multiple variations, including long-term trends, seasonal cycles, and irregular fluctuations, are typically mixed and overlapped. Along the variate axis, correlated series interact through latent dependencies, where the evolution of one variable can influence, or be influenced by, others with time delays. Accordingly, a series of deep models have been proposed to capture *temporal dependencies* and *variate dependencies*. From the perspective of backbone architectures, existing methods can be broadly categorized into RNN-, CNN-, MLP-, and Transformer-based methods [9–14]. Typically, RNN-based models [15–17] leverage recurrent structures to model temporal state transitions, while CNN-based models [18–20] employ temporal convolutional networks (TCNs) to extract local variation patterns. However, both RNN- and CNN-based methods are constrained by limited effective

receptive fields, which hinders their ability to capture long-term dependencies. Lightweight MLP-based models [21,22] utilize stacked fully connected layers to model temporal patterns, where dense connections implicitly capture measurement-free relationships among time points within each variable [13]. These linear forecasters are primarily designed for forecasting tasks and are computationally efficient. Nevertheless, due to their architectural simplicity and limited representational capacity, MLP-based models suffer from an information bottleneck, making it difficult to represent long-range and complex temporal dependencies. Transformer-based models, which adopt attention mechanisms to capture pairwise dependencies with a global receptive field, have become mainstream for MTS modeling [10,13,23–27]. These methods typically employ different tokenization strategies [28–31], such as point-wise tokens [23–26], patch-wise tokens [10,27], and series-wise tokens [13], and then apply attention to model temporal dependencies, variate dependencies, or both, as illustrated in Fig. 1(a)(i–iv). Despite their remarkable performance, Transformer-based models face growing concerns about their *quadratic computational complexity* [21,32], leading to substantial computational demands and memory overhead. Specifically, the cost of self-attention scales quadratically with the token length, as listed in Fig. 1(b)(i–iv). For an MTS instance with N variates, each represented by L tokens embedded in D dimensions, the complexity of fully self-attention models can reach unacceptably $O(N^2L^2D)$ [11,12]. As both N and L grow, quadratic scaling becomes a major bottleneck for long-term, large-scale MTS modeling, motivating the development of more computationally efficient architectures for MTS analysis.

Recently, **Mamba** has emerged as a promising backbone for capturing complex dependencies in sequential data [33–35]. By leveraging selective mechanisms and hardware-aware computational designs [34], Mamba attains modeling performance comparable to

Transformers while maintaining linear complexity with respect to sequence length. Furthermore, Mamba-2 [34] further improves efficiency by exploiting the structured State Space Duality (SSD) property to reformulate computations into highly parallelizable matrix operations. With *high expressiveness* enabled by the selective mechanism, *efficient training and inference* supported by parallel computation, and *linear scalability* with respect to context length, Mamba offers a compelling alternative to attention-based architectures for sequential modeling. Consequently, an increasing number of Mamba-based models have been proposed across various domains, such as Jamba [36] in natural language processing, Vision Mamba [37] in computer vision, Caduceus [38] in genomics, and SSD4Rec [39] for recommender systems.

Despite the recent success of Mamba-style techniques in enabling efficient MTS analysis, directly applying standard Mamba to multivariate time series often yields suboptimal performance compared with state-of-the-art methods [40]. This gap can be mainly attributed to the following limitations. (1) **Difficulty in disentangling entangled temporal and variate contexts:** Unlike language tokens, which typically lie in relatively discrete contexts, MTS observations are jointly governed by intra-series temporal dynamics and inter-series interactions. These factors often overlap and intertwine, making it difficult for vanilla Mamba to disentangle meaningful and structured representation. (2) **Lack of explicit cross-variate modeling:** Mamba was originally developed for language-like, essentially univariate sequences, and therefore lacks explicit mechanisms for capturing dependencies among multiple correlated variates. Prior studies have shown that modeling inter-variable interactions is crucial for effective MTS representation learning [13,27,41]. (3) **Insufficient explicit modeling of lag dependencies:** Mamba relies on recurrent hidden-state updates, in which each state depends primarily on the immediately preceding timestep. This



Category	Representative Model	Temporal Dependencies	Variate Dependencies	Computational Complexity
(i)	Transformer [9]	✓	✗	$O(T^2D)$
(ii)	PatchTST [10] Timer [11]	✓ ✓	✗ ✗	$O(NL^2D)$ $O(NL^2D)$
(iii)	Moirai [12]	✓	✓	$O(N^2L^2D)$
(iv)	iTransformer [13]	✓	✓	$O(N^2D)$
(v)	DeMa (Ours)	✓	✓	$O(2NLD^2)$

Fig. 1 Dependency modeling paradigms and complexity comparison of representative multivariate time-series architectures. (a) Illustration of tokenization schemes and dependency modeling strategies, including variate-mixing, variate-independence, and variate-dependence designs. (b) Computational complexity comparison of representative models. Here, T , N , L , and D denote the lookback length, number of variates, token length, and embedding dimension, respectively. In typical long-horizon settings, $T > L \gg N, D$. **DeMa** decouples temporal modeling from delay-aware cross-variate interactions, achieving linear-time complexity

design may limit its ability to explicitly capture lagged effects that are pervasive in real-world MTS, where changes in one variate may influence another only after a non-negligible delay [3,41–43]. For example, in traffic forecasting, an accident in one region may not immediately affect adjacent areas; instead, its impact often propagates through the network over several minutes. Precisely capturing such propagation delays is therefore crucial for accurate dependency modeling and reliable prediction.

Motivated by the above limitations and the urgent practical demands of long-term, large-scale MTS analysis, we aim to develop a Mamba-based time-series representation backbone that simultaneously captures intricate temporal dependencies and delay-aware cross-variate interactions while preserving linear-time efficiency. Toward this goal, we explicitly decompose the time-series context into (i) *intra-series temporal dynamics*, which can be learned independently and in parallel for each variate, and (ii) *inter-series interactions*, which are modeled in a delay-aware manner to capture cross-variate dependencies, as illustrated in Fig. 1(a)(v).

To this end, we propose a dual-path Delay-aware Mamba for efficient multivariate time series analysis, namely **DeMa**. Specifically, DeMa first adaptively selects task-relevant spectra via an Adaptive Fourier Filter and decomposes the input into a *Cross-Time Component* and a *Cross-Variate Component*. These two components are then fed into stacked **DuoMNet** blocks. Each DuoMNet block contains two parallel paths, each coupled with a scan operator and a lightweight Mamba-based module. Concretely, the *Cross-Time Scan* serializes each variate along the temporal axis into a 1D sequence, which is processed by *Mamba-SSD*. This design supports MTS-independent parallel computation across variates and efficiently captures long-range temporal dependencies within each series, with computation scaling linearly with the series length. In parallel, the *Cross-Variate Scan* reorganizes tokens to emphasize inter-variable interactions at each token step. The scanned sequence is processed by *Mamba-DALA*, which integrates Delay-Aware Linear Attention (DALA) to explicitly model cross-variate dependencies with both a global correlation delay and a token-level relative delay. As a result, DeMa achieves delay-aware variate interaction modeling while preserving linear-time computation. Finally, we fuse the temporal-path and variate-path representations through a weighted fusion layer and project to task-specific outputs via lightweight heads. To sum up, our major contributions include:

- We propose **DeMa**, a dual-path Mamba-based backbone for general MTS analysis that jointly captures intra-series temporal dynamics via a temporal path and delay-aware cross-variate dependencies via a variate path. Benefiting from a linear-time state-space backbone, DeMa achieves a favorable trade-off between accuracy and efficiency for long-term and large-scale MTS modeling.
- We introduce Mamba-SSD to model intra-series temporal dependencies within each variate. It accommodates variable-length inputs and enables series-independent, parallel modeling via block-based matrix multiplication, thereby improving scalability and throughput.
- We design Mamba-DALA, which integrates Delay-Aware Linear Attention to explicitly model cross-variate interactions with both global correlation delays and token-level relative delays, thereby enhancing fine-grained, delay-sensitive dependency learning.
- We conduct extensive experiments on five mainstream tasks, including long- and short-term forecasting, imputation, classification, and anomaly detection. The results demonstrate that DeMa achieves consistently strong performance across tasks while significantly reducing training time and GPU memory usage, suggesting its practicality and scalability for real-world large-scale MTS analysis.

The remainder of this paper is structured as follows. Section 2 introduces the preliminaries. Section 3 introduces the proposed model, which is evaluated and discussed in Section 4. Then, Section 5 summarizes the recent development of time series analysis. Finally, conclusions are drawn in Section 6.

■ 2 Preliminaries

■ 2.1 Notations and definitions

We denote a multivariate time series (MTS) within a lookback window as $\mathcal{X} \in \mathbb{R}^{N \times T}$, where N denotes the number of variates and T denotes the number of time steps. The i th variate is represented as $\mathcal{X}_{i,:} = \{x_{i1}, x_{i2}, \dots, x_{iT}\} \in \mathbb{R}^T$, while the multivariate observation at time step t is denoted by $\mathcal{X}_{:,t} = \{x_{1t}, x_{2t}, \dots, x_{Nt}\} \in \mathbb{R}^N$.

The primary objective of this work is to learn a task-agnostic representation $\mathbf{Z} = \mathcal{F}_{\Theta}(\mathcal{X})$, which can be adapted to diverse downstream tasks through task-specific heads. These tasks include point-level tasks, such as forecasting the future H time steps with output $\mathcal{Y} \in \mathbb{R}^{N \times H}$, as well as anomaly detection and missing-value imputation on the input window with output $\mathcal{Y} \in \mathbb{R}^{N \times T}$. In addition, we consider the sequence-level classification task, which assigns the input MTS to one of C categories, with output $\mathcal{Y} \in \mathbb{R}^{1 \times C}$. Table 1 summarizes the main notations used throughout the paper.

■ 2.2 Mamba

■ 2.2.1 State space model (SSM)

The classical State Space Models (SSMs) describe the state evolution of a linear time-invariant system, which map input signal $\mathbf{x} = \{x(1), \dots, x(t), \dots\} \in \mathbb{R}^{L \times D} \mapsto \mathbf{y} = \{y(1), \dots, y(t), \dots\} \in \mathbb{R}^{L \times D}$ through implicit latent state $h(t) \in \mathbb{R}^{N \times D}$, where t , L , D , and N indicate the time step, sequence length, channel number of the signal and state size, respectively. These models can be formulated as the following linear ordinary differential equations:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \quad y(t) = \mathbf{C}h(t) + \mathbf{D}x(t), \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the state transition matrix that describes how states change over time, $\mathbf{B} \in \mathbb{R}^{N \times D}$ is the input matrix that controls how inputs affect state changes, $\mathbf{C} \in \mathbb{R}^{N \times D}$ denotes the output matrix that indicates how outputs are generated based on current states and $\mathbf{D} \in \mathbb{R}^D$ represents the command coefficient that determines how inputs affect outputs directly. Most SSMs exclude the second term in the observation equation, i.e., set $\mathbf{D}x(t) = 0$, which corresponds to a

Table 1 Main notations in DeMa

Symbol	Meaning
$\mathcal{X} \in \mathbb{R}^{N \times T}$	Input multivariate time series.
N, T	Number of variates and lookback length.
L, D	Token length after patching and embedding dimension.
H, C	Prediction horizon and number of classes.
$\mathcal{Y}$	Task-specific output.
a, b	Variate indices.
l, j	Token indices.
$\tilde{\mathbf{X}} \in \mathbb{R}^{N \times L \times D}$	Cross-Time component produced by AFF.
$\hat{\mathbf{X}} \in \mathbb{R}^{L \times N \times D}$	Cross-Variate component produced by AFF.
τ_{ab}	Global correlation delay between variates a and b .
ρ_{ab}	Correlation strength for cross-variate aggregation.
Δ_{ab}	Token-level delay shift derived from τ_{ab} .
$\delta_{l,j}^{a \leftarrow b}$	Effective delay between query (a, l) and key (b, j) .
$\hat{\mathbf{Y}}$	Output of the temporal path (Mamba-SSD).
$\hat{\mathbf{Y}}$	Output of the variate path (Mamba-DALA).
$\mathbf{Z}$	Final task-agnostic representation.

skip connection in deep learning models. The time-continuous nature poses challenges for integration into deep learning architectures. To alleviate this issue, most methods utilize the Zero-Order Hold rule [33] to discretize continuous time into K intervals, which assumes that the function value remains constant over the interval $\Delta \in \mathbb{R}^D$. Equation (1) can be reformulated as

$$h_t = \bar{\mathbf{A}} h_{t-1} + \bar{\mathbf{B}} x_t, \quad y_t = \mathbf{C} h_t, \quad (2)$$

where $\bar{\mathbf{A}} = \exp(\Delta \mathbf{A})$ and $\bar{\mathbf{B}} = (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B}$. Discrete SSMs can be interpreted as a combination of CNNs and RNNs. Typically, the model employs a convolutional mode for efficient, parallelizable training and switches to a recurrent mode for efficient autoregressive inference. The formulations in Eq. (2) are equivalent to the following convolution [44]:

$$\begin{aligned} \bar{\mathbf{K}} &= (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \mathbf{C}\bar{\mathbf{A}}^{k-1}\bar{\mathbf{B}}, \dots), \\ \mathbf{y} &= \mathbf{x} * \bar{\mathbf{K}}. \end{aligned} \quad (3)$$

Thus, the overall process can be represented as

$$\mathbf{y} = \text{SSM}(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C})(\mathbf{x}). \quad (4)$$

2.2.2 Selective state space model (Mamba)

The discrete SSMs are based on data-independent parameters, meaning that parameters $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, and $\mathbf{C}$ are time-invariant and the same for any input, limiting their effectiveness in compressing context into a smaller state [33]. Mamba introduces a selective mechanism to selectively retain, propagate, or suppress information according to the current token [33,35]. This directly addresses a major weakness of earlier subquadratic sequence models, namely

their limited ability for content-based reasoning. As a result, Mamba preserves a form of context-aware sequence modeling that is much closer in spirit to attention, while avoiding the dense token-to-token interactions of standard self-attention. Specifically, it utilizes linear projection to parameterize the weight matrices $\{\mathbf{B}_t\}_{t=1}^L$, $\{\mathbf{C}_t\}_{t=1}^L$ and $\{\Delta_t\}_{t=1}^L$ according to model input $\{x_t\}_{t=1}^L$, improving the context-aware ability, i.e.,

$$\begin{aligned} \bar{\mathbf{B}}_t &= \text{Linear}_{\mathbf{B}}(x_t), \\ \bar{\mathbf{C}}_t &= \text{Linear}_{\mathbf{C}}(x_t), \\ \Delta_t &= \text{Softplus}(\text{Linear}_{\Delta}(x_t)). \end{aligned} \quad (5)$$

Then the output sequence $\{y_t\}_{t=1}^L$ can be computed with those input-adaptive discretized parameters as follows:

$$h_t = \bar{\mathbf{A}}_t h_{t-1} + \bar{\mathbf{B}}_t x_t, \quad y_t = \bar{\mathbf{C}}_t h_t. \quad (6)$$

During training, both computation and memory scale linearly with sequence length, rather than quadratically as in standard Transformers. During autoregressive inference, Mamba only needs to update a recurrent hidden state rather than maintaining an ever-growing KV cache over the full context. Moreover, Mamba introduces a hardware-aware parallel scan algorithm [45], enabling selective and time-varying SSMs to be practically efficient on modern accelerators.

2.2.3 Mamba-2

Mamba-2 [34] introduces a comprehensive framework, Structured State-Space Duality (SSD). Leveraging SSD, it reformulates SSMs as semi-separable matrices via matrix transformation and develops a more hardware-efficient computation method based on block-decomposed matrix multiplication, i.e.,

$$\begin{aligned} \mathbf{y} &= \text{SSD}(\mathbf{A}, \mathbf{B}, \mathbf{C})(\mathbf{x}) \\ &= \mathbf{M}\mathbf{x} \\ \text{s.t. } \mathbf{M} &= \begin{pmatrix} \mathbf{C}_0^\top \mathbf{A}_{0:0} \mathbf{B}_0 & & & \\ \mathbf{C}_1^\top \mathbf{A}_{1:0} \mathbf{B}_0 & \mathbf{C}_1^\top \mathbf{A}_{1:1} \mathbf{B}_1 & & \\ \dots & \dots & \dots & \dots \\ \mathbf{C}_j^\top \mathbf{A}_{j:0} \mathbf{B}_0 & \mathbf{C}_j^\top \mathbf{A}_{j:1} \mathbf{B}_1 & \dots & \mathbf{C}_j^\top \mathbf{A}_{j:i} \mathbf{B}_i \end{pmatrix}, \end{aligned} \quad (7)$$

where $\mathbf{M}$ denotes the matrix form of SSMs that uses the sequentially semiseparable representation, and $\mathbf{M}_{ji} = \mathbf{C}_j^\top \mathbf{A}_{ji} \mathbf{B}_i$, $\mathbf{C}_j$ and $\mathbf{B}_i$ represent the selective space state matrices associated with input tokens x_j and x_i , respectively. $\mathbf{A}_{ji}$ denotes the selective matrix of hidden states corresponding to the input tokens ranging from j to i . Mamba-2 achieves a 2-8 $\times$ faster training process than Mamba-1's parallel associative scan while remaining competitive with Transformers.

