mathematics

MDPI

Article

TSC-Mamba: Adaptive Decomposition and Channel Interaction Fusion for Time Series Forecasting

Chenjie Zhao [1], Xiaobo Wang [2] and Ling Zhang [1,*]

[1] School of Automation, Central South University, Changsha 410083, China; 8207230115@csu.edu.cn
[2] School of Electronics and Information, Central South University, Changsha 410083, China; 8206230106@csu.edu.cn
* Correspondence: lingzhang2019@csu.edu.cn

Abstract

Multivariate time series forecasting (TSF) is a fundamental task in intelligent systems, yet accurate and efficient modeling remains challenging under high dimensionality, non-stationarity, and complex cross-variate dependencies. The Mamba architecture provides an efficient linear-time backbone, but it still suffers from a multivariate representational bottleneck caused by unified state modeling. To address this limitation, we propose TSC-Mamba, a Mamba-centered framework built on a "Decoupling and Specialization" paradigm and organized as a cohesive "Decompose–Propagate–Correlate" pipeline. Specifically, the Adaptive Decomposition Fusion Module separates predictable low-frequency trends from high-frequency residual dynamics, while the Channel Interaction Fusion Module explicitly models structured cross-variate dependencies through an efficient low-rank mechanism. Experiments on eight public benchmark datasets show that TSC-Mamba achieves an average error reduction of up to 3.5% over the direct baseline S-Mamba while strictly maintaining linear complexity. Ablation studies validate the effectiveness of both modules, and Wilcoxon signed-rank analysis further confirms that the gains over S-Mamba are statistically significant. Additional experiments indicate strong run-to-run stability, robustness to input-length variation, improved generalization under partially visible variates, and more concentrated empirical predictive bands than S-Mamba. These results show that structured responsibility allocation is an effective strategy for enhancing state-space models in multivariate TSF.

Keywords: time series forecasting; Mamba; adaptive decomposition fusion; channel interaction fusion

MSC: 62M10; 68T07

Check for updates

Academic Editor: Jonathan Blackledge

Received: 14 March 2026
Revised: 7 April 2026
Accepted: 15 April 2026
Published: 18 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Time series forecasting (TSF) is a fundamental task in intelligent sensing, energy management, traffic scheduling, environmental monitoring, and industrial control [1,2]. With the rapid development of the Internet of Things and multi-source sensor networks, real-world time series increasingly exhibit high dimensionality, multiple variables, non-stationarity, and strong noise [3,4]. Developing accurate, robust, and computationally efficient models for multivariate TSF in such dynamic environments remains a critical challenge for improving the autonomous decision-making and stable operation capabilities of intelligent systems.

Traditional statistical models are limited in their ability to capture nonlinear dependencies and multi-scale variations, whereas deep learning methods have substantially advanced temporal representation learning through end-to-end feature extraction [5,6]. Among them, Transformer-based models, such as Transformer [7], Informer [8], Autoformer [3], FEDformer [4], Reformer [9], Pyraformer [10], PatchTST [11], ETSformer [12], Non-stationary Transformer [13], and iTransformer [14], have achieved strong performance by modeling long-range dependencies with attention mechanisms. However, standard self-attention incurs considerable computational overhead in long-sequence and high-dimensional settings [11,15]. In parallel, lightweight alternatives, including DLinear [5], TiDE [16], TSMixer [6], FreTS [17], and TimeMixer [15], improve efficiency through linear mapping, MLP mixing, or frequency-domain modeling, yet they often do not jointly address adaptive non-stationarity and explicit multivariate interaction modeling.

Against this backdrop, State Space Models (SSMs) have attracted increasing attention, especially Mamba [18], which provides linear time complexity $O(D \cdot L)$ together with strong long-range sequence modeling ability [19,20]. These properties make Mamba an appealing backbone for efficient TSF. Nevertheless, when applied to multivariate forecasting, the standard Mamba architecture still suffers from a structural limitation: it relies on a unified latent state to simultaneously encode heterogeneous dynamics. In practice, this single state is expected to preserve slowly varying low-frequency trends, respond to rapidly changing high-frequency fluctuations, and implicitly carry cross-variate dependencies at the same time. Such entangled responsibilities restrict the model's capacity to specialize and may lead to two concrete failure modes: the model may become overly smooth in trend-dominant regions, or overly sensitive to short-term perturbations and local noise, while cross-channel dependencies remain insufficiently modeled. This ultimately gives rise to a multivariate representational bottleneck.

Recent Mamba-based forecasting studies have partially improved this situation from different perspectives. S-Mamba [21] captures bidirectional dependencies with low computational overhead, but it lacks explicit trend modeling mechanisms. DecMamba [22] improves forecasting mainly from a decomposition-oriented perspective by introducing adaptive decomposition and separate modeling strategies for trend and seasonal components. CMMamba [23] improves forecasting mainly from a correlation-guided channel mixing perspective, where correlated channels are selected and fused to enhance multivariate feature representation. In addition, FMamba [24], SOR-Mamba [25], MambaTS [26], ms-Mamba [27], ss-Mamba [28], Mamba4Cast [29], and UncMamba [30] further extend Mamba-based forecasting toward fast attention augmentation, order robustness, multi-scale modeling, zero-shot transfer, and uncertainty quantification. Very recent studies continue this trend. DMamba [31] further strengthens decomposition-enhanced Mamba forecasting under complex non-stationary temporal patterns, while ASGMamba [32] introduces adaptive spectral gating to suppress noisy high-frequency interference and improve robust sequence modeling. These studies confirm the flexibility and potential of the Mamba family, but they still mainly strengthen one aspect of the forecasting problem at a time.

Meanwhile, recent work also reveals a broader hybrid trend in efficient long-sequence modeling. In the TSF domain, SST [33] combines Mamba and Transformer experts to capture long-range patterns and short-range variations, while MoU [34] hierarchically integrates Mamba, convolution, and self-attention from a hybrid perspective. Beyond TSF, InfoMamba [35] and Jamba [36] further exemplify the growing trend of hybrid Mamba-based design for balancing efficiency and representational flexibility in long-context sequence modeling. These developments suggest that combining Mamba-style state-space modeling with complementary mechanisms has become an important direction for improving expressive capacity without giving up efficiency advantages.

However, from our perspective, the key novelty of the present work is not merely the integration of a decomposition module and a channel interaction module. Rather, we argue that the fundamental limitation of existing Mamba-based multivariate forecasting lies in the representational conflict caused by unified state modeling itself. Existing methods usually strengthen decomposition, channel mixing, robustness, multi-scale processing, or hybrid expert fusion separately, without explicitly assigning low-frequency trend modeling, high-frequency residual propagation, and cross-channel interaction to specialized yet coordinated components within a unified Mamba-centered framework. Therefore, the central research gap is the lack of a coherent Mamba-based forecasting architecture that can resolve this multivariate representational bottleneck while preserving linear complexity. This gap directly motivates the present study.

A concise synthesis of representative Mamba-based forecasting variants is summarized in Table 1.

Table 1. Comparative synthesis of representative Mamba-based forecasting variants and the proposed TSC-Mamba.

Model	Main Emphasis	Limitation Relative to TSC-Mamba
S-Mamba [21]	Bidirectional dependency modeling with an efficient Mamba backbone	Lacks explicit trend separation and structured cross-channel interaction modeling
DecMamba [22]	Decomposition-oriented modeling of trend and seasonal components	Strengthens decomposition but does not explicitly organize cross-channel interaction as a coordinated component
CMMamba [23]	Correlation-guided channel mixing for multivariate feature fusion	Strengthens channel interaction but lacks explicit trend separation and residual specialization
SST [33]	Hybrid Mamba–Transformer experts for long-/short-range forecasting	Improves performance through hybrid expert fusion but does not directly resolve the internal representational conflict of unified Mamba state modeling
MoU [34]	Hybrid Mamba / convolution / self-attention integration	Provides hybrid architectural enhancement but not a Mamba-centered responsibility allocation for trend, residual dynamics, and cross-channel interaction
TSC-Mamba (Ours)	Unified "Decompose – Propagate – Correlate" pipeline with structured responsibility allocation	Explicitly assigns trend extraction, residual temporal propagation, and cross-channel interaction to specialized yet coordinated components while preserving linear complexity