3 The proposed method: DeMa

3.1 Structure overview

As shown in Fig. 2 (left), DeMa is designed in a challenge-driven manner to address three key limitations in multivariate time-series (MTS) modeling: the entanglement of temporal and variate contexts, the lack of explicit cross-variate modeling, and the inadequate

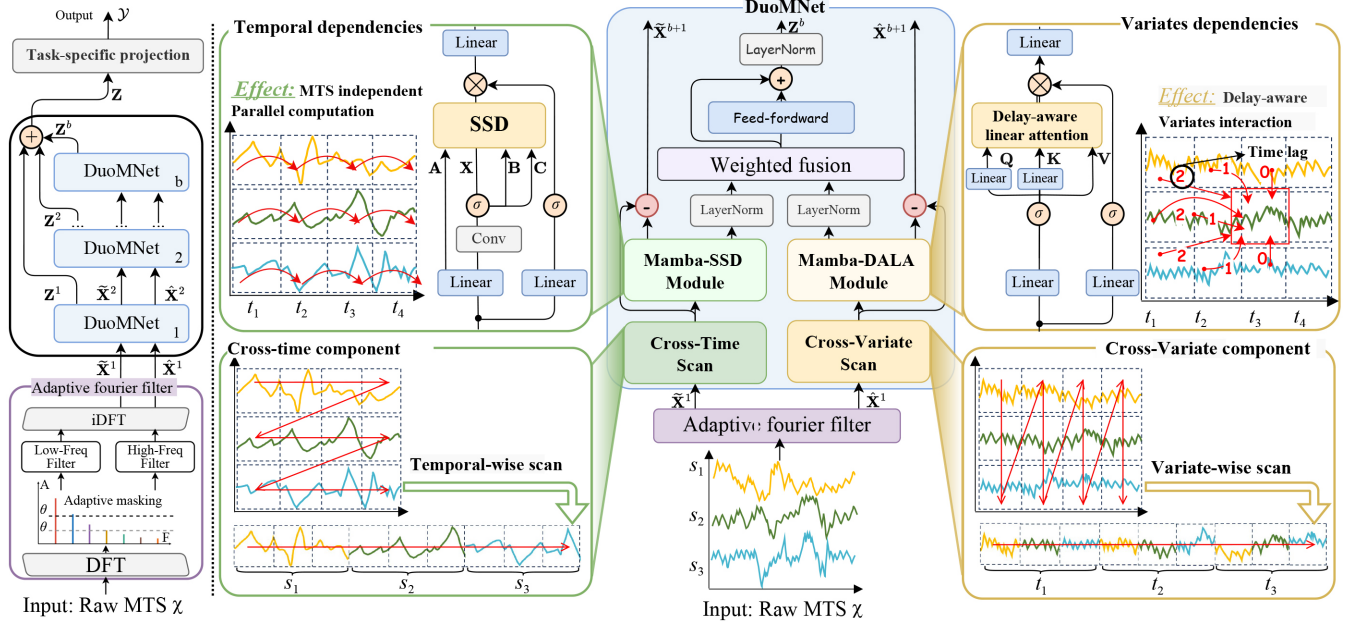


Fig. 2 The overall framework of DeMa. The proposed DeMa (Left) comprises three key components: an Adaptive Fourier Filter, stacked DuoMNet Blocks, and Task-specific Projection. Importantly, the DuoMNet Block (Right) integrates insights from the Mamba-SSD and Mamba-DALA paths to comprehensively model cross-time and cross-variate dependencies

modeling of lag-aware interactions. Accordingly, the framework consists of three stages: the *Adaptive Fourier Filter*, stacked *DuoMNet Blocks*, and *Task-specific Projection*. The *Adaptive Fourier Filter* first decomposes the raw MTS into an intra-series temporal component and an inter-series variate component, thereby reducing interference between temporal dynamics and cross-variate interactions before representation learning. Built upon this decomposition, the stacked *DuoMNet Blocks* address the second challenge through a dual-path design. As shown in Fig. 2 (right), each block contains a temporal path for modeling intra-series dependencies and a variate path for capturing inter-series dependencies, allowing the model to explicitly and separately learn these two types of structure. Furthermore, the variate path incorporates delay-aware interaction modeling, enabling DeMa to capture not only whether variables are correlated, but also how their interactions evolve. Finally, the learned shared representation is passed to lightweight *Task-specific Projection* layers for downstream tasks such as forecasting, imputation, classification, and anomaly detection.

3.1.1 Adaptive fourier filter (AFF)

Real-world multivariate time series often exhibit overlapping and entangled patterns arising from (i) *intra-series temporal dynamics* within each variate and (ii) *inter-series interactions* among correlated variates. In the frequency domain, slowly varying structures such as trends and seasonal cycles are typically concentrated in low-frequency bands. In contrast, short-term variations, including abrupt changes and time-varying cross-variate effects, tend to appear at higher frequencies [46,47]. Meanwhile, different tasks emphasize different spectral contents: numerical prediction tasks (such as forecasting, imputation, and anomaly detection) are usually sensitive

to local dynamics, whereas semantic tasks (e.g., classification) rely more on global and long-range patterns [48]. These characteristics motivate a task-adaptive spectral decomposition that separates and highlights task-relevant components.

To facilitate modeling of intra-series dynamics and inter-series interactions, we split the input into two complementary frequency subsets. The *globally dominant* frequencies capture broadly shared, long-term structures and mainly reflect per-variate temporal dynamics, while the *residual* frequencies encode more window-specific variations that are often indicative of time-varying cross-variate interactions. Concretely, given an input window $X \in \mathbb{R}^{N \times T}$ with length T , we first apply the Fast Fourier Transform (FFT) along the temporal axis for each variate and obtain the averaged amplitude for each frequency index in $S = \{0, 1, \dots, [T/2]\}$. We rank S by amplitude, select the top θ fraction as S_θ , and denote the complement by $\bar{S}_\theta = S \setminus S_\theta$. The Adaptive Fourier Filter AFF($\cdot$) then decomposes X as

$$\text{Cross-Time Component : } \tilde{X} = \text{FFT}^{-1}(\text{Filter}(S_\theta, \text{FFT}(X))),$$

$$\text{Cross-Variate Component : } \hat{X} = \text{FFT}^{-1}(\text{Filter}(\bar{S}_\theta, \text{FFT}(X))), \quad (8)$$

where FFT^{-1} denotes the inverse FFT and $\text{Filter}(\cdot)$ retains only the coefficients indexed by the specified set. The *Cross-Time Component* $\tilde{X}$ is dominated by global, low-frequency spectral components that primarily reflect intra-series temporal dynamics, and *Cross-Variate Component* $\hat{X}$ preserves the remaining variations, typically higher-frequency and window-specific, which are related to inter-series interactions.

This decomposition is particularly valuable when a state-space model serves as the backbone for MTS modeling. In Mamba, the HiPPO-based state-space formulation has been shown to induce

approximately orthogonal transformations [49]. However, the orthogonality of the learned transform does not imply that the mixed components in the raw MTS can be cleanly separated. If two underlying components are *not* orthogonal in the input space, their projections onto any orthogonal basis must share some basis directions. We formalize this limitation below.

Theorem 1 (Non-orthogonality implies basis overlap). Given the input series $\mathcal{X} = \tilde{\mathcal{X}} + \hat{\mathcal{X}}$ and let $E = \{e_1, \dots, e_N\}$ be an orthogonal basis. Suppose

$$\tilde{\mathcal{X}} = \sum_{i=1}^N a_i e_i, \quad \hat{\mathcal{X}} = \sum_{i=1}^N b_i e_i,$$

and define the supports $\tilde{E} = \{e_i \in E : a_i \neq 0\}$ and $\hat{E} = \{e_i \in E : b_i \neq 0\}$. If $\tilde{\mathcal{X}}$ and $\hat{\mathcal{X}}$ are not orthogonal, i.e., $\langle \tilde{\mathcal{X}}, \hat{\mathcal{X}} \rangle \neq 0$, then $\tilde{E} \cap \hat{E} \neq \emptyset$. Equivalently, two non-orthogonal components cannot be represented on disjoint subsets of an orthogonal basis.

Proof Since $E = \{e_1, \dots, e_N\}$ is an orthogonal basis, we have $\langle e_i, e_j \rangle = 0$ for $i \neq j$ and $\langle e_i, e_i \rangle > 0$ for all i . Thus,

$$\begin{aligned} \langle \tilde{\mathcal{X}}, \hat{\mathcal{X}} \rangle &= \left\langle \sum_{i=1}^N a_i e_i, \sum_{j=1}^N b_j e_j \right\rangle \\ &= \sum_{i=1}^N \sum_{j=1}^N a_i b_j \langle e_i, e_j \rangle \\ &= \sum_{i=1}^N a_i b_i \langle e_i, e_i \rangle. \end{aligned} \quad (9)$$

If $\tilde{E} \cap \hat{E} = \emptyset$, then for every i at least one of a_i or b_i is zero, implying $a_i b_i = 0$ for all i , which implies

$$\langle \tilde{\mathcal{X}}, \hat{\mathcal{X}} \rangle = \sum_{i=1}^N a_i b_i \langle e_i, e_i \rangle = 0. \quad (10)$$

This contradicts the assumption $\langle \tilde{\mathcal{X}}, \hat{\mathcal{X}} \rangle \neq 0$. Therefore, there exists at least one index i such that $a_i \neq 0$ and $b_i \neq 0$, i.e., $e_i \in \tilde{E} \cap \hat{E}$.

Theorem 1 implies that if the Cross-Time component $\tilde{\mathcal{X}}$ and the Cross-Variate component $\hat{\mathcal{X}}$ are non-orthogonal, their representations on any orthogonal basis must overlap, making a complete separation by an orthogonal transform impossible. In practice, this non-orthogonality often arises because different generative factors are *not* confined to disjoint frequency bands: for example, trend, seasonality, and interaction-driven fluctuations can partially overlap in the same spectral range and thus superimpose in both the time and frequency domains. Therefore, although HiPPO-based SSMs tend to learn orthogonal transformations [49], applying a single Mamba model directly to raw MTS may still entangle these factors. In contrast, our proposed AFF($\cdot$) explicitly decomposes $\mathcal{X}$ into complementary spectral parts, allowing subsequent modules to model them separately and thereby reducing representational entanglement and avoiding redundant modeling capacity.

■ 3.2 DuoMNet block

In this section, we detail the design of the proposed DuoMNet block, which comprises a temporal path to capture cross-time dependencies and a variate path to learn cross-variate dependencies.

■ 3.2.1 Cross-time scan and cross-variate scan

Given the decomposed Cross-Time Component $\tilde{\mathcal{X}} \in \mathbb{R}^{N \times T}$ and Cross-Variate Component $\hat{\mathcal{X}} \in \mathbb{R}^{N \times T}$, we tokenize each component into patch tokens to reduce point-wise redundancy and encode local temporal semantics [10,27]. Concretely, for each variate i , we apply reversible instance normalization (RevIN) [50] and partition the normalized series into L contiguous patches of length P . A lightweight 1-D convolutional encoder maps each patch to a D -dimensional embedding. To enable complementary dependency modeling, we organize the resulting embeddings with two scan orders:

(a) *Cross-Time Scan (temporal-wise)*. We preserve the temporal order of patch tokens in the Cross-Time Component and stack all variates in parallel:

$$\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_{1,:}; \tilde{\mathbf{X}}_{2,:}; \dots; \tilde{\mathbf{X}}_{N,:}] \in \mathbb{R}^{N \times L \times D}, \quad (11)$$

where $\tilde{\mathbf{X}}_{i,:} \in \mathbb{R}^{L \times D}$ contains the L patch embeddings of the i th variate. This scan order enables the temporal module to model *intra-series* dependencies along the temporal axis independently for each variate.

(b) *Cross-Variate Scan (variate-wise)*. We instead group tokens in the Cross-Variate Component by the same patch position and scan across variates:

$$\hat{\mathbf{X}} = (\hat{\mathbf{X}}_{:,1}, \hat{\mathbf{X}}_{:,2}, \dots, \hat{\mathbf{X}}_{:,L}) \in \mathbb{R}^{L \times N \times D}, \quad (12)$$

where $\hat{\mathbf{X}}_{:,l} = (\hat{\mathbf{x}}_{1,l}, \dots, \hat{\mathbf{x}}_{N,l}) \in \mathbb{R}^{N \times D}$ collects the embeddings of all N variates at the l th token step. This scan order preserves the chronological order over token indices and facilitates modeling time-varying cross-variate interactions at different token steps.

These two scanned representations, $\tilde{\mathbf{X}}$ and $\hat{\mathbf{X}}$, are then fed into subsequent modules to learn cross-time and cross-variate dependencies, respectively.

■ 3.2.2 Cross-time dependencies modeling via Mamba-SSD

Given the temporal scanned embeddings $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times L \times D}$ of the Cross-Time component, our goal is to capture *intra-series* temporal dependencies within each variate. We therefore adopt an *independent and parallel* strategy: each variate is modeled without cross-variate coupling, enabling efficient parallel computation across N variates. The proposed Mamba-SSD module consists of four stages: (a) two-branch gating and local mixing, (b) selective SSM parameterization, (c) Structured State Space Duality (SSD) based parallel scan, and (d) gated output projection.

(a) *Two-branch gating and local mixing*. We first apply linear projections to obtain a content branch $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times L \times D_c}$ and a gate branch $\tilde{\mathbf{X}}_{\text{gate}} \in \mathbb{R}^{N \times L \times D_g}$. The gate branch serves as a residual controller that adaptively filters and reweights the content features. We then perform local temporal mixing by applying a lightweight 1-D convolution along the token axis on the content branch (for each variate in parallel), which injects an explicit short-range inductive bias to model local continuity that is common in real-world time series, and complements the long-range selective SSM by providing locally aggregated features that stabilize token-dependent parameter generation.

(b) **Selective SSM parameterization.** Following Mamba-style selective SSMs, we generate token-dependent parameters from $\tilde{\mathbf{X}}$. Let D_h denote the state dimension. The step size and the output projections are computed as

$$\begin{aligned}\Delta &= \text{softplus}(\tilde{\mathbf{X}}\mathbf{W}_\Delta + \mathbf{b}_\Delta) \in \mathbb{R}^{N \times L \times D_h}, \\ \mathbf{B} &= \tilde{\mathbf{X}}\mathbf{W}_B + \mathbf{b}_B \in \mathbb{R}^{N \times L \times D_h}, \\ \mathbf{C} &= \tilde{\mathbf{X}}\mathbf{W}_C + \mathbf{b}_C \in \mathbb{R}^{N \times L \times D_h},\end{aligned}\quad (13)$$

where $\mathbf{W}_\Delta, \mathbf{W}_B, \mathbf{W}_C \in \mathbb{R}^{D_u \times D_h}$. To ensure stable dynamics, we parameterize the continuous-time transition with a negative diagonal vector $\mathbf{A} = -\exp(\mathbf{A}_{\log}) \in \mathbb{R}^{D_h}$ and discretize it using Δ :

$$\begin{aligned}\bar{\mathbf{A}} &= \exp(\Delta \odot \mathbf{A}) \in \mathbb{R}^{N \times L \times D_h}, \\ \bar{\mathbf{B}} &= (\mathbf{I} - \bar{\mathbf{A}}) \odot \mathbf{B} \in \mathbb{R}^{N \times L \times D_h},\end{aligned}\quad (14)$$

where $\odot$ denotes element-wise multiplication. Here, Δ controls the effective memory scale at each token, while $\mathbf{B}$ and $\mathbf{C}$ modulate how content is written into and read from the latent state, yielding content-adaptive dynamics.