To address the above challenges, this paper proposes an improved state-space model, Time-Series Channel-Decomposition Mamba (TSC-Mamba). TSC-Mamba is built on a "Decoupling and Specialization" design philosophy and establishes a cohesive "Decompose–Propagate–Correlate" modeling pipeline. Specifically, it integrates two dedicated modules into the Mamba backbone: (1) an Adaptive Decomposition Fusion Module (ADFM), which isolates predictable low-frequency trend components and thereby explicitly unburdens the Mamba Core so that it can focus on modeling high-frequency residual dynamics; and (2) a Channel Interaction Fusion Module (CIFM), which explicitly captures cross-variate dependencies through an efficient low-rank fusion strategy. In this way, TSC-Mamba does

not simply attach auxiliary modules to Mamba. Instead, it provides a structured allocation of modeling responsibilities: ADFM handles trend separation, the Mamba Core specializes in temporal propagation of residual dynamics, and CIFM restores structured cross-channel interactions that are not naturally modeled by the per-channel SSM mechanism.

The key contributions of this work are summarized as follows:

1. We propose TSC-Mamba, a unified Mamba-based framework for multivariate TSF that addresses the representational bottleneck of unified state modeling through a "Decoupling and Specialization" perspective.
2. We design the ADFM to explicitly separate predictable low-frequency trends from high-frequency residual dynamics, and the CIFM to efficiently model dynamic cross-channel dependencies via low-rank fusion, thereby forming a coherent "Decompose–Propagate–Correlate" pipeline.
3. We conduct extensive experiments on eight public benchmark datasets, and the results demonstrate that TSC-Mamba consistently outperforms representative baselines in predictive accuracy, while also exhibiting strong computational efficiency, stability, and generalization in additional analyses.

Overall, TSC-Mamba extends the expressive power of state-space models for multivariate TSF and provides an efficient, scalable, and structurally interpretable solution for modeling complex multi-source sensor data [1,2].

2. Proposed Method

2.1. Overall Architecture

The overall architecture of TSC-Mamba is shown in Figure 1. This architecture implements our proposed framework of decoupling and specialization, following a cohesive "Decompose–Propagate–Correlate" modeling pipeline to address the multifaceted challenges of TSF.

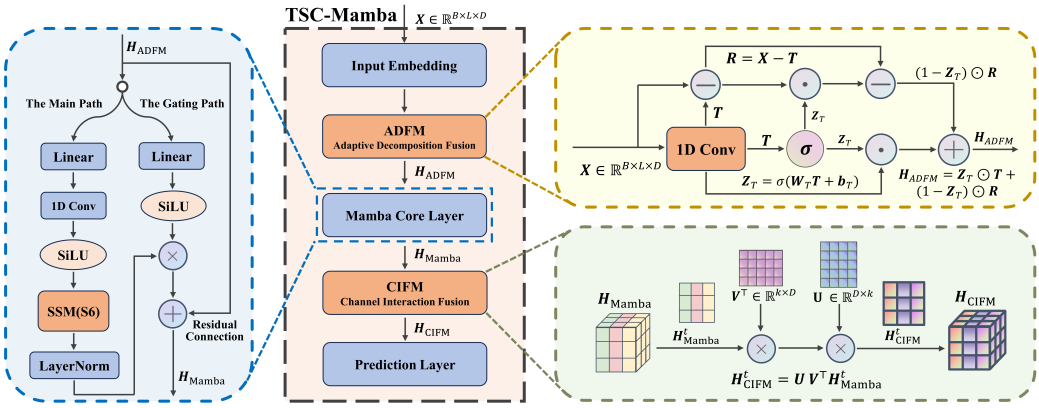


Figure 1. The overall architecture of TSC-Mamba. (**Center**) Main five-stage pipeline. (**Left**) Mamba Core Layer structure. (**Top-Right**) ADFM mechanism. (**Bottom-Right**) CIFM's efficient $O(D \cdot L)$ low-rank fusion.