(c) **SSD-based parallel computation.** Given the input token sequence $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times L \times D_u}$, the selective SSM at step t can be formulated as follows:

$$\begin{aligned}h_t &= \bar{\mathbf{A}}_t h_{t-1} + \bar{\mathbf{B}}_t \tilde{\mathbf{X}}_t, \\ \mathbf{Y}_t &= \mathbf{C}_t^\top h_t,\end{aligned}\quad (15)$$

which yields $\mathbf{Y} \in \mathbb{R}^{N \times L \times D_h}$. Equivalently, the induced linear operator is *block-diagonal* across variates by leveraging Structured State-Space Duality (SSD) properties in Mamba-2 [34]. Specifically, the selective SSM model presented in Eq. (15) can be reformulated as

$$\begin{aligned}\mathbf{Y}_{\text{time}} &= \text{SSD}(\mathbf{A}, \mathbf{B}, \mathbf{C})(\tilde{\mathbf{X}}) = \mathbf{M}\tilde{\mathbf{X}} \\ \text{s.t. } \mathbf{M} &= \begin{pmatrix} \mathbf{C}_0^\top \mathbf{A}_{0:0} \mathbf{B}_0 & & & \\ \mathbf{C}_1^\top \mathbf{A}_{1:0} \mathbf{B}_0 & \mathbf{C}_1^\top \mathbf{A}_{1:1} \mathbf{B}_1 & & \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_j^\top \mathbf{A}_{j:0} \mathbf{B}_0 & \mathbf{C}_j^\top \mathbf{A}_{j:1} \mathbf{B}_1 & \cdots & \mathbf{C}_j^\top \mathbf{A}_{j:j} \mathbf{B}_j \end{pmatrix},\end{aligned}\quad (16)$$

where $\mathbf{M}$ denotes the matrix form of SSMs that uses the sequentially semiseparable representation, and $\mathbf{M}_{ji} = \mathbf{C}_j^\top \mathbf{A}_{j:i} \mathbf{B}_i$ represents the mapping from the i th input token to the j th output token. Here, $\mathbf{B}_i$ and $\mathbf{C}_j$ are the selective state-space parameters associated with the i th and j th tokens, respectively, and $\mathbf{A}_{ji}$ denotes the product of the transition terms from step i to j .

By representing state-space models as semiseparable matrices via matrix transformations, the structured state-space model enables block-based matrix multiplication, allowing content-aware modeling analogous to attention while achieving subquadratic-time computation. Building on this insight, we perform the SSD operator in parallel over all N variates:

$$\begin{aligned}\mathbf{Y}_{\text{time}} &= \text{SSD}(\mathbf{A}, \mathbf{B}, \mathbf{C})(\tilde{\mathbf{X}}) \\ &= (\mathbf{M}_1 \tilde{\mathbf{X}}_{1,:}; \mathbf{M}_2 \tilde{\mathbf{X}}_{2,:}; \dots; \mathbf{M}_N \tilde{\mathbf{X}}_{N,:}),\end{aligned}\quad (17)$$

where each $\mathbf{M}_n$ is a semiseparable operator acting only on the n th variate sequence. Notably, during this process, we prevent any cross-series information flow by re-initializing (i.e., zeroing) the hidden state at the beginning of each series, thereby forcing the Mamba-SSD

path to focus exclusively on temporal modeling. This explicitly enforces that the Cross-Time branch captures temporal dynamics *within* each variate while remaining fully parallelizable across N variates.

(d) **Gated output projection.** Finally, we combine the SSD output $\mathbf{Y}_{\text{time}}$ from the content branch with the gate branch to obtain the final output:

$$\tilde{\mathbf{Y}} = \text{Linear}(\mathbf{Y}_{\text{time}} \odot \sigma(\tilde{\mathbf{X}}_{\text{gate}})), \quad (18)$$

where $\sigma(\cdot)$ is the sigmoid function and $\odot$ denotes element-wise multiplication. The purpose of this gated projection is to provide an explicit, token-adaptive modulation on the propagated state features, selectively enhancing informative temporal responses while attenuating irrelevant or noisy activations, thereby improving robustness and expressiveness without introducing cross-variational coupling. Thus, the overall process in the Mamba-SSD module is

$$\tilde{\mathbf{Y}} = \text{Mamba-SSD}(\tilde{\mathbf{X}}). \quad (19)$$

3.2.3 Cross-variate dependencies modeling via mamba-DALA

A critical yet often overlooked factor in multivariate time series modeling is the *delay dependency*: variations in one variate may influence another only after a *time lag*. For instance, in traffic forecasting, congestion at an intersection typically propagates to nearby intersections with a non-negligible delay rather than instantaneously. This lagged propagation property makes forecasting more challenging. Accurately capturing such delayed interactions is essential for both predictive performance and practical utility, yet this effect is often overlooked in existing time series models.

To efficiently capture cross-variate dependencies in the Cross-Variate component $\hat{\mathbf{X}} \in \mathbb{R}^{L \times N \times D}$ while explicitly accounting for delay effect, we develop a Mamba-style mixing block that replaces the recurrent SSM in the Mamba backbone with a **Delay-Aware Linear Attention (DALA)** operator, termed **Mamba-DALA**. Built upon Mamba's efficient block design [51], Mamba-DALA performs token-wise pairwise interaction modeling across variates, yielding delay-aware cross-variate aggregation among multiple correlated latent series.

(a) **Mamba-style gating and parameterization.** Following the two-branch design in Mamba, we apply linear projections to obtain a content branch $\hat{\mathbf{X}} \in \mathbb{R}^{L \times N \times D_u}$ and a gate branch $\hat{\mathbf{X}}_{\text{gate}} \in \mathbb{R}^{L \times N \times D_u}$. We further construct the linear-attention operator by $\hat{\mathbf{X}}^q = \hat{\mathbf{X}}\mathbf{W}_Q$, $\hat{\mathbf{X}}^k = \hat{\mathbf{X}}\mathbf{W}_K$, $\hat{\mathbf{X}}^v = \hat{\mathbf{X}}\mathbf{W}_V$, where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D_u \times D_u}$.

(b) **Delay-aware linear attention (DALA).** To robustly encode propagation delays, we consider two delay signals: a *global correlation delay* and a *token-level relative delay*. The key idea is to first correct coarse cross-series misalignment with a global offset, and then explicitly model the relative temporal distance between interacting tokens. In this way, the model can both identify the most plausible cross-series alignment and preserve ordered, causal interactions at the token level. Accordingly, we first estimate a correlation-based delay by maximizing the cross-correlation between variates, which serves as an explicit prior for cross-series alignment.

Importantly, this correlation-based delay is not imposed as a hard alignment constraint. Instead, it provides a coarse global prior. At the same time, the final delay-aware interaction is further refined by token-level relative delay encoding via Rotary Position Embedding (RoPE) [52], which captures time lags between query-key token pairs.

Global correlation delay. we first estimate the global propagation delay by maximizing the cross-correlation [53] between latent correlated series $\mathcal{X}_{a,:} \in \mathbb{R}^T$ and $\mathcal{X}_{b,:} \in \mathbb{R}^T$ via temporal shifting [42]. Meanwhile, we quantify the *correlation strength* as the peak correlation value:

$$\begin{aligned}\tau_{ab} &= \arg \max_t \text{corr}(\mathcal{X}_{a,:} \xrightarrow{t} \mathcal{X}_{b,:}), \\ \rho_{ab} &= \text{corr}(\mathcal{X}_{a,:} \xrightarrow{\tau_{ab}} \mathcal{X}_{b,:}),\end{aligned}\quad (20)$$

where $\xrightarrow{t}$ denotes a t -step shift applied to $\mathcal{X}_{a,:}$, and $\text{corr}(\cdot)$ represents the Pearson correlation function. The strength ρ_{ab} is used to weight cross-variate aggregation, so that more strongly correlated pairs contribute more. In contrast, noisier or less reliable pairs are naturally downweighted, reducing the influence of noisy or weakly correlated variate pairs. Since τ_{ab} is defined in the original time point scale, we map it to the token scale as an integer shift:

$$\Delta_{ab} = \text{round}\left(\frac{\tau_{ab}}{P}\right) \in \mathbb{Z}, \quad (21)$$

where P is the number of time points per token. Here, $\Delta_{ab} > 0$ indicates that series a lags behind series b on the token scale.

$$\Delta_{ab} = \text{round}\left(\frac{\tau_{ab}}{P}\right) \in \mathbb{Z}, \quad (22)$$

where P is the time points per token, $\Delta_{ab} > 0$ indicates that series a lags behind series b on the token scale.

Token-level relative delay. For a query token $\hat{\mathbf{X}}_{a,l}^q$ at position l and a key token $\hat{\mathbf{X}}_{b,j}$ at position j , RoPE parameterizes their token-wise relative delay $(l-j)$ as

$$\mathcal{R}_{l-j}^\Theta = \mathcal{R}_l^\Theta (\mathcal{R}_j^\Theta)^T, \quad (23)$$

where $\mathcal{R}_l^\Theta$ and $\mathcal{R}_j^\Theta$ denote rotary position embeddings for the l th and j th tokens, respectively, and Θ is the shared predefined RoPE parameterization [52]. To incorporate the global token shift, we align the key token $\hat{\mathbf{X}}_{b,j}$ from series b to the effective position $j + \Delta_{ab}$ when interacting with series a . Consequently, the query-key interaction is governed by a unified effective delay:

$$\delta_{l,j}^{a \leftarrow b} = l - (j + \Delta_{ab}) = (l - j) - \Delta_{ab}, \quad (24)$$

which jointly accounts for the global propagation offset and the local token-wise lag.

Delay-aware attention aggregation. Given a query token $\hat{\mathbf{X}}_{a,l}^q$ at position l in series a , we allow it to attend to key tokens $\hat{\mathbf{X}}_{b,j}^k$ from all variates $b \in \{1, \dots, N\}$, while enforcing the causal constraint $j + \Delta_{ab} \leq l$ to avoid information leakage. We inject the effective delay by applying RoPE to the query at position l and to the key at position $j + \Delta_{ab}$, yielding $\mathcal{R}_l^\Theta$ and $\mathcal{R}_{j+\Delta_{ab}}^\Theta$, respectively. The delay-aware linear

attention output $\mathbf{y}_{a,l} \in \mathbb{R}^{D_u}$ is computed as

$$\mathbf{y}_{a,l} = \mathcal{R}_l^\Theta \phi(\hat{\mathbf{X}}_{a,l}^q) \frac{\sum_{b=1}^N \rho_{ab} \sum_{j: j+\Delta_{ab} \leq l} (\mathcal{R}_{j+\Delta_{ab}}^\Theta \phi(\hat{\mathbf{X}}_{b,j}^k))^\top \hat{\mathbf{X}}_{b,j}^v}{\phi(\hat{\mathbf{X}}_{a,l}^q) \sum_{b=1}^N \rho_{ab} \sum_{j: j+\Delta_{ab} \leq l} \phi(\hat{\mathbf{X}}_{b,j}^k)^\top}, \quad (25)$$

where $\phi(\cdot)$ is the kernel function in [54], i.e., $\phi(x) = f(\text{ReLU}(x))$ with $f(x) = \frac{\|x\|}{\|x\|^{**p}} x^{**p}$, and x^{**p} denotes the element-wise power p . This kernelization enhances the expressiveness of linear attention by amplifying informative query-key similarity patterns, while ρ_{ab} further emphasizes reliable cross-variate interactions. Applying Eq. (25) to all tokens yields the DALA output $\mathbf{Y}_{\text{variate}} \in \mathbb{R}^{L \times N \times D_u}$, which captures delay-aware cross-variate dependencies among all patched series tokens. Overall, the robustness of DALA stems from three aspects: (i) the global delay provides coarse alignment, (ii) token-level relative delay encoding further refines interactions locally, and (iii) correlation-strength weighting suppresses unreliable variate pairs.

(c) **Gated output projection.** Following Mamba's gated mixing mechanism, we modulate $\mathbf{Y}_{\text{variate}}$ with the gate branch $\hat{\mathbf{X}}_{\text{gate}} \in \mathbb{R}^{L \times N \times D_u}$ through an element-wise multiplicative gate, and then project it back to the model dimension:

$$\hat{\mathbf{Y}} = \text{Linear}(\mathbf{Y}_{\text{variate}} \odot \sigma(\hat{\mathbf{X}}_{\text{gate}})), \quad (26)$$

where $\odot$ denotes element-wise multiplication and $\sigma(\cdot)$ is the gating activation. Therefore, the overall Mamba-DALA module can be summarized as

$$\hat{\mathbf{Y}} = \text{Mamba-DALA}(\hat{\mathbf{X}}). \quad (27)$$

3.2.4 DuoMNet block design

As illustrated in Fig. 2, each DuoMNet block consists of two complementary paths: (i) a *temporal path* implemented by Mamba-SSD to capture cross-time dependencies, and (ii) a *variate path* implemented by Mamba-DALA to model delay-aware cross-variate dependencies. We update the input $\tilde{\mathbf{X}}^{b+1}$ and $\hat{\mathbf{X}}^{b+1}$ of both paths via residual connections and pass them to the next block:

$$\begin{aligned}\tilde{\mathbf{X}}^{b+1} &= \tilde{\mathbf{X}}^b - \tilde{\mathbf{Y}}^b = \tilde{\mathbf{X}}^b - \text{Mamba-SSD}(\tilde{\mathbf{X}}^b), \\ \hat{\mathbf{X}}^{b+1} &= \hat{\mathbf{X}}^b - \hat{\mathbf{Y}}^b = \hat{\mathbf{X}}^b - \text{Mamba-DALA}(\hat{\mathbf{X}}^b).\end{aligned}\quad (28)$$

To integrate the two complementary outputs, we first apply LayerNorm to each path and then perform a weighted fusion:

$$\mathbf{U}^b = \alpha \text{LN}(\tilde{\mathbf{Y}}^b) + \beta \text{LN}(\hat{\mathbf{Y}}^b), \quad (29)$$

where α and β are mixing hyperparameters. We then apply a feed-forward network (FFN) to the mixed representation and use a residual connection followed by LN to obtain the block output:

$$\mathbf{Z}^b = \text{LN}(\mathbf{U}^b + \text{FFN}(\mathbf{U}^b)). \quad (30)$$

Here, $\mathbf{Z}^b$ is the output of the b -th DuoMNet block. By stacking B DuoMNet blocks, we obtain $\{\mathbf{Z}^b\}_{b=1}^B$. Following a layer-wise aggregation strategy, the final representation is computed as

$$\mathbf{Z} = \sum_{b=1}^B \mathbf{Z}^b. \quad (31)$$

3.3 Task-specific projection

To adapt DeMa to diverse downstream tasks (e.g., forecasting, imputation, anomaly detection, and classification), we attach a lightweight task head on top of the final task-agnostic representation $\mathbf{Z} \in \mathbb{R}^{N \times L \times D}$ to learn task-specific targets. For regression-oriented tasks such as forecasting, imputation, and anomaly detection, we employ an MLP as the task head $\mathcal{Y} = \text{MLP}(\mathbf{Z})$. For forecasting, the prediction is $\mathcal{Y} \in \mathbb{R}^{N \times S}$, where S is the prediction horizon. For imputation and anomaly detection, the head produces point-wise outputs over the lookback window, i.e., $\mathcal{Y} \in \mathbb{R}^{N \times T}$, where T is the lookback length. For series-level classification, we first aggregate token-wise representations into a global feature, and then apply an MLP classifier with softmax $\mathcal{Y} = \text{softmax}(\text{MLP}(\mathbf{Z}))$, where $\mathcal{Y} \in \mathbb{R}^{1 \times C}$ and C is the number of classes.

3.4 Training process and complexity analysis

The overall training procedure is summarized in Algorithm 1. The computational cost of DeMa is dominated by two parallel modules, Mamba-SSD and Mamba-DALA. Let L be the number of patches and D the model dimension. For Mamba-SSD, the Cross-Time scan yields $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times L \times D}$, which can be viewed as a scanned sequence of length NL for SSD-based selective state-space computation. The resulting FLOPs scale linearly with the scanned length and quadratically with the feature dimension, giving a per-block complexity of $O((NL)D^2)$. Similarly, Mamba-DALA operates on $\hat{\mathbf{X}} \in \mathbb{R}^{L \times N \times D}$, which can equivalently be viewed as a sequence of length LN . Its core delay-aware linear mixing is implemented via linear attention and thus also requires $O((LN)D^2)$ FLOPs per block. Since the two paths are computed in parallel, the overall complexity

Algorithm 1 DeMa

Input: Raw MTS window $\mathcal{X} \in \mathbb{R}^{N \times T}$; top-ratio θ ; patch length P ; #blocks B ; fusion weights (α, β) ; task head $\text{MLP}(\cdot)$.
Output: Task output $\mathcal{Y}$ and model $\mathcal{F}_\Theta$.

```

/* Adaptive Fourier Filter (AFF) */
 $(\tilde{\mathcal{X}}, \hat{\mathcal{X}}) \leftarrow \text{AFF}(\mathcal{X}; \theta)$  (Eq. (8));
/* Tokenization + Scan */
 $\tilde{\mathbf{X}}^1 \leftarrow \text{CrossTimeScan}(\text{PatchEmbed}(\tilde{\mathcal{X}}; P)) \in \mathbb{R}^{N \times L \times D}$ ;
 $\hat{\mathbf{X}}^1 \leftarrow \text{CrossVariateScan}(\text{PatchEmbed}(\hat{\mathcal{X}}; P)) \in \mathbb{R}^{L \times N \times D}$ ;
/* Delay priors */
Compute  $(\tau_{ab}, \rho_{ab})$  by Eq. (20) and  $\Delta_{ab}$  by Eq. (22) for all series pair  $(a, b)$ ;
/* Stacked DuoMNet Blocks */
for  $b = 1, 2, \dots, B$  do
    /* Parallel paths (Temporal & Variate) */
     $\tilde{\mathbf{Y}}^b \leftarrow \text{Mamba-SSD}(\tilde{\mathbf{X}}^b)$  (Eqs. (13)-(19));
     $\hat{\mathbf{Y}}^b \leftarrow \text{Mamba-DALA}(\hat{\mathbf{X}}^b; \{\Delta_{ab}\}, \{\rho_{ab}\})$  (Eqs. (24)-(27));
    /* Residual update */
     $(\tilde{\mathbf{X}}^{b+1}, \hat{\mathbf{X}}^{b+1}) \leftarrow (\tilde{\mathbf{X}}^b - \tilde{\mathbf{Y}}^b, \hat{\mathbf{X}}^b - \hat{\mathbf{Y}}^b)$  (Eq. (28));
    /* Fusion + FFN */
     $\mathbf{U}^b \leftarrow \alpha \text{LN}(\tilde{\mathbf{Y}}^b) + \beta \text{LN}(\hat{\mathbf{Y}}^b)$  (Eq. (29));
     $\mathbf{Z}^b \leftarrow \text{LN}(\mathbf{U}^b + \text{FFN}(\mathbf{U}^b))$  (Eq. (30));
/* Aggregation + Task head */
 $\mathbf{Z} \leftarrow \sum_{b=1}^B \mathbf{Z}^b$  (Eq. (31));
 $\mathcal{Y} \leftarrow \text{MLP}(\mathbf{Z})$ ;
return  $\mathcal{Y}, \mathcal{F}_\Theta$ 

```

per DuoMNet block remains $O((NL)D^2)$. Consequently, DeMa achieves linear-time complexity with respect to the effective scanned length NL , enabling efficient modeling for long-horizon and large-scale multivariate time series in regimes where the effective token length dominates the feature dimension, i.e., $N, L \gg D$. Therefore, although DeMa contains multiple modules, its dominant computation remains concentrated in two linear-time paths, making the overall architecture computationally practical.

4 Experiments

In this section, we investigate the following research questions to validate the effectiveness, efficiency, and scalability of DeMa:

- **RQ1:** How does DeMa perform across five representative time series tasks?
- **RQ2:** Does DeMa achieve a favorable accuracy-efficiency trade-off as a general backbone for multivariate time-series analysis?
- **RQ3:** How does DeMa scale with increasing input length in terms of per-iteration runtime and GPU memory, compared with Transformer-based baselines?
- **RQ4:** How does each key component contribute to the overall performance of DeMa?
- **RQ5:** How sensitive is DeMa to major hyperparameters?

4.1 Experimental setup

4.1.1 Datasets

To validate the effectiveness of DeMa, we conduct extensive experiments across five mainstream time-series analysis tasks: long-term forecasting, short-term forecasting, imputation, classification, and anomaly detection. All datasets are drawn from the benchmark collections in the Time Series Library (TSLib) [40]. Table 2 summarizes the datasets used for each task.

- **Forecasting.** We evaluate both long-term and short-term forecasting. For long-term forecasting, we adopt widely used benchmarks including Electricity [55], ETT with four subsets (ETTh1, ETTh2, ETTm1, ETTm2) [23], Exchange [56], Traffic [25], Weather [25], and Solar-Energy [56]. For short-term forecasting, we use four traffic datasets from the PEMS family [57]: PEMS03, PEMS04, PEMS07, and PEMS08.
- **Imputation.** Time-series imputation aims to recover missing values from contextual observations. We evaluate on ETTh1, ETTh2, ETTm1, ETTm2, Electricity, and Weather. Following TimesNet [18], we randomly mask time points with ratios in {12.5%, 25%, 37.5%, 50%} to assess robustness under varying missing rates.
- **Anomaly detection.** Anomaly detection aims to identify abnormal patterns in time series. We adopt five widely used industrial benchmarks: Server Machine Dataset (SMD) [58], Mars Science Laboratory rover (MSL) [59], Soil Moisture Active Passive satellite (SMAP) [59], Secure Water Treatment (SWaT) [60], and Pooled Server Metrics (PSM) [61].
- **Classification.** For time-series classification, we use ten

Table 2 Summary of experimental datasets. #Variates denotes the number of variables in each multivariate time series

Task	Dataset	#Variates	(Train, Validation, Test) Size	Frequency	Domain
Long-term forecasting	ETTh1, ETTh2	7	(8545, 2881, 2881)	1 hour	Electricity
	ETTm1, ETTm2	7	(34465, 11521, 11521)	15 min	Electricity
	Electricity	321	(18317, 2633, 5261)	1 hour	Electricity
	Weather	21	(36792, 5271, 10540)	10 min	Environment
	Traffic	862	(12185, 1757, 3509)	1 hour	Transportation
	Exchange	8	(5120,665, 1422)	1 day	Economic
	Solar-Energy	137	(36601, 5161, 10417)	10 min	Energy
Short-term forecasting	PEMS03	358	(15617, 5135, 5135)	5 min	Transportation
	PEMS04	307	(10172, 3375, 3375)	5 min	Transportation
	PEMS07	883	(16911, 5622, 5622)	5 min	Transportation
	PEMS08	170	(10690, 3548, 3548)	5 min	Transportation
Imputation	ETTh1, ETTh2	7	(8545, 2881, 2881)	1 hour	Electricity
	ETTm1, ETTm2	7	(34465, 11521, 11521)	15 min	Electricity
	Electricity	321	(18317, 2633, 5261)	1 hour	Electricity
	Weather	21	(36792, 5271, 10540)	10 min	Environment
Anomaly detection	SMD	38	(566724, 141681, 708420)	1 min	Server Machine
	MSL	55	(44653, 11664, 73729)	–	Spacecraft
	SMAP	25	(108146, 27037, 427617)	–	Spacecraft
	SWaT	51	(396000, 99000, 449919)	1 sec	Infrastructure
	PSM	25	(105984, 26497, 87841)	–	Server Machine
Classification	EthanolConcentration	3	(261, 0,263)	–	Chemistry
	FaceDetection	144	(5890, 0, 3524)	–	Neuroscience
	Handwriting	3	(150, 0,850)	–	Motion
	Heartbeat	61	(204, 0,205)	–	Health
	JapaneseVowels	12	(270, 0,370)	–	Speech
	PEMS-SF	963	(267, 0,173)	–	Transportation
	SelfRegulationSCP1	6	(268, 0,293)	–	Health
	SelfRegulationSCP2	7	(200, 0,180)	–	Health
	SpokenArabicDigits	13	(6599, 0, 2199)	–	Speech
	UWaveGestureLibrary	3	(120, 0,320)	–	Gesture

multivariate datasets from the UEA Time Series Classification Archive [62], covering diverse domains such as gesture recognition, action recognition, audio recognition, and medical diagnosis.

■ 4.1.2 Baselines

To ensure a comprehensive comparison, we include a broad set of strong baselines that are representative of the latest advances in the time series community. Specifically, our baselines span four families:

(1) Mamba-based models: Affirm [63], S-Mamba [64], CMamba [65], and SAMBA [66]; (2) Transformer-based models: iTransformer [13], Crossformer [27], and PatchTST [10]; (3) MLP-based models: TimeMixer [22], DLinear [21], and RLinear [67]; and (4) TCN-based models: ModernTCN [20] and TimesNet [18].

■ 4.2 Metrics and implementation details

To ensure a fair and comprehensive comparison, we follow the experimental protocol of the well-established Time Series Library

[40]. For long-term forecasting and imputation, we report mean squared error (MSE) and mean absolute error (MAE). For time-series classification, we report accuracy. For anomaly detection, we report the F1-score, which balances precision and recall and is well-suited to the highly imbalanced nature of anomalous events. In each table, the best and second-best results are highlighted in **red** and underlined, respectively.

We use the published optimal hyperparameter settings for all baselines. All experiments are implemented in PyTorch and conducted on three NVIDIA RTX A6000 GPUs (48 GB). We train all models with the Adam optimizer [68] under an ℓ_2 loss, and tune the initial learning rate in $\{1e-4, .5e-4, .1e-3\}$. The number of DuoMNet blocks is selected from $\{2, 3, 4, 5\}$ via hyperparameter search, and the representation dimension is chosen from $\{32, 64, 128, 256\}$. We set the batch size to 32 and train for 50 epochs. For the Adaptive Fourier Filter, the top-ratio θ is selected from a limited candidate set, and we use $\theta = 60\%$ in the final configuration. We further conduct a grid search over the fusion weights $\alpha, \beta \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$, and use $(\alpha, \beta) = (0.6, 0.4)$ for forecasting and classification, and $(\alpha, \beta) = (0.2, 0.8)$ for imputation and anomaly detection. For tokenization, we fix the patch size and stride to $P = S = 8$. DeMa follows a modular pipeline. Once the temporal path (Mamba-SSD) and the variate path (Mamba-DALA) are instantiated, the model mainly repeats the same DuoMNet block, which helps keep the implementation structured and reproducible.

■ 4.3 Main results across five time-series tasks (RQ1)

■ 4.3.1 Long-term forecasting results

Time-series forecasting is a fundamental task in time-series analysis, aiming to predict future values from historical observations. Table 3 reports the long-term forecasting results with a fixed lookback length $T = 96$ and prediction horizons $S \in \{96, 192, 336, 720\}$. Overall, DeMa delivers top-tier accuracy consistently across all datasets and horizons, ranking within the top two in every setting. In particular, DeMa achieves the best results in 34/36 cases for MSE and 30/36 cases for MAE, demonstrating strong robustness and generalization across diverse temporal dynamics.

Compared with Transformer-based models such as iTransformer, PatchTST, and Crossformer, DeMa consistently performs better across all settings, indicating that explicitly disentangling and modeling temporal and variate dependencies is more effective for long-horizon forecasting than relying primarily on global attention. DeMa also consistently surpasses Mamba variants and non-attention baselines. For example, on Traffic (Avg), DeMa reduces the MSE from 0.414 to 0.372, achieving a 10.1% improvement over the strongest Mamba baseline, S-Mamba, and also outperforms representative MLP- and TCN-based models, including TimeMixer (0.484) and ModernTCN (0.546). These gains can be attributed to DeMa's comprehensive dependency modeling: Mamba-SSD efficiently captures long-range temporal dependencies through parallel selective state-space computation, while Mamba-DALA enhances cross-variate interaction modeling. Their synergy enables DeMa to exploit long-range temporal structures and inter-series correlations jointly, yielding consistently strong performance in long-

term forecasting. We note that DeMa remains highly competitive even on datasets with only a few variates. As summarized in Table 2, ETTh1, ETTh2, ETTm1, and ETTm2 each contain 7 variates, while Exchange contains 8 variates. On these small-scale benchmarks, DeMa still achieves best or near-best averaged results in Table 3. These results indicate that the benefit of DeMa persists in small-scale scenarios, though its relative advantage becomes more pronounced as scalability pressure increases.

We further examine DeMa under relatively weak inter-variable dependencies, as analyzed on ETTh2, ETTm1, and ETTm2. Even in these settings, DeMa remains highly competitive, achieving the best average MSE on ETTm1 (0.367) and ETTm2 (0.270), while maintaining comparable performance on ETTh2 (0.370). This observation suggests that when the correlation-based delay prior becomes less informative, the temporal path can still capture the dominant intra-series dynamics, whereas the variate path provides complementary gains.

■ 4.3.2 Short-term forecasting results

We evaluate short-term forecasting on four PEMS traffic benchmarks, where accurate prediction is challenging due to intricate spatiotemporal dependencies among sensors and strong inter-series interactions that drive citywide traffic dynamics. Table 4 summarizes the results under a fixed lookback length $T = 96$ and prediction horizons $S \in \{12, 24, 48\}$. Overall, DeMa achieves the best performance in all 16 settings (both MSE and MAE), demonstrating its strong effectiveness for short-term forecasting. In terms of averaged MSE, DeMa improves over the strongest competitor by a clear margin, reducing errors by 25.8% on PEMS03, 15.7% on PEMS04, 12.7% on PEMS07, and 16.4% on PEMS08. Notably, variate-independent architectures exhibit pronounced performance degradation on these datasets. For example, PatchTST, which largely models each variate independently, is consistently inferior to DeMa across all horizons; a similar trend is observed for MLP-based methods, whose channel-independent mixing fails to capture dynamic cross-sensor interactions. We attribute the consistent gains of DeMa to its explicit and effective cross-variate modeling: Mamba-DALA strengthens cross-variate interactions, including delay-aware dependencies among traffic sensors, and complements the cross-time temporal modeling of Mamba-SSD. Their synergy enables DeMa to jointly exploit temporal dynamics and delay-aware inter-sensor dependencies, resulting in substantially improved short-term forecasting performance on the PEMS benchmarks.

■ 4.3.3 Imputation results

Time series imputation aims to recover missing values from partially observed series, which is crucial in real-world scenarios where data acquisition is imperfect due to sensor malfunctions or unexpected transmission glitches. As reported in Table 5, DeMa consistently achieves the best performance across all datasets and masking ratios, surpassing both Transformer-based and SSM-based competitors by clear margins. A notable trend is that the advantage of DeMa becomes more pronounced as the missing rate increases, where accurate recovery increasingly relies on fine-grained modeling of

Table 3 Long-term forecasting results with lookback length $T = 96$ and prediction length $S \in \{96, 192, 336, 720\}$. Lower MSE or MAE is better