The input $X \in \mathbb{R}^{B\times L\times D}$, where B denotes the batch size, L the sequence length, and D the number of variables, is first mapped to a latent space representation via an embedding layer. The subsequent processing unfolds in three specialized stages:

1. **Decompose**: The embedded features pass through the ADFM, which isolates the predictable low-frequency trend components. This initial step mitigates non-stationarity and allows the subsequent layers to focus on more complex patterns.
2. **Propagate**: The Mamba Core then specializes its modeling capacity on the high-frequency residual dynamics. By leveraging selective state-space modeling, it effec-

tively propagates temporal information to capture long-range dependencies with linear complexity.

3. **Correlate**: Finally, the CIFM intervenes to supplement the representation by explicitly modeling the cross-variate interactions via a low-rank fusion strategy, thereby resolving the core's inherent representational bottleneck.

The final fused representation is then fed into a prediction head to generate the forecast. This modular design ensures that each component specializes in a distinct dimension of the time-series dynamics while maintaining an efficient global data flow. From this viewpoint, the proposed pipeline is not only an architectural arrangement but also a structured allocation of modeling responsibility: ADFM separates predictable trend information, the Mamba Core specializes in residual temporal propagation, and CIFM explicitly supplements the cross-channel interactions that are not naturally modeled by the per-channel SSM mechanism.

2.2. Adaptive Decomposition Fusion Module (ADFM)

To facilitate the specialization of the Mamba Core, we first employ the ADFM, detailed in Figure 1 (Top-Right), to isolate trend components and mitigate non-stationarity [22]. Given an input X, it is decomposed into a low-frequency trend component T and a high-frequency residual component R:

$$T = \text{Conv1D}_{\theta_T}(X), \quad R = X - T, \tag{1}$$

where Conv1D_{θ_T} is a 1D convolution with learnable weights θ_T.

We adopt a 1D convolution for trend decomposition because it serves as a lightweight and learnable temporal smoothing operator that is well aligned with the role of ADFM in our framework. In ADFM, the objective is not to perform an exhaustive decomposition of all temporal components but to isolate the more predictable low-frequency trend part so that the subsequent Mamba Core can focus on modeling the high-frequency residual dynamics. Compared with moving-average-based decomposition commonly used in prior forecasting models, a learnable 1D convolution avoids fixed and uniformly weighted kernels, and can adapt its temporal filtering behavior to different datasets during end-to-end training. Compared with frequency-domain decomposition strategies, it does not rely on additional transform-domain assumptions and remains straightforward to integrate into the original temporal pipeline. Compared with more complex adaptive decomposition techniques such as variable-level kernel search, the 1D convolution offers a simpler and more computationally efficient design, which is consistent with our goal of preserving the lightweight and linear-complexity nature of TSC-Mamba. Therefore, our choice of 1D convolution reflects a trade-off among adaptability, efficiency, and architectural compatibility, rather than a claim that it is universally superior to all other decomposition techniques.

To dynamically recombine these components for stable representation, we introduce a Gated Fusion Mechanism:

$$Z_T = \sigma(W_T T + b_T), \tag{2}$$

$$H_{\text{ADFM}} = Z_T \odot T + (1 - Z_T) \odot R, \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function, W_T and b_T are learnable weight and bias parameters, and Z_T is a gate that adaptively adjusts the fusion ratio. In Equation (3), $\odot$ denotes element-wise multiplication. This effective isolation of the trend component facilitates the specialization of the subsequent Mamba Core, allowing it to better distinguish and specialize on the more complex, high-frequency residual dynamics.

2.3. Mamba Core and Specialization

After the ADFM isolates the trend, the specialized Mamba Core, shown in Figure 1 (Left), models the temporal dependencies of the resulting residual component [18]. The core of Mamba is a selective SSM, defined by

$$x_{t+1} = A_t x_t + B_t u_t, \quad y_t = C x_t, \tag{4}$$

where u_t is the input from ADFM, x_t is the hidden state, y_t is the output, and C is the observation matrix. The matrices A_t and B_t are made input dependent (selective) via

$$A_t = g_\alpha(u_t)A, \quad B_t = g_\beta(u_t)B, \tag{5}$$

where g_α and g_β are gating functions, A is the base state transition matrix, and B is the base input matrix. This selective mechanism allows the model to capture long-range dependencies with linear complexity [21].

As illustrated in the Mamba Core Layer detail, this selective SSM is embedded within a complete block structure. The input u_t is processed through two parallel paths: the main path and the gating path. The main path applies a linear projection, a 1D convolution, a Sigmoid-weighted Linear Unit (SiLU) activation, and the selective SSM, followed by a Layer Normalization (LayerNorm). Concurrently, the gating path computes a modulation gate using its own linear projection and SiLU. The Layer-Normalized output of the main path is then multiplicatively modulated by the output of the gating path, and this result is added to the input u_t via a residual connection to produce H_{Mamba}. More specifically, the two branches play complementary roles in information filtering. In the main path, the linear projection and the subsequent 1D convolution first perform local temporal mixing, while the SiLU activation introduces input-dependent nonlinearity before the sequence is processed by the selective SSM. This design allows the selective SSM to operate on a feature space that has already been smoothed and nonlinearly reweighted so that informative temporal patterns can be propagated while irrelevant local perturbations are suppressed. In the gating path, the separate linear projection followed by SiLU produces an adaptive modulation signal conditioned on the same input u_t. This signal does not independently generate temporal features; instead, it acts as an input-dependent gate that rescales the normalized output of the main path through element-wise multiplication. As a result, the main path is responsible for temporal propagation and state updating, whereas the gating path controls how strongly the propagated features are exposed at the output. The final residual connection further preserves the original input information and stabilizes optimization so that the block can selectively enhance useful temporal responses without overwriting the input representation.

While effective at capturing temporal patterns, the core SSM structure is inherently per-channel. It models each variable's state evolution independently, thus lacking a built-in mechanism to model cross-variate dependencies.

2.4. Channel Interaction Fusion Module (CIFM)

To address the inherent per-channel limitation of the Mamba Core, we propose the CIFM, shown in Figure 1 (Bottom-Right), to supplement its representation by efficiently fusing cross-variable information.

Our use of a low-rank channel interaction matrix should be understood as a principled approximation rather than an exact structural claim. From a linear-algebraic perspective, for any interaction matrix $M \in \mathbb{R}^{D\times D}$, the Eckart–Young–Mirsky theorem states that the

best rank-k approximation under the Frobenius norm is given by the truncated singular value decomposition [37,38]. Specifically, if

$$M = \widetilde{U}\Sigma\widetilde{V}^\top, \tag{6}$$

where $\widetilde{U}_k$, Σ_k, and $\widetilde{V}_k$ denote the truncated factors associated with the top-k singular values, then the optimal rank-k approximation is

$$M_k = \widetilde{U}_k\Sigma_k\widetilde{V}_k^\top = \arg\min_{\mathrm{rank}(\hat{M})\le k} \|M - \hat{M}\|_F, \tag{7}$$

which yields

$$\|M - M_k\|_F = \min_{\mathrm{rank}(\hat{M})\le k} \|M - \hat{M}\|_F. \tag{8}$$

This implies that, even when the true cross-channel interaction matrix is not exactly low-rank, its dominant interaction patterns can still be captured by an optimal compact low-rank approximation. Therefore, if the major cross-channel dependencies are concentrated in a limited number of principal interaction modes, a low-rank parameterization can preserve the most informative structure while significantly reducing computational complexity and the risk of overfitting. Related multivariate forecasting studies have also shown that low-rank or factorized latent structures can be effective for modeling high-dimensional time series, including Temporal Regularized Matrix Factorization [39], Temporal Latent Auto-Encoder [40], and Low Rank Forecasting [41], while recent channel-aware low-rank adaptation methods further support the usefulness of compact low-rank parameterizations in time series forecasting [42]. Motivated by this theoretical and empirical background, we implement CIFM using an efficient learnable low-rank mechanism, which acts as an implicit regularizer to capture the dominant and discriminative channel correlations.