Models	SSD + DALA				SSM-based				Transformer-based models						MLP-based models				TCN-based models								
	DeMa		Affirm		S-Mamba		CMamba		SAMBA		iTransformer		Crossformer		PatchTST		TimeMixer		RLinear		DLinear		ModernTCN		TimesNet		
	(Ours)		[63]		[64]		[65]		[66]		[13]		[27]		[10]		[22]		[67]		[21]		[20]		[18]		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
Electricity	96	0.137	0.201	0.148	<u>0.231</u>	<u>0.139</u>	0.235	0.141	0.231	0.146	0.244	0.148	0.240	0.219	0.314	0.181	0.270	0.153	0.247	0.201	0.281	0.197	0.282	0.171	0.269	0.168	0.272
	192	0.155	0.220	0.158	<u>0.237</u>	<u>0.159</u>	0.255	<u>0.157</u>	0.245	0.162	0.258	0.162	0.243	0.231	0.322	0.188	0.274	0.166	0.256	0.201	0.283	0.196	0.285	0.193	0.287	0.184	0.289
	336	0.172	0.252	0.177	0.268	0.176	0.272	<u>0.175</u>	<u>0.265</u>	0.177	0.274	0.178	0.269	0.246	0.337	0.204	0.293	0.185	0.277	0.215	0.298	0.209	0.301	0.205	0.297	0.198	0.300
	720	0.195	0.271	0.206	0.295	0.204	0.298	0.203	0.289	<u>0.202</u>	0.297	0.225	0.317	0.280	0.363	0.246	0.324	0.225	0.310	0.257	0.331	0.245	0.333	0.217	<u>0.276</u>	0.220	0.320
	Avg	0.165	0.236	0.172	<u>0.258</u>	0.170	0.265	<u>0.169</u>	0.258	0.172	0.268	0.178	0.270	0.244	0.334	0.205	0.290	0.182	0.272	0.219	0.298	0.212	0.300	0.197	0.282	0.192	0.295
ETT h1	96	0.359	0.372	0.375	0.396	0.386	0.405	<u>0.372</u>	<u>0.386</u>	0.376	0.400	0.386	0.405	0.423	0.448	0.414	0.419	0.375	0.400	0.386	0.395	0.386	0.400	0.371	0.389	0.384	0.402
	192	0.413	0.415	<u>0.421</u>	0.428	0.443	0.437	0.422	<u>0.416</u>	0.432	0.429	0.441	0.436	0.471	0.474	0.460	0.445	0.429	0.421	0.437	0.424	0.437	0.432	0.423	0.439	0.436	0.429
	336	0.451	<u>0.438</u>	<u>0.458</u>	0.472	0.489	0.468	0.466	0.438	0.477	0.437	0.487	0.458	0.570	0.546	0.501	0.466	0.484	0.458	0.479	0.446	0.481	0.459	0.486	0.445	0.491	0.469
	720	0.467	0.452	0.493	0.469	0.502	0.489	<u>0.470</u>	<u>0.461</u>	0.488	0.471	0.503	0.491	0.653	0.621	0.500	0.488	0.498	0.482	0.481	0.470	0.519	0.516	0.501	0.486	0.521	0.500
	Avg	0.423	0.419	0.434	0.441	0.455	0.450	<u>0.433</u>	<u>0.425</u>	0.443	0.432	0.454	0.447	0.529	0.522	0.469	0.454	0.447	0.440	0.446	0.434	0.456	0.452	0.445	0.440	0.458	0.450
ETT h2	96	<u>0.285</u>	<u>0.335</u>	0.300	0.344	0.296	0.348	0.281	0.329	0.288	0.340	0.297	0.349	0.745	0.584	0.302	0.348	0.289	0.341	0.288	0.338	0.333	0.387	0.321	0.348	0.340	0.374
	192	<u>0.365</u>	<u>0.383</u>	0.382	0.397	0.376	0.396	0.361	0.381	0.373	0.390	0.380	0.400	0.877	0.656	0.388	0.400	0.372	0.392	0.374	0.390	0.477	0.476	0.362	0.394	0.402	0.414
	336	0.412	0.420	0.415	0.442	0.424	0.431	0.413	0.419	0.380	0.406	0.428	0.432	1.043	0.731	0.426	0.433	<u>0.386</u>	<u>0.414</u>	0.415	0.426	0.594	0.541	0.420	0.435	0.452	0.452
	720	<u>0.418</u>	0.446	0.427	0.467	0.426	0.444	0.419	0.419	0.412	<u>0.435</u>	0.427	0.445	1.104	0.763	0.431	0.446	0.412	0.434	0.420	0.440	0.831	0.657	0.431	0.440	0.462	0.468
	Avg	0.370	0.396	0.381	0.413	0.381	0.405	0.368	0.391	0.363	<u>0.392</u>	0.383	0.407	0.942	0.684	0.387	0.407	<u>0.364</u>	0.395	0.374	0.398	0.559	0.515	0.381	0.404	0.414	0.427
ETT m1	96	0.292	<u>0.343</u>	0.337	0.369	0.333	0.368	<u>0.308</u>	0.338	0.315	0.357	0.334	0.368	0.404	0.426	0.329	0.367	0.320	0.357	0.355	0.376	0.345	0.372	0.327	0.365	0.338	0.375
	192	0.365	0.387	0.372	0.382	0.376	0.390	<u>0.359</u>	0.364	0.360	0.383	0.377	0.391	0.450	0.451	0.367	0.385	0.361	0.381	0.391	0.392	0.380	0.389	0.357	<u>0.377</u>	0.374	0.387
	336	0.388	<u>0.402</u>	0.418	0.427	0.408	0.413	0.390	0.389	<u>0.389</u>	0.405	0.426	0.420	0.532	0.515	0.399	0.410	0.390	0.404	0.424	0.415	0.413	0.413	0.402	0.410	0.410	0.411
	720	0.422	0.423	0.472	0.450	0.475	0.448	<u>0.447</u>	<u>0.425</u>	0.448	0.440	0.491	0.459	0.666	0.589	0.454	0.439	0.454	0.441	0.487	0.450	0.474	0.453	0.458	0.453	0.478	0.450
	Avg	0.367	<u>0.389</u>	0.400	0.407	0.398	0.405	<u>0.376</u>	0.379	0.378	0.394	0.407	0.410	0.513	0.496	0.387	0.400	0.381	0.395	0.414	0.407	0.403	0.407	0.386	<u>0.401</u>	0.400	0.406
ETT m2	96	0.169	0.256	0.179	0.264	0.179	0.263	<u>0.171</u>	<u>0.248</u>	0.172	0.259	0.180	0.264	0.287	0.366	0.175	0.259	0.175	0.258	0.182	0.265	0.193	0.292	0.171	0.262	0.187	0.267
	192	<u>0.230</u>	<u>0.290</u>	0.251	0.308	0.250	0.309	0.235	0.292	0.238	0.301	0.250	0.309	0.414	0.492	0.241	0.302	0.237	0.299	0.246	0.304	0.284	0.362	0.227	0.286	0.249	0.309
	336	0.282	0.327	0.317	0.350	0.312	0.349	<u>0.296</u>	<u>0.334</u>	0.300	0.340	0.311	0.348	0.597	0.542	0.305	0.343	0.298	0.340	0.307	0.342	0.369	0.427	0.325	0.335	0.321	0.351
	720	0.398	0.397	0.405	0.401	0.411	0.406	0.392	0.391	0.394	<u>0.394</u>	0.412	0.407	1.730	1.042	0.402	0.400	<u>0.391</u>	0.396	0.407	0.398	0.554	0.522	0.388	0.405	0.408	0.403
	Avg	0.270	<u>0.318</u>	0.288	0.331	0.288	0.332	<u>0.273</u>	0.316	0.276	0.322	0.288	0.332	0.757	0.610	0.281	0.326	0.275	0.323	0.286	0.327	0.350	0.401	0.278	0.322	0.291	0.333
Exchange	96	0.075	0.186	0.089	0.202	0.086	0.207	0.085	0.205	<u>0.083</u>	<u>0.202</u>	0.086	0.206	0.256	0.367	0.088	0.205	0.094	0.218	0.093	0.217	0.088	0.218	0.089	0.202	0.107	0.234
	192	0.152	0.283	0.178	0.298	0.182	0.304	0.178	0.306	0.176	<u>0.298</u>	0.177	0.299	0.470	0.509	0.176	0.299	0.184	0.307	0.184	0.307	<u>0.176</u>	0.315	0.187	0.314	0.226	0.344
	336	0.301	0.395	0.351	0.412	0.332	0.418	0.330	0.417	0.327	0.413	0.331	0.417	1.268	0.883	<u>0.301</u>	<u>0.397</u>	0.349	0.431	0.351	0.432	0.313	0.427	0.328	0.428	0.367	0.448
	720	<u>0.841</u>	0.682	0.887	0.716	0.867	0.703	0.843	0.693	0.839	<u>0.689</u>	0.847	0.691	1.767	1.068	0.901	0.714	0.852	0.698	0.886	0.714	0.839	0.695	0.925	0.721	0.964	0.746
	Avg	0.342	0.387	0.376	0.407	0.367	0.408	0.359	0.405	0.356	<u>0.401</u>	0.360	0.403	0.940	0.707	0.367	0.404	0.370	0.413	0.378	0.417	<u>0.354</u>	0.414	0.382	0.416	0.416	0.443
Traffic	96	0.341	0.235	0.443	0.277	<u>0.382</u>	0.261	0.414	<u>0.251</u>	0.388	0.261	0.395	0.268	0.522	0.290	0.462	0.295	0.462	0.285	0.649	0.389	0.650	0.396	0.513	0.335	0.593	0.321
	192	0.343	0.232	0.452	0.279	<u>0.396</u>	0.267	0.432	<u>0.257</u>	0.411	0.271	0.417	0.276	0.530	0.293	0.466	0.296	0.473	0.296	0.601	0.366	0.598	0.370	0.533	0.347	0.617	0.336
	336	0.374	0.229	0.489	0.302	<u>0.417</u>	0.276	0.446	<u>0.265</u>	0.428	0.278	0.433	0.283	0.558	0.305	0.482	0.304	0.498	0.296	0.609	0.369	0.605	0.373	0.554	0.342	0.629	0.336
	720	0.428	0.267	0.501	0.322	<u>0.460</u>	0.300	0.485	<u>0.286</u>	0.461	0.297	0.467	0.302	0.589	0.328	0.514	0.322	0.506	0.313	0.647	0.387	0.645	0.394	0.582	0.368	0.640	0.350
	Avg	0.372	0.241	0.471	0.295	<u>0.414</u>	0.276	0.444	<u>0.265</u>	0.422	0.276	0.428	0.282	0.550	0.304	0.481	0.304	0.484	0.297	0.626	0.378	0.625	0.383	0.546	0.348	0.620	0.336

Table 3 (Continued)

Models	SSD + DALA				SSM-based				Transformer-based models						MLP-based models						TCN-based models						
	DeMa (Ours)		Affirm [63]		S-Mamba [64]		CMamba [65]		SAMBA [66]		iTransformer [13]		Crossformer [27]		PatchTST [10]		TimeMixer [22]		RLinear [67]		DLinear [21]		ModernTCN [20]		TimesNet [18]		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
Weather	96	0.135	<u>0.205</u>	0.173	0.216	0.165	0.210	<u>0.150</u>	0.187	0.165	0.214	0.174	0.214	0.158	0.230	0.177	0.218	0.163	0.209	0.192	0.232	0.196	0.255	0.158	0.218	0.172	0.220
	192	0.183	0.231	0.220	0.255	0.214	0.252	<u>0.200</u>	<u>0.236</u>	0.214	0.255	0.221	0.254	0.206	0.277	0.225	0.259	0.208	0.250	0.240	0.271	0.237	0.296	0.201	0.244	0.219	0.261
	336	0.241	0.273	0.272	0.293	0.274	0.297	0.260	<u>0.281</u>	0.271	0.297	0.278	0.296	0.272	0.335	0.278	0.297	<u>0.251</u>	0.287	0.292	0.307	0.283	0.335	0.255	0.286	0.280	0.306
	720	0.323	0.337	0.356	0.343	0.350	0.345	0.339	<u>0.334</u>	0.346	0.347	0.358	0.347	0.398	0.418	0.354	0.348	<u>0.339</u>	0.341	0.364	0.353	0.345	0.381	0.346	0.337	0.365	0.359
	Avg	0.221	<u>0.262</u>	0.255	0.277	0.251	0.276	<u>0.237</u>	0.260	0.249	0.278	0.258	0.278	0.259	0.315	0.259	0.281	0.240	0.271	0.272	0.291	0.265	0.317	0.240	<u>0.271</u>	0.259	0.287
Solar-Energy	96	0.193	0.233	0.232	0.283	0.205	0.244	0.218	0.242	0.206	0.247	<u>0.203</u>	<u>0.237</u>	0.310	0.331	0.234	0.286	0.219	0.260	0.322	0.339	0.290	0.378	0.224	0.245	0.250	0.292
	192	0.226	0.257	0.258	0.301	0.237	0.270	0.247	0.276	0.238	0.271	<u>0.233</u>	<u>0.261</u>	0.734	0.725	0.267	0.310	0.238	0.288	0.359	0.356	0.320	0.398	0.258	0.271	0.296	0.318
	336	<u>0.251</u>	0.270	0.282	0.311	0.258	0.288	0.261	0.291	0.258	0.282	0.248	<u>0.273</u>	0.750	0.735	0.290	0.315	0.262	0.295	0.397	0.369	0.353	0.415	0.265	0.278	0.319	0.330
	720	<u>0.254</u>	0.273	0.291	0.318	0.260	0.288	0.265	0.293	0.267	0.292	0.249	<u>0.275</u>	0.769	0.765	0.289	0.317	0.259	0.299	0.397	0.356	0.356	0.413	0.326	0.335	0.338	0.337
	Avg	0.231	0.258	0.266	0.303	0.240	0.273	0.245	0.276	0.245	0.273	<u>0.233</u>	<u>0.262</u>	0.641	0.639	0.270	0.307	0.245	0.290	0.369	0.356	0.330	0.401	0.268	0.282	0.301	0.319
1 st Count	34	30	0	0	0	0	2	<u>11</u>	2	2	2	0	0	0	0	0	0	1	1	0	0	1	0	<u>3</u>	1	0	0

Table 4 Short-term forecasting results with lookback length $T = 96$ and prediction length $S \in \{12, 24, 48\}$. Lower MSE or MAE is better

Models	SSD + DALA				SSM-based				Transformer-based models						MLP-based models						TCN-based models						
	DeMa (Ours)		Affirm [63]		S-Mamba [64]		CMamba [65]		SAMBA [66]		iTransformer [13]		Crossformer [27]		PatchTST [10]		TimeMixer [22]		RLinear [67]		DLinear [21]		ModernTCN [20]		TimesNet [18]		
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	12	0.050	0.146	0.091	0.200	<u>0.065</u>	<u>0.169</u>	0.068	0.172	0.066	0.182	0.071	0.174	0.090	0.203	0.099	0.216	0.073	0.181	0.126	0.236	0.122	0.243	0.072	0.186	0.085	0.192
	24	0.063	0.172	0.104	0.228	0.087	<u>0.196</u>	0.079	0.201	<u>0.075</u>	0.203	0.093	0.201	0.121	0.240	0.142	0.259	0.091	0.232	0.246	0.334	0.201	0.317	0.095	0.217	0.118	0.223
	48	0.093	0.204	0.199	0.281	0.133	0.243	0.132	0.239	0.147	0.255	<u>0.125</u>	<u>0.236</u>	0.202	0.317	0.211	0.319	0.138	0.244	0.551	0.529	0.333	0.425	0.136	0.249	0.155	0.260
	Avg	0.069	0.174	0.131	0.236	0.095	<u>0.203</u>	<u>0.093</u>	0.204	0.096	0.213	0.096	0.204	0.138	0.253	0.151	0.265	0.101	0.219	0.308	0.366	0.219	0.328	0.101	0.217	0.119	0.225
PEMS04	12	0.061	0.168	0.092	0.201	0.076	0.180	0.079	0.199	<u>0.073</u>	<u>0.172</u>	0.078	0.183	0.098	0.218	0.105	0.224	0.082	0.201	0.138	0.252	0.148	0.272	0.082	0.192	0.087	0.195
	24	0.062	0.172	0.119	0.256	<u>0.084</u>	0.193	0.091	0.228	0.086	<u>0.185</u>	0.095	0.205	0.131	0.256	0.153	0.275	0.102	0.222	0.258	0.348	0.224	0.340	0.093	0.209	0.103	0.215
	48	0.087	0.193	0.187	0.283	0.115	0.224	0.156	0.266	<u>0.092</u>	<u>0.209</u>	0.120	0.233	0.205	0.326	0.229	0.339	0.157	0.289	0.572	0.544	0.355	0.437	0.123	0.227	0.136	0.250
	Avg	0.070	0.178	0.133	0.247	0.092	0.199	0.109	0.231	<u>0.083</u>	<u>0.189</u>	0.098	0.207	0.145	0.267	0.162	0.279	0.114	0.238	0.323	0.381	0.242	0.350	0.099	0.209	0.109	0.220
PEMS07	12	0.059	0.147	0.083	0.192	<u>0.063</u>	<u>0.159</u>	0.069	0.172	0.067	0.177	0.067	0.165	0.094	0.200	0.095	0.207	0.078	0.181	0.118	0.235	0.115	0.242	0.073	0.175	0.082	0.181
	24	0.069	0.162	0.128	0.227	<u>0.081</u>	<u>0.183</u>	0.092	0.194	0.094	0.211	0.088	0.190	0.139	0.247	0.150	0.262	0.108	0.239	0.242	0.341	0.210	0.329	0.095	0.193	0.101	0.204
	48	0.079	0.174	0.205	0.299	<u>0.093</u>	<u>0.192</u>	0.137	0.255	0.122	0.236	0.110	0.215	0.311	0.369	0.253	0.340	0.167	0.292	0.562	0.541	0.398	0.458	0.116	0.221	0.134	0.238
	Avg	0.069	0.161	0.139	0.240	<u>0.079</u>	<u>0.178</u>	0.099	0.207	0.094	0.208	0.088	0.190	0.181	0.272	0.166	0.270	0.118	0.237	0.307	0.372	0.241	0.343	0.095	0.196	0.106	0.208
PEMS08	12	0.052	0.158	0.141	0.192	<u>0.076</u>	<u>0.178</u>	0.082	0.199	0.080	0.191	0.079	0.182	0.165	0.214	0.168	0.232	0.144	0.196	0.133	0.247	0.154	0.276	0.082	0.181	0.112	0.212
	24	0.091	0.183	0.183	0.299	<u>0.104</u>	<u>0.209</u>	0.134	0.238	0.116	0.221	0.115	0.219	0.215	0.260	0.224	0.281	0.181	0.272	0.249	0.343	0.248	0.353	0.129	0.227	0.141	0.238
	48	0.149	0.206	0.237	0.307	<u>0.167</u>	<u>0.228</u>	0.200	0.255	0.199	0.241	0.186	0.235	0.315	0.355	0.321	0.354	0.259	0.323	0.569	0.544	0.440	0.470	0.183	0.231	0.198	0.283
	Avg	0.097	0.182	0.187	0.266	<u>0.116</u>	<u>0.205</u>	0.139	0.231	0.132	0.218	0.127	0.212	0.232	0.276	0.238	0.289	0.195	0.264	0.317	0.378	0.281	0.366	0.131	0.213	0.150	0.244
1 st Count	16	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

local fluctuations and precise temporal alignment between observed contexts and missing entries. For instance, on ETTh1 with 50% masking, DeMa attains a low error of 0.064/0.159, whereas

Transformer baselines (e.g., PatchTST: 0.173/0.271; iTransformer: 0.142/0.286) and SSM-based methods (e.g., S-Mamba: 0.142/0.281) degrade substantially, indicating limited capability in

Table 5 Imputation results on 96-length input windows with random masking ratios {12.5%,25%,37.5%,50%}. Lower MSE or MAE is better

Models	SSD + DALA				SSM-based				Transformer-based models						MLP-based models						TCN-based models						
	DeMa (Ours)		Affirm [63]		S-Mamba [64]		CMamba [65]		SAMBA [66]		iTransformer [13]		Crossformer [27]		PatchTST [10]		TimeMixer [22]		RLinear [67]		DLinear [21]		ModernTCN [20]		TimesNet [18]		
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh1	12.5%	0.032	0.122	0.133	0.242	0.112	0.221	0.101	0.211	0.116	0.256	0.111	0.223	0.099	0.218	0.094	0.199	0.127	0.232	0.098	0.206	0.151	0.267	0.035	0.128	0.057	0.159
	25%	0.042	0.131	0.142	0.249	0.131	0.242	0.127	0.238	0.138	0.247	0.129	0.257	0.125	0.243	0.119	0.225	0.135	0.267	0.123	0.229	0.180	0.292	0.042	0.140	0.069	0.178
	37.5%	0.052	0.137	0.151	0.276	0.140	0.251	0.139	0.244	0.142	0.251	0.133	0.263	0.146	0.263	0.145	0.248	0.149	0.279	0.153	0.253	0.215	0.318	0.054	0.157	0.084	0.196
	50%	0.064	0.159	0.164	0.292	0.142	0.281	0.142	0.279	0.146	0.288	0.142	0.286	0.158	0.281	0.173	0.271	0.151	0.291	0.188	0.278	0.257	0.347	0.067	0.174	0.102	0.215
	Avg	0.048	0.137	0.147	0.265	0.131	0.249	0.127	0.243	0.136	0.261	0.129	0.257	0.132	0.251	0.133	0.236	0.141	0.267	0.141	0.242	0.201	0.306	0.050	0.150	0.078	0.187
ETTh2	12.5%	0.035	0.120	0.099	0.183	0.071	0.172	0.067	0.165	0.073	0.174	0.056	0.152	0.103	0.220	0.057	0.150	0.044	0.113	0.057	0.152	0.100	0.216	0.037	0.121	0.040	0.130
	25%	0.038	0.126	0.103	0.190	0.077	0.176	0.072	0.167	0.082	0.181	0.061	0.162	0.110	0.229	0.062	0.158	0.051	0.132	0.062	0.160	0.127	0.247	0.040	0.127	0.046	0.141
	37.5%	0.041	0.138	0.118	0.199	0.082	0.188	0.078	0.179	0.088	0.190	0.076	0.169	0.129	0.246	0.068	0.168	0.062	0.191	0.068	0.168	0.158	0.276	0.043	0.134	0.052	0.151
	50%	0.045	0.142	0.122	0.217	0.085	0.203	0.083	0.191	0.090	0.208	0.079	0.177	0.148	0.265	0.076	0.179	0.083	0.200	0.076	0.179	0.183	0.299	0.048	0.143	0.060	0.162
	Avg	0.040	0.132	0.111	0.197	0.079	0.185	0.075	0.176	0.083	0.188	0.068	0.165	0.122	0.240	0.066	0.164	0.060	0.159	0.066	0.165	0.142	0.259	0.042	0.131	0.049	0.146
ETTm1	12.5%	0.015	0.081	0.061	0.152	0.047	0.131	0.052	0.135	0.055	0.123	0.059	0.128	0.037	0.137	0.041	0.128	0.042	0.113	0.047	0.137	0.058	0.162	0.015	0.082	0.019	0.092
	25%	0.020	0.083	0.073	0.177	0.065	0.169	0.063	0.157	0.068	0.163	0.062	0.142	0.038	0.141	0.043	0.130	0.055	0.135	0.061	0.157	0.080	0.193	0.018	0.088	0.023	0.101
	37.5%	0.023	0.093	0.079	0.196	0.077	0.188	0.072	0.182	0.083	0.191	0.075	0.203	0.041	0.142	0.044	0.133	0.068	0.197	0.077	0.175	0.103	0.219	0.021	0.095	0.029	0.111
	50%	0.024	0.100	0.099	0.216	0.091	0.201	0.091	0.199	0.088	0.202	0.096	0.215	0.047	0.152	0.050	0.142	0.083	0.202	0.096	0.195	0.132	0.248	0.026	0.105	0.036	0.124
	Avg	0.021	0.090	0.078	0.185	0.070	0.169	0.070	0.168	0.074	0.170	0.073	0.172	0.041	0.143	0.045	0.133	0.062	0.162	0.070	0.166	0.093	0.206	0.020	0.093	0.027	0.107
ETTm2	12.5%	0.016	0.074	0.037	0.139	0.029	0.122	0.035	0.117	0.033	0.108	0.030	0.094	0.044	0.148	0.025	0.092	0.031	0.103	0.026	0.093	0.062	0.166	0.017	0.076	0.018	0.080
	25%	0.021	0.079	0.044	0.142	0.035	0.129	0.042	0.128	0.040	0.112	0.031	0.096	0.047	0.151	0.027	0.095	0.040	0.108	0.030	0.103	0.085	0.196	0.018	0.080	0.020	0.085
	37.5%	0.023	0.095	0.059	0.155	0.046	0.132	0.053	0.137	0.047	0.137	0.039	0.107	0.044	0.145	0.029	0.099	0.044	0.117	0.034	0.113	0.106	0.222	0.020	0.084	0.023	0.091
	50%	0.025	0.099	0.066	0.161	0.051	0.145	0.061	0.166	0.059	0.159	0.041	0.129	0.047	0.150	0.032	0.106	0.047	0.139	0.039	0.123	0.131	0.247	0.022	0.090	0.026	0.098
	Avg	0.021	0.087	0.052	0.149	0.040	0.132	0.047	0.137	0.045	0.129	0.035	0.107	0.046	0.149	0.028	0.098	0.041	0.117	0.032	0.108	0.096	0.208	0.019	0.082	0.022	0.088
Electricity	12.5%	0.052	0.142	0.077	0.189	0.063	0.171	0.072	0.189	0.077	0.190	0.064	0.172	0.068	0.181	0.073	0.188	0.094	0.205	0.079	0.199	0.092	0.214	0.059	0.171	0.085	0.202
	25%	0.067	0.161	0.081	0.192	0.072	0.187	0.088	0.199	0.081	0.218	0.073	0.188	0.079	0.198	0.082	0.200	0.103	0.213	0.105	0.233	0.118	0.247	0.071	0.188	0.089	0.206
	37.5%	0.076	0.171	0.089	0.219	0.077	0.193	0.092	0.208	0.099	0.223	0.087	0.200	0.087	0.203	0.097	0.217	0.116	0.231	0.131	0.262	0.144	0.276	0.077	0.190	0.094	0.213
	50%	0.077	0.182	0.095	0.226	0.089	0.204	0.118	0.213	0.115	0.237	0.091	0.205	0.096	0.212	0.110	0.232	0.132	0.255	0.160	0.291	0.175	0.305	0.085	0.200	0.100	0.221
	Avg	0.068	0.164	0.086	0.207	0.075	0.189	0.093	0.202	0.093	0.217	0.079	0.191	0.083	0.199	0.091	0.209	0.111	0.226	0.119	0.246	0.132	0.260	0.073	0.187	0.092	0.210
Weather	12.5%	0.021	0.035	0.034	0.041	0.025	0.039	0.031	0.045	0.029	0.051	0.027	0.041	0.036	0.092	0.029	0.049	0.032	0.043	0.029	0.048	0.039	0.084	0.023	0.038	0.025	0.045
	25%	0.023	0.039	0.039	0.052	0.028	0.046	0.037	0.059	0.033	0.055	0.031	0.047	0.035	0.088	0.031	0.053	0.035	0.049	0.032	0.055	0.048	0.103	0.025	0.041	0.029	0.052
	37.5%	0.025	0.041	0.042	0.066	0.033	0.053	0.042	0.063	0.037	0.062	0.033	0.059	0.035	0.088	0.034	0.058	0.038	0.058	0.036	0.062	0.057	0.117	0.027	0.046	0.031	0.057
	50%	0.029	0.049	0.050	0.069	0.035	0.059	0.049	0.071	0.046	0.067	0.036	0.062	0.038	0.092	0.039	0.066	0.044	0.066	0.040	0.067	0.066	0.134	0.031	0.051	0.034	0.062
	Avg	0.025	0.041	0.041	0.057	0.030	0.049	0.040	0.060	0.036	0.059	0.032	0.052	0.036	0.090	0.033	0.057	0.037	0.054	0.034	0.058	0.052	0.110	0.027	0.044	0.030	0.054
1 st Count	23	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	5	0	0

capturing the local variations required for reliable completion. These results suggest that imputation is not merely a global dependency modeling problem, but a fine-grained completion task that hinges on (i) local context continuity around missing positions and (ii) accurate

relative lag relations between correlated variates. Concretely, Mamba-SSD provides efficient temporal context aggregation to supply informative local cues. At the same time, Mamba-DALA enables delay-aware cross-variate interactions, enabling observed

variates to contribute aligned evidence for reconstructing missing values. Together, their synergy enables more precise recovery of local dynamics and inter-series signals, resulting in consistently superior imputation accuracy.

4.4 Classification results

Time series classification aims to assign a semantic label to a multivariate series by recognizing both global patterns and discriminative local cues. We evaluate on 10 multivariate datasets from the UEA Time Series Archive [62] and report the average classification accuracy of each method across the 10 benchmarks. As shown in Fig. 3(a), DeMa achieves the highest average accuracy of 76.3%, consistently outperforming strong baselines from all major families, including ModernTCN (74.2%), TimesNet (73.6%), Crossformer (73.2%), and TimeMixer (73.1%). A key observation is that TCN-based methods achieve competitive classification performance. This is because temporal convolutions provide a strong inductive bias for extracting discriminative local motifs, e.g., shape-like patterns, that are often sufficient to determine class labels. With hierarchical convolutions, TCNs can further aggregate multi-scale evidence while remaining robust to small phase shifts and noise through parameter sharing and localized receptive fields, making them particularly effective for semantic recognition. In contrast, MLP-based approaches remain inferior (e.g., DLinear: 67.5%, RLinear: 70.0%), since their predominantly linear mixing is less capable of building high-level semantic features and modeling complex inter-series interactions. The superior performance of DeMa stems from its fine-grained and hierarchical dependency encoding: stacking DuoMNet blocks progressively aggregates multi-scale temporal evidence, while the dual-path design explicitly captures both temporal variations and cross-variate cues. In particular, Mamba-SSD strengthens long-range temporal structure modeling, and Mamba-DALA enhances cross-variate interactions, enabling DeMa to form more informative and semantically aligned representations for classification beyond purely convolutional local feature extraction.

4.4.1 Anomaly detection results

Anomaly detection aims to identify rare and abnormal patterns in time series, which often correspond to faults, critical events, or

outliers requiring timely intervention. Following prior work [40], we evaluate on five widely used anomaly detection benchmarks and report the average F1-score across datasets [69]. For fair comparison, we adopt reconstruction error [18] as the anomaly criterion for all methods. As shown in Fig. 3(b), DeMa achieves the best average F1-score of 88.2%, outperforming strong competitors such as ModernTCN (86.6%) and TimesNet (86.3%). This gain highlights the advantage of our fine-grained modeling. The proposed SSD+DALA design jointly strengthens (i) local temporal continuity and long-range temporal context via Mamba-SSD, and (ii) delay-aware cross-variate interactions via Mamba-DALA, enabling more faithful reconstruction of normal dynamics. As a result, abnormal deviations yield sharper and more separable reconstruction residuals, leading to improved detection accuracy. In contrast, iTransformer [13] and S-Mamba [64] perform notably worse, likely because prevalent normal patterns can dominate their series-wise global similarity modeling by attention; consequently, the subtle abnormal segments are more easily diluted when aggregating pairwise correlations, leading to inferior detection. Finally, we observe that methods benefiting from explicit decomposition cues (e.g., TimesNet and DeMa) tend to achieve stronger overall detection performance, underscoring the importance of separating stable periodic structures from irregular fluctuations so that violations of normal regularities become more detectable.

4.5 Performance–efficiency trade-off (RQ2)

Figure 4 summarizes the overall effectiveness and efficiency of DeMa. Figure 4(a) provides an overall comparison of baselines across five representative time-series tasks. Overall, DeMa achieves consistently strong results on all tasks, demonstrating robust task generality rather than tuning to a single forecasting setting. Figure 4(b) further evaluates the accuracy–efficiency trade-off on a large-scale traffic dataset with 862 variates under long-term forecasting, where the y -axis denotes MSE, the x -axis indicates training time, and the bubble size reflects GPU memory footprint. As shown, DeMa attains the lowest MSE (0.372) with moderate training time (232 ms/iter) and manageable memory usage (6.53 GB), offering a favorable balance between accuracy and computational cost. These results suggest that DeMa is a practical and scalable

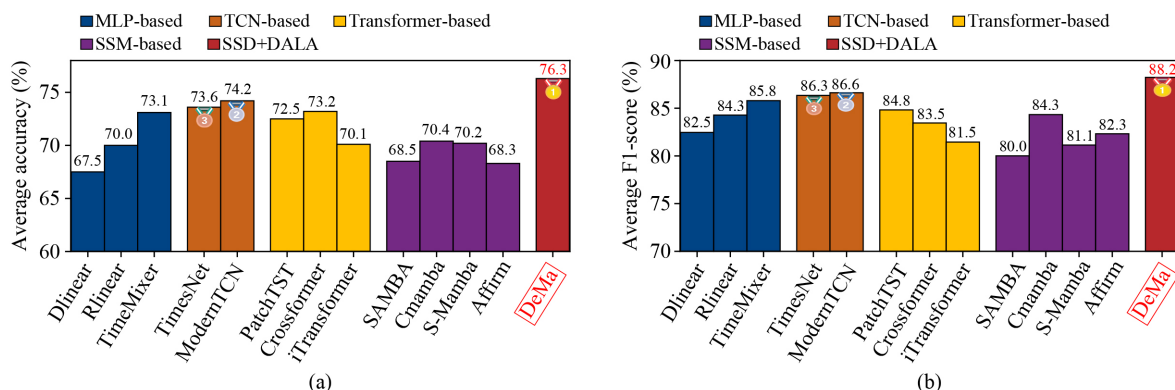


Fig. 3 Performance comparison on classification and anomaly detection tasks. Results are averaged over multiple datasets, and higher values indicate better performance. (a) Classification results (average accuracy); (b) anomaly detection results (average F1-score)

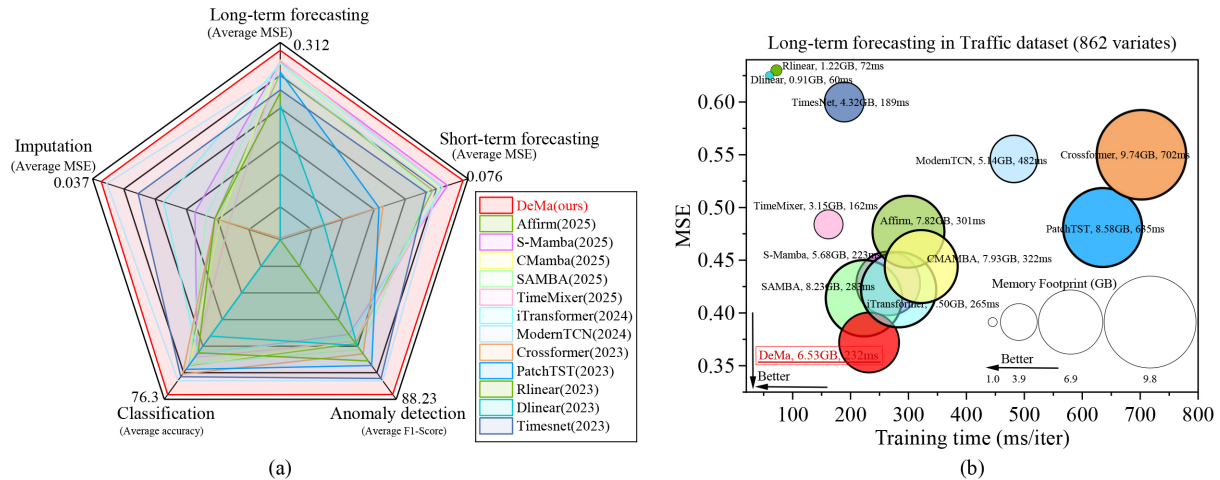


Fig. 4 Comparison of model performance and efficiency. DeMa achieves consistent state-of-the-art results across five mainstream time-series tasks. It also attains the lowest mean squared error while requiring less training time and GPU memory. (a) Overall performance across five tasks; (b) performance–efficiency trade-off

solution for large-scale multivariate time-series analysis.