This mechanism utilizes two learnable matrices, $U \in \mathbb{R}^{D\times k}$ and $V \in \mathbb{R}^{D\times k}$, where k is the rank (a small hyperparameter, $k \ll D$). We initialize U and V using Xavier uniform initialization and train them end-to-end. Given the feature representation H^t_{Mamba} from the Mamba Core at time t, the output H^t_{CIFM} is computed by applying the implicit channel correlation matrix $\Phi = UV^\top$ (where $\top$ denotes transpose):

$$H^t_{\text{CIFM}} = \Phi H^t_{\text{Mamba}}. \tag{9}$$

This operation is implemented efficiently by avoiding the explicit computation of the $O(D^2)$ matrix Φ. By reordering the computation as $H^t_{\text{CIFM}} = U(V^\top H^t_{\text{Mamba}})$ using matrix associativity, the total complexity is kept linear at $O(D \cdot L)$ with respect to the channel dimension D, making the module highly efficient.

More explicitly, to analyze the complexity over the full temporal dimension, we collect the channel features across all L time steps into a matrix $H_{\text{Mamba}} \in \mathbb{R}^{D\times L}$, whose t-th column corresponds to H^t_{Mamba}. Then the CIFM operation can be written in matrix form as

$$H_{\text{CIFM}} = \Phi H_{\text{Mamba}} = (UV^\top)H_{\text{Mamba}} = U\Big(V^\top H_{\text{Mamba}}\Big). \tag{10}$$

This equality is exact and follows directly from matrix associativity, so the reordered computation does not alter the resulting representation under the same low-rank parameterization. In other words, the representational capacity is determined by the rank-k structure of $\Phi = UV^\top$ itself rather than by the reordering operation. If one explicitly forms $\Phi \in \mathbb{R}^{D\times D}$ and then multiplies it by H_{Mamba}, the cost is dominated by $O(D^2L)$. In contrast, in the reordered form, the intermediate term $V^\top H_{\text{Mamba}} \in \mathbb{R}^{k\times L}$ is computed first with

complexity $O(kDL)$, and the subsequent multiplication by $U \in \mathbb{R}^{D\times k}$ also costs $O(kDL)$. Therefore, the total complexity becomes

$$O(kDL) + O(kDL) = O(2kDL). \tag{11}$$

Since $k \ll D$ and is treated as a small hyperparameter, this complexity remains linear with respect to both the channel dimension and the sequence length, i.e.,

$$O(2kDL) \approx O(D \cdot L). \tag{12}$$

Combined with the preceding Eckart–Young–Mirsky analysis, this shows that the reordered low-rank fusion preserves the dominant cross-channel interaction structure while strictly maintaining linear complexity.

3. Results

This section systematically evaluates the performance of the proposed TSC-Mamba model on eight widely used public multivariate TSF benchmark datasets. First, we will detail the experimental setup in Section 3.1, including datasets, evaluation metrics, comparative baselines, and implementation details. Subsequently, we will present the performance comparison results of TSC-Mamba against mainstream methods, ablation studies of key modules, and an in-depth analysis of its computational efficiency, stability, and generalization capability.

3.1. Experimental Setup

This section details the experimental setup used to evaluate the performance of the TSC-Mamba model, including the datasets, baseline models, evaluation metrics, and specific implementation details.

- Datasets. We selected eight widely used public multivariate TSF benchmark datasets, covering multiple domains such as electricity, transportation, weather, and finance. These datasets include: ETTh1 [8], ETTh2 [8], ETTm1 [8], ETTm2 [8] (Electricity Transformer Temperature), Electricity [8] (customer electricity consumption), Weather [8] (meteorological indicators), Traffic [3] (road traffic occupancy), and Exchange [21] (daily exchange rates). Detailed statistics for each dataset are summarized in Table 2. Following standard data processing procedures [3,8], all datasets were zero-mean standardized. The ETT datasets are partitioned using a 6:2:2 ratio, and all other datasets use a 7:1:2 ratio for training, validation, and testing.

Table 2. Statistical summary of the eight benchmark datasets.

Dataset	Category	Variates	Timesteps	Sampling Interval
ETTh1	Electricity	7	17,420	1 h
ETTh2	Electricity	7	17,420	1 h
ETTm1	Electricity	7	69,680	15 min
ETTm2	Electricity	7	69,680	15 min
Electricity	Electricity	321	26,304	1 h
Weather	Weather	21	52,696	10 min
Traffic	Traffic	862	17,544	1 h
Exchange	Finance	8	7588	1 day

- Baseline Models. To comprehensively validate the effectiveness of TSC-Mamba, we selected 11 representative mainstream baseline models for comparison, covering various advanced technical routes in the current TSF field: (1) Transformer based: Informer [8], Autoformer [3], FEDformer [4], PatchTST [11], and iTransformer [14];

(2) Linear/MLP based: DLinear [5] and TSMixer [6]; (3) Convolutional Neural Network (CNN) based: TimesNet [43]; and (4) SSM based: S-Mamba [21], DecMamba [22], and CMMamba [23].

- Evaluation Metrics. Following widely adopted evaluation standards [3,5,8], we use Mean Absolute Error (MAE) and Mean Squared Error (MSE) as the primary performance metrics:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{13}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

To evaluate the model's long-term forecasting capability, we report the performance of all models across multiple prediction lengths $H \in \{96, 192, 336, 720\}$.

- Implementation Details. All models were implemented in PyTorch 2.5.1 [17,23] and trained on NVIDIA A100 GPUs [21,22]. To ensure a fair comparison, all methods were evaluated under the same experimental protocol, including the same datasets, data splits, forecasting horizons $H \in \{96, 192, 336, 720\}$, evaluation metrics, hardware platform, and repetition strategy. Moreover, for all training settings and hyperparameters that are shared and directly comparable across models, we adopted the same configuration for all methods, including the AdamW optimizer [8,43], a cosine annealing schedule [8,43], weight decay 10^{-2} [14,21], and batch size 32 [4,44], among others. For TSC-Mamba, the key hyperparameters, including the input sequence length L, the initial learning rate (LR), and the CIFM rank k, were selected by grid search, with detailed sensitivity analysis provided in Appendix A. Based on the quantitative results, we adopt $L = 192$, an initial LR of 10^{-4}, and $k = 16$ as the default setting. The initial LR candidates were selected over a logarithmic grid, and the default value 10^{-4} was chosen because it consistently provided the best accuracy–stability trade-off across representative datasets. The cosine annealing schedule was adopted to allow relatively larger updates in the early training stage for broad parameter exploration, while gradually reducing the step size in later epochs to stabilize the joint optimization of the ADFM and the Mamba Core, thereby mitigating oscillation and reducing the risk of convergence to poor local solutions. For baseline models, only architecture-specific hyperparameters that are unique to a particular model and not directly comparable across heterogeneous methods were taken from their reported optimal settings or official public codebases. All experiments were independently repeated five times, and the average results are reported. For the additional deployment-oriented benchmarking, wall-clock inference time was measured after 20 warm-up forward passes and then averaged over 100 repeated forward passes using the same trained checkpoint, while peak GPU memory usage was recorded by torch.cuda.max_memory_allocated().

3.2. Overall Performance Comparison

To validate the overall performance of the TSC-Mamba model, we conducted extensive comparative experiments against 11 representative mainstream baseline models on eight public benchmark datasets. The evaluation was conducted on all datasets for four different prediction lengths $H \in \{96, 192, 336, 720\}$ to assess the model's long-term forecasting capability.

Table 3 and its continuation table show the detailed MAE and MSE results for all models across the eight datasets and four prediction lengths. As seen from the tables, TSC-Mamba achieved the best or second-best performance in the vast majority of dataset and prediction length settings. Particularly on the ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2) and

the high-dimensional Electricity and Traffic datasets, TSC-Mamba demonstrated a consistent advantage over other Mamba variants (S-Mamba, DecMamba, and CMMamba).

A comprehensive analysis averaging the results across all datasets and prediction lengths reveals that TSC-Mamba reduces the average MAE and MSE by approximately 2.5% to 3.5% compared to the baseline