DeMa exhibits clear efficiency advantages over Transformer-based baselines while remaining more accurate, benefiting from the linear-time state-space backbone. For instance, Crossformer and PatchTST require substantially higher training time and memory footprint (e.g., 702 ms/iter and 9.74 GB for Crossformer), yet still yield higher MSE than DeMa in Fig. 4(b). Moreover, DeMa consistently outperforms TCN-based models in the multi-task comparison (Fig. 4(a)). While TCNs are efficient and effective at capturing local motifs, their limited receptive field and forecasting-oriented inductive bias may limit transferability to tasks that require long-range aggregation. In contrast, DeMa learns more transferable representations by jointly encoding long-range temporal dynamics and delay-aware inter-series interactions. Finally, DeMa improves the trade-off between accuracy and efficiency compared to recent Mamba variants. As shown in Fig. 4(b), DeMa is faster and lighter than Affirm and CMamba (e.g., 232 ms/iter and 6.53 GB for DeMa versus 301 ms/iter and 7.82 GB for Affirm, and 322 ms/iter and 7.93 GB for CMamba), while achieving lower MSE.

In summary, DeMa combines state-of-the-art accuracy with practical training speed and memory efficiency, making it a strong candidate for scalable, general time-series analysis.

4.6 Scalability vs. input length (RQ3)

Semantic information in time series is often formed by aggregating evidence over long horizons rather than isolated timestamps. Although longer lookback windows provide richer context, they also place stricter demands on training efficiency and GPU memory. To evaluate scalability with respect to the input length, Fig. 5 reports the per-iteration running time and GPU memory footprint of DeMa and representative Transformer-based baselines on ETTh1. We increase the lookback length from {384, 768, 1536, 3072} while fixing the embedding dimension to $d = 256$ for a fair comparison.

As shown, DeMa consistently achieves the lowest running time and memory usage across all lengths, and its growth remains close to linear as the series length increases. By contrast, Transformer-based

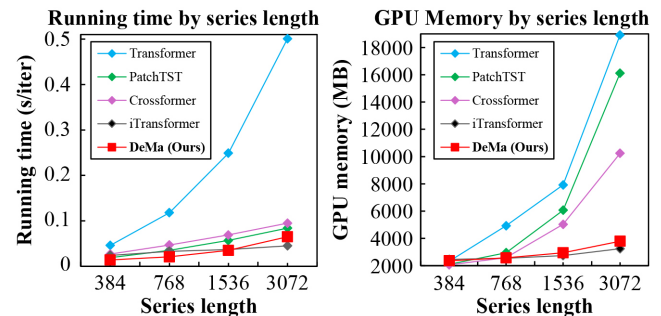


Fig. 5 Efficiency analysis of GPU memory and running time in a long-term lookback-window scenario. Our proposed DeMa scales linearly with the series length, whereas the vanilla Transformer exhibits quadratic complexity with respect to the series length

methods (e.g., Transformer, PatchTST, and Crossformer) exhibit rapidly increasing computation and memory overhead with longer inputs, which substantially limits their practicality in long-term settings. Notably, iTransformer is more efficient than token-wise Transformers because it computes series-wise attention, avoiding quadratic growth with respect to the token length; however, it remains consistently slower and more memory-intensive than DeMa at large input lengths. These results further explain why the advantages of DeMa become especially pronounced when the input context grows: the dual path preserves near-linear scaling with series length, whereas attention-based baselines incur substantially steeper computational and memory growth.

These observations align with our complexity analysis in Fig. 1(b). Leveraging the parallelizable Mamba-SSD and Mamba-DALA blocks, DeMa scales linearly with the token length, with time complexity $O(2NLD^2)$, where N is the number of variates, L is the number of tokens after tokenization, and D is the embedding dimension. In typical long-term MTS scenarios, $L, N \gg D$, making DeMa a more scalable backbone for long-term and large-scale multivariate time-series modeling.

■ 4.7 Ablation study (RQ4)

To quantify the contribution of each key component in DeMa, including the Adaptive Fourier Filter, Mamba-SSD, and Mamba-DALA, we conduct ablation studies on Traffic (long-term forecasting) and PEMS04 (short-term forecasting). As reported in Table 6, we consider two types of ablations: *replacement (Repl.)*, which substitutes a target module with an alternative design, and *removal (w/o)*, which disables the module entirely. Overall, each component consistently improves forecasting accuracy, and the full model yields the best performance on both datasets.

- **Adaptive fourier filter.** Comparing Rows 1 and 5, removing the decomposition module causes clear degradation on both datasets (Traffic MSE: $0.382 \rightarrow 0.415$; PEMS04 MSE: $0.061 \rightarrow 0.097$). This verifies that the Adaptive Fourier Filter provides a beneficial decomposition prior by separating slowly varying structures (e.g., trend and seasonality) from short-term fluctuations, thereby reducing spectral interference and facilitating more reliable dependency modeling in both the temporal and variate paths, consistent with decomposition-based forecasting literature [25,47].
- **Mamba-SSD for temporal modeling.** Rows 2–3 replace the temporal Mamba-SSD with attention-style alternatives, leading to substantial performance drops (e.g., Row 3: Traffic MSE $0.382 \rightarrow 0.507$, a 32.7% increase). This indicates that Mamba-SSD is critical for efficiently capturing long-range temporal dynamics. Moreover, removing the temporal path entirely (Row 7) results in the largest degradation (Traffic MSE $0.382 \rightarrow 0.597$, +56.3%; PEMS04 MSE $0.061 \rightarrow 0.225$), confirming that strong cross-time modeling is indispensable for accurate forecasting.
- **Mamba-DALA for cross-variate interaction modeling.** Rows 4 and 6 highlight the necessity of Mamba-DALA for modeling cross-variate dependencies. Replacing Mamba-DALA with a temporal-style Mamba-SSD interaction (Row 4) already hurts performance (Traffic MSE $0.382 \rightarrow 0.412$; PEMS04 MSE $0.061 \rightarrow 0.087$), suggesting that directly reusing temporal mechanisms is insufficient for variate interactions. More importantly, removing Mamba-DALA

(Row 6) causes a pronounced drop (Traffic MSE $0.382 \rightarrow 0.539$, +41.1%; PEMS04 MSE $0.061 \rightarrow 0.142$), verifying that delay-aware cross-variate modeling provides critical complementary cues beyond temporal dynamics.

Overall, the ablation results show that the main components of DeMa are functionally complementary rather than redundant. This supports that the structured design of DeMa is challenge-driven and empirically justified, rather than an unnecessarily complicated composition.

■ 4.8 Hyperparameter sensitivity analysis (RQ5)

In DeMa, we adopt a weighted fusion layer to combine the temporal-path representation and the variate-path representation, controlled by the fusion weights α and β in Eq. (29). To examine the sensitivity of this fusion trade-off, we perform a grid search over $\alpha, \beta \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$ on forecasting and imputation on Weather [25], classification on Heartbeat [62], and anomaly detection on MSL [59]. The results are summarized in Figs. 6(a)–6(d), where higher is better for Accuracy, F1-score, and 1/MAE.

Across all tasks, DeMa is most reliable when both paths remain active (i.e., neither α nor β is overly small), confirming that temporal modeling and cross-variate interaction are complementary rather than substitutable. Forecasting is relatively less sensitive to α and β (Fig. 6(a)), showing only mild variation across the grid. An explanation is that long-horizon prediction is often dominated by stable structures such as trend and seasonality; thus, once the temporal path is sufficiently emphasized, the model can still perform well even under moderate fusion bias, and the variate path provides a relatively smaller marginal gain unless the dataset is strongly coupled. Similarly, classification remains stable across a wide range of (α, β) (Fig. 6(b)). This is because classification primarily relies on global semantic discrimination: the stacked DuoMNet blocks can progressively aggregate high-level evidence and compensate for moderate fusion imbalance, as long as neither path is completely suppressed and both temporal and cross-variate cues are retained.

In contrast, imputation is more sensitive to the fusion weights (Fig. 6(c)), because recovering missing points typically requires both local temporal continuity (temporal path) and cross-variate

Table 6 Ablation study of DeMa

No.	Design	Decomp.	Variate Module	Temporal Module	Traffic		PEMS04	
					MSE	MAE	MSE	MAE
1	DeMa	AFF	Mamba-DALA	Mamba-SSD	0.382	0.247	0.061	0.168
2		AFF	Mamba-SSD	Mamba-DALA	0.416	0.297	0.093	0.204
3	Repl.	AFF	Mamba-DALA	Mamba-DALA	0.507	0.314	0.126	0.257
4		AFF	Mamba-SSD	Mamba-SSD	0.412	0.295	0.087	0.192
5		w/o	Mamba-DALA	Mamba-SSD	0.415	0.279	0.097	0.215
6	w/o	w/o	w/o	Mamba-SSD	0.539	0.317	0.142	0.267
7		w/o	Mamba-DALA	w/o	0.597	0.383	0.225	0.337

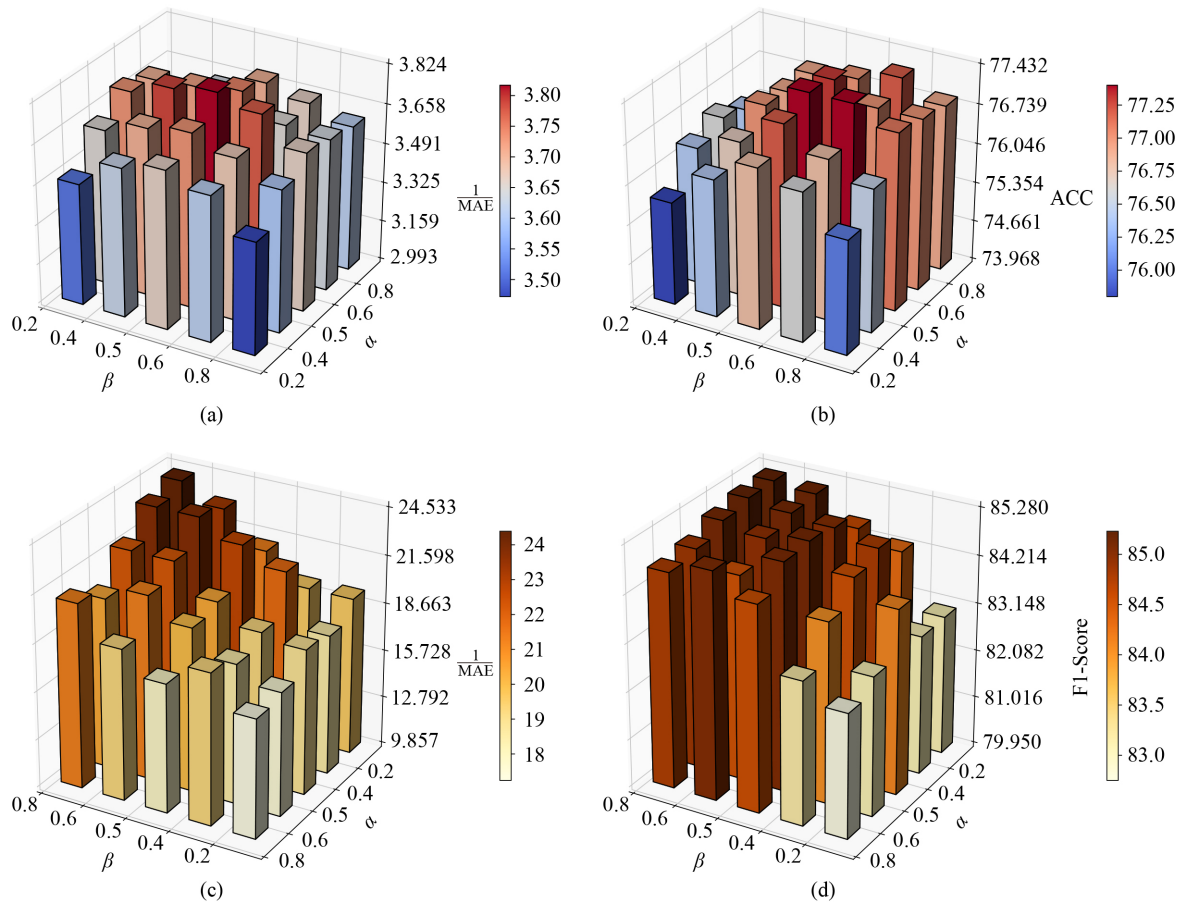


Fig. 6 Sensitivity of Fusion Weights α and β Across Tasks. (a) Weather forecasting: MAE vs. (α, β); (b) Heartbeat classification: accuracy vs. (α, β); (c) weather imputation: MAE vs. (α, β); (d) MSL anomaly detection: F1-score vs. (α, β)

complementarity (variate path). Over-weighting either path can lead to incomplete recovery, e.g., temporal-only fusion may miss informative cross-variate cues, while variate-only fusion may fail to preserve fine-grained local dynamics, especially under higher missing ratios. Similarly, anomaly detection exhibits strong sensitivity (Fig. 6(d)), since anomalies are rare, local patterns that demand temporal modeling to capture abrupt deviations, while cross-variate modeling helps validate abnormality via multi-sensor consistency; improper weighting may either dilute anomalies under dominant normal patterns or amplify spurious fluctuations, increasing false alarms. These observations suggest that α and β should be selected task-awarely, with balanced fusion particularly important for fine-grained, locality-critical tasks such as imputation and anomaly detection.

5 Related work

5.1 Transformer-based time series models

Transformers have become a dominant paradigm for multivariate time series (MTS) modeling due to their strong ability to capture pairwise dependencies with a global receptive field. Representative approaches include Informer [23], Fedformer [24], Autoformer [25], Pyraformer [26], PatchTST [10], Crossformer [27], and UniTime [70]. A fundamental limitation of these models lies in the quadratic complexity of self-attention with respect to the token length L , which

hinders scalability to long-horizon settings. To alleviate this issue, sparse or structured attention has been explored to reduce the computational burden (e.g., to $O(L \log L)$) [23,55,71]. Yet, it may overlook informative dependencies when only a subset of interactions is retained. More recently, iTransformer [13] reformulates MTS tokens by treating each variate as a token and applying attention over variates, shifting the quadratic term from L to the number of variates N . While efficient when $N \ll L$, this design may sacrifice fine-grained local temporal interactions that are crucial for locality-sensitive tasks.

5.2 MLP-based time series models

MLP-style architectures offer an attractive alternative for efficient time-series modeling by replacing attention with lightweight mixing operations. Classic representatives include DLinear [21] and its variants such as RLinear [67], which adopt decomposition-aware linear mapping strategies to achieve strong forecasting performance with low computational cost. TimeMixer [22] further enhances expressiveness via MLP-based mixing across temporal and channel dimensions. Despite their efficiency, purely MLP-based designs often rely on relatively limited inductive biases for explicitly modeling complex cross-variate interactions and fine-grained dependency structures, which can restrict their generality across diverse MTS tasks beyond forecasting.

■ 5.3 TCN-based time series models

Temporal convolutional networks (TCNs) model temporal dependencies via convolutional receptive fields and have been widely used in time series forecasting and representation learning. ModernTCN [20] strengthens classical TCNs with modern architectural practices to improve both accuracy and efficiency, while TimesNet [18] leverages convolutional modeling together with period-aware representations to capture multi-period temporal patterns. Although convolution provides strong locality and parallelism, TCN-based models typically require carefully designed receptive fields to capture long-range dependencies, and explicitly modeling cross-variate interactions remains non-trivial when scaling to high-dimensional MTS.

■ 5.4 Mamba-based time series models

State space models (SSMs), particularly Mamba-style selective SSMs, have recently emerged as a compelling linear-time alternative to Transformers, demonstrating strong performance across diverse domains [35]. This progress has spurred increasing interest in adapting Mamba to time series analysis [63–66,72,73]. For example, CMamba [65] augments Mamba-style temporal modeling with extra modules to capture cross-variate interactions, while S-Mamba [64], MambaMixer [72], MambaTS [73], and more recent variants such as Affirm [63] and SAMBA [66] explore architectural adaptations to improve effectiveness on time series benchmarks. Despite these advances, existing Mamba-based designs still face notable challenges for general MTS analysis. First, dependency entanglement is common: temporal dynamics and cross-variate interactions frequently overlap and are mixed, which can obscure task-relevant cues. Second, fine-grained cross-variate modeling remains insufficient: many methods primarily emphasize global interactions and may under-exploit local, delay-sensitive cross-variate effects. Motivated by these gaps, we propose **DeMa**, which disentangles cross-time and cross-variate dependency encodings via a dual-path architecture, enabling efficient yet effective modeling for general time-series analysis.

■ 6 Conclusion

In this work, we propose DeMa, a dual-path delay-aware Mamba backbone for efficient multivariate time-series analysis. DeMa explicitly decomposes the time-series context into *Cross-Time* and *Cross-Variate* components via an Adaptive Fourier Filter. These components are processed by stacked DuoMNet blocks with two parallel, scan-coupled pathways: (i) a temporal path that applies Cross-Time Scan and Mamba-SSD to capture long-range intra-series dependencies in a series-independent and parallel manner, and (ii) a variate path that applies Cross-Variate Scan and Mamba-DALA to model delay-aware inter-series interactions using DALA with both global correlation delays and token-level relative delays. The resulting representations are fused through a lightweight weighted fusion layer and projected to task-specific outputs. Extensive experiments across five mainstream tasks demonstrate that DeMa achieves consistently strong accuracy while substantially reducing training time and GPU memory usage, suggesting a favorable

accuracy-efficiency trade-off and practical scalability for long-horizon and large-scale MTS modeling.

■ Acknowledgements

The research described in this paper was partially supported by the National Natural Science Foundation of China (Grant Nos. 62433016 and 62537001), the General Research Funds from the Hong Kong Research Grants Council (PolyU 15207322, 15200023, 15206024, and 15224524), Hong Kong Research Grants Council's Theme-based Research Scheme (T43-513/23-N), Hong Kong Research Grants Council's Research Impact Fund (R1015-23), Hong Kong Research Grants Council's Collaborative Research Fund (C1043-24GF), Internal research funds from The Hong Kong Polytechnic University (P0059586, P0042693, P0048625, and P0051361), and Sheertek International (HK) Limited. This work was supported by computational resources provided by The Centre for Large AI Models (CLAIM) of The Hong Kong Polytechnic University.

■ Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

■ Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit website at creativecommons.org/licenses/by/4.0/.

■ References

- [1] Kong X, Chen Z, Liu W, Ning K, Zhang L, Marier S M, Liu Y, Chen Y, Xia F. Deep learning for time series forecasting: a survey. *International Journal of Machine Learning and Cybernetics*, 2025, 16(7–8): 5079–5112
- [2] Liang Z, An R, Fan W, Rao Y, Liang Y. iTFKAN: interpretable time series forecasting with kolmogorov-arnold network. 2025, arXiv preprint arXiv: 2504.16432
- [3] An R, Zhang Y, Liang Z, Fan W, Liang Y, Shang X, Li Q. Damba-ST: domain-adaptive mamba for efficient urban spatio-temporal prediction. 2025, arXiv preprint arXiv: 2506.18939
- [4] Huo G, Zhang Y, Wang B, Gao J, Hu Y, Yin B. Hierarchical spatio-temporal graph convolutional networks and transformer network for traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(4): 3855–3867

- [5] Fang C, Wang C. Time series data imputation: a survey on deep learning approaches. 2020, arXiv preprint arXiv: 2011.11347
- [6] Chen F, Qin Z, Zhou M, Zhang Y, Deng S, Fan L, Pang G, Wen Q. LARA: a light and anti-overfitting retraining approach for unsupervised time series anomaly detection. In: Proceedings of the ACM Web Conference 2024. 2024, 4138–4149
- [7] Blázquez-García A, Conde A, Mori U, Lozano J A. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys*, 2021, 54(3): Article 56
- [8] Qin Z, Zhang Y, Meng S, Qin Z, Choo K K R. Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion*, 2020, 53: 80–87
- [9] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 6000–6010
- [10] Nie Y, Nguyen N H, Sinthong P, Kalagnanam J. A time series is worth 64 words: long-term forecasting with transformers. In: Proceedings of the 11th International Conference on Learning Representations. 2023
- [11] Liu Y, Zhang H, Li C, Huang X, Wang J, Long M. Timer: generative pre-trained transformers are large time series models. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 32369–32399
- [12] Woo G, Liu C, Kumar A, Xiong C, Savarese S, Sahoo D. Unified training of universal time series forecasting transformers. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, 2024, 235: 53140–53164
- [13] Liu Y, Hu T, Zhang H, Wu H, Wang S, Ma L, Long M. iTransformer: inverted transformers are effective for time series forecasting. In: Proceedings of the 12th International Conference on Learning Representations. 2024
- [14] Han L, Ye H J, Zhan D C. The capacity and robustness trade-off: revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2024, 36(11): 7129–7142
- [15] Lin S, Lin W, Wu W, Zhao F, Mo R, Zhang H. SegRNN: segment recurrent neural network for long-term time-series forecasting. *IEEE Internet of Things Journal*, 2026, 13(5): 9861–9871
- [16] Hewamalage H, Bergmeir C, Bandara K. Recurrent neural networks for time series forecasting: current status and future directions. *International Journal of Forecasting*, 2021, 37(1): 388–427
- [17] Rangapuram S S, Seeger M, Gasthaus J, Stella L, Wang Y, Januschowski T. Deep state space models for time series forecasting. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018, 7796–7805
- [18] Wu H, Hu T, Liu Y, Zhou H, Wang J, Long M. TimesNet: temporal 2D-variation modeling for general time series analysis. In: Proceedings of the 11th International Conference on Learning Representations. 2023
- [19] Liu M, Zeng A, Chen M, Xu Z, Lai Q, Ma L, Xu Q. SCINet: time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 2022, 35: 5816–5828
- [20] Luo D, Wang X. ModernTCN: a modern pure convolution structure for general time series analysis. In: Proceedings of the 12th International Conference on Learning Representations. 2024
- [21] Zeng A, Chen M, Zhang L, Xu Q. Are transformers effective for time series forecasting? In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023, 11121–11128
- [22] Wang S, Wu H, Shi X, Hu T, Luo H, Ma L, Zhang J Y, Zhou J. TimeMixer: decomposable multiscale mixing for time series forecasting. In: Proceedings of the 12th International Conference on Learning Representations. 2024
- [23] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, Zhang W. Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. 2021, 11106–11115
- [24] Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R. FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 27268–27286
- [25] Wu H, Xu J, Wang J, Long M. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021, 22419–22430
- [26] Liu S, Yu H, Liao C, Li J, Lin W, Liu A X, Dustdar S. Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting. In: Proceedings of the 10th International Conference on Learning Representations. 2022
- [27] Zhang Y, Yan J. Crossformer: transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: Proceedings of the 11th International Conference on Learning Representations. 2023
- [28] Jia J, Gao J, Xue B, Wang J, Cai Q, Chen Q, Zhao X, Jiang P, Gai K. From principles to applications: a comprehensive survey of discrete tokenizers in generation, comprehension, recommendation, and information retrieval. 2025, arXiv preprint arXiv: 2502.12448
- [29] Qu H, Fan W, Zhao Z, Li Q. TokenRec: learning to tokenize ID for LLM-based generative recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 2025, 37(10): 6216–6231
- [30] Qu H, Lin S, Ding Y, Wang Y, Fan W. Diffusion generative recommendation with continuous tokens. In: Proceedings of the ACM Web Conference 2026. 2026, 7259–7270
- [31] Zhou Y, Qu H, Liu Y, Lin S, Song L, Fan W. HD-Prot: a protein language model for joint sequence-structure modeling with continuous structure tokens. 2025, arXiv preprint arXiv: 2512.15133
- [32] Ekambaram V, Jati A, Nguyen N, Sinthong P, Kalagnanam J. TSMixer: lightweight MLP-mixer model for multivariate time series forecasting. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 459–469
- [33] Gu A, Dao T. Mamba: linear-time sequence modeling with selective state spaces. 2023, arXiv preprint arXiv: 2312.00752
- [34] Dao T, Gu A. Transformers are SSMS: generalized models and efficient algorithms through structured state space duality. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 10041–10071
- [35] Qu H, Ning L, An R, Fan W, Derr T, Liu H, Xu X, Li Q. A survey of Mamba. *ACM Transactions on Intelligent Systems and Technology*,

2026. DOI: 10.1145/3819582

- [36] Lenz B, Lieber O, Arazi A, Bergman A, Manevich A, et al. Jamba: hybrid transformer-mamba language models. In: Proceedings of the 13th International Conference on Learning Representations. 2025
- [37] Zhu L, Liao B, Zhang Q, Wang X, Liu W, Wang X. Vision Mamba: efficient visual representation learning with bidirectional state space model. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, 2024, 235: 62429–62442
- [38] Schiff Y, Kao C H, Gokaslan A, Dao T, Gu A, Kuleshov V. Caduceus: bi-directional equivariant long-range DNA sequence modeling. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, 2024, 235: 43632–43648
- [39] Zhang Y, Qu H, Ning L, Fan W, Li Q. SSD4Rec: a structured state space duality model for efficient sequential recommendation. ACM Transactions on Information Systems, 2026, 44(2): 29
- [40] Wang Y, Wu H, Dong J, Liu Y, Wang C, Long M, Wang J. Deep time series models: a comprehensive survey and benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2026, volume(issue): 1–20
- [41] Zhao L, Shen Y. Rethinking channel dependence for multivariate time series forecasting: learning from leading indicators. In: Proceedings of the 12th International Conference on Learning Representations. 2024
- [42] Long Q, Fang Z, Fang C, Chen C, Wang P, Zhou Y. Unveiling delay effects in traffic forecasting: a perspective from spatial-temporal delay differential equations. In: Proceedings of the ACM Web Conference 2024. 2024, 1035–1044
- [43] Jiang J, Han C, Zhao W X, Wang J. PDFormer: propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023, 4365–4373
- [44] Gu A, Dao T, Ermon S, Rudra A, Ré C. HiPPO: recurrent memory with optimal polynomial projections. Advances in Neural Information Processing Systems, 2020, 33: 1474–1487
- [45] Harris M, Sengupta S, Owens J D. Parallel prefix sum (scan) with CUDA. In: Nguyen H, ed. GPU Gems 3. Chapter 39. Addison-Wesley Professional, 2007, 851–876
- [46] Yi K, Zhang Q, Fan W, Wang S, Wang P, He H, Lian D, An N, Cao L, Niu Z. Frequency-domain MLPs are more effective learners in time series forecasting. Advances in Neural Information Processing Systems, 2023, 36: 76656–76679
- [47] Liu Y, Li C, Wang J, Long M. Koopa: learning non-stationary time series dynamics with Koopman predictors. Advances in Neural Information Processing Systems, 2023, 36: 12271–12290
- [48] Yue Z, Wang Y, Duan J, Yang T, Huang C, Tong Y, Xu B. TS2Vec: towards universal representation of time series. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence. 2022, 8980–8987
- [49] Gu A, Johnson I, Timalina A, Rudra A, Ré C. How to train your HIPPO: state space models with generalized orthogonal basis projections. In: Proceedings of the 11th International Conference on Learning Representations. 2023
- [50] Kim T, Kim J, Tae Y, Park C, Choi J H, Choo J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In: Proceedings of the 10th International Conference on Learning Representations. 2022
- [51] Han D, Wang Z, Xia Z, Han Y, Pu Y, Ge C, Song J, Song S, Zheng B, Huang G. Demystify Mamba in vision: a linear attention perspective. Advances in Neural Information Processing Systems, 2024, 37: 127181–127203
- [52] Su J, Ahmed M, Lu Y, Pan S, Bo W, Liu Y. RoFormer: enhanced transformer with rotary position embedding. Neurocomputing, 2024, 568: 127063
- [53] Hertz D. Time delay estimation by combining efficient algorithms and generalized cross-correlation methods. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1986, 34(1): 1–7
- [54] Han D, Pan X, Han Y, Song S, Huang G. Flatten transformer: vision transformer using focused linear attention. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision. 2023, 5938–5948
- [55] Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang Y X, Yan X. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019, 5243–5253
- [56] Lai G, Chang W C, Yang Y, Liu H. Modeling long-and short-term temporal patterns with deep neural networks. In: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018, 95–104
- [57] Chen C, Petty K, Skabardonis A, Varaiya P, Jia Z. Freeway performance measurement system: mining loop detector data. Transportation Research Record: Journal of the Transportation Research Board, 2001, 1748(1): 96–102
- [58] Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, 2828–2837
- [59] Hundman K, Constantinou V, Laporte C, Colwell I, Soderstrom T. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018, 387–395
- [60] Mathur A P, Tippenhauer N O. SWaT: a water treatment testbed for research and training on ICS security. In: Proceedings of the 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks. 2016, 31–36
- [61] Abdulaal A, Liu Z, Lancewicki T. Practical approach to asynchronous multivariate time series anomaly detection and localization. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, 2485–2494
- [62] Bagnall A, Dau H A, Lines J, Flynn M, Large J, Bostrom A, Southam P, Keogh E. The UEA multivariate time series classification archive, 2018. 2018, arXiv preprint arXiv: 1811.00075
- [63] Wu Y, Meng X, Hu H, Zhang J, Dong Y, Lu D. Affirm: interactive mamba with adaptive Fourier filters for long-term time series forecasting. In: Proceedings of the 39th AAAI Conference on Artificial Intelligence. 2025, 21599–21607

- [64] Wang Z, Kong F, Feng S, Wang M, Yang X, Zhao H, Wang D, Zhang Y. Is Mamba effective for time series forecasting? *Neurocomputing*, 2025, 619: 129178
- [65] Zeng C, Liu Z, Zheng G, Kong L. CMamba: channel correlation enhanced state space models for multivariate time series forecasting. 2024, arXiv preprint arXiv: 2406.05316
- [66] Weng Z, Han J, Jiang W, Liu H. Simplified mamba with disentangled dependency encoding for long-term time series forecasting. 2024, arXiv preprint arXiv: 2408.12068v2
- [67] Li Z, Qi S, Li Y, Xu Z. Revisiting long-term time series forecasting: an investigation on linear mapping. 2023, arXiv preprint arXiv: 2305.10721
- [68] Kingma D P, Ba J. Adam: a method for stochastic optimization. In: *Proceedings of the 3rd International Conference on Learning Representations*. 2015
- [69] Kim S, Choi K, Choi H S, Lee B, Yoon S. Towards a rigorous evaluation of time-series anomaly detection. In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. 2022, 7194–7201
- [70] Liu X, Hu J, Li Y, Diao S, Liang Y, Hooi B, Zimmermann R. UniTime: a language-empowered unified model for cross-domain time series forecasting. In: *Proceedings of the ACM Web Conference 2024*. 2024, 4095–4106
- [71] Kitaev N, Kaiser L, Levskaya A. Reformer: the efficient transformer. In: *Proceedings of the 8th International Conference on Learning Representations*. 2020
- [72] Behrouz A, Santacatterina M, Zabih R. MambaMixer: efficient selective state space models with dual token and channel selection. 2024, arXiv preprint arXiv: 2403.19888
- [73] Cai X, Zhu Y, Wang X, Yao Y. MambaTS: improved selective state space models for long-term time series forecasting. 2024, arXiv preprint arXiv: 2405.16440



Rui AN is currently a PhD student of the Department of Computing (COMP), The Hong Kong Polytechnic University (PolyU), China and Northwestern Polytechnical University, China, under the supervision of Dr. Wenqi Fan, Prof. Qing Li, and Prof. Xuequn Shang. Her research interests include state-space models, time series analysis, and spatiotemporal foundation models. She has authored or coauthored several papers in top conferences and journals, such as ICDE and KBS.



Haohao QU is currently a PhD candidate in the Department of Computing (COMP) at The Hong Kong Polytechnic University (PolyU), China, under the supervision of Dr. Wenqi Fan and Prof. Qing Li. Before joining PolyU, he received both a Master's degree and a Bachelor's

degree from Sun Yat-sen University, China in 2022 and 2019, respectively. His research interests cover recommender systems and large language models. He has authored and coauthored around 20 innovative papers, most of which have been published in top journals and conferences, such as IEEE TKDE, ACM TOIS, ICLR, and KDD. For more information, please visit website at quhaoh233.github.io/ page.



Wenqi FAN is currently an assistant professor with the Department of Computing (COMP) and the Department of Management and Marketing (MM), The Hong Kong Polytechnic University (PolyU), China. His research interests include machine learning and data mining, with a particular focus on recommender systems, large language models (LLMs), and trustworthy AI. He has authored or coauthored innovative papers in top-tier journals and conferences, including TKDE, TIST, KDD, WWW, ICDE, NeurIPS, ICLR, SIGIR, IJCAI, AAAI, RecSys, and WSDM. He was a top-tier conference (Area/Senior) Program Committee member and session chair for ICML, ICLR, NeurIPS, KDD, WWW, AAAI, IJCAI, WSDM, EMNLP, and ACL, and a journal reviewer for TKDE, TIST, TKDD, TOIS, and TAI.



Xuequn SHANG is a professor in the School of Computer Science at Northwestern Polytechnical University, China. She received the PhD degree in Computer Science from Otto-von-Guericke-Universität Magdeburg, Germany in Oct. 2005. Her research interests include data mining, bioinformatics, educational data mining, and data management. She has published many academic papers in international journals, including *Nature Methods*, *Cell Reports*, and *Briefings in Bioinformatics*. She served many international journals as editor and reviewer, and served many international conferences as a committee member.



Qing LI (Fellow, IEEE) received his BEng. degree from Hunan University, China, and MSc and PhD degrees from the University of Southern California, USA, all in computer science. He is currently a Chair Professor and Head of the Department of Computing at The Hong Kong Polytechnic University, China. His research interests include multimodal data fusion and mining, machine learning, recommender systems, object modeling, social media computing, and e-learning systems. He is a Fellow of IEEE and IET, in addition to being a distinguished member of the CCF (Chinese Computer Federation). He is the chairperson of the Hong Kong Web Society and served as a steering committee member of DASFAA, ACM RecSys, IEEE U-Media, ICWL, and the WISE Society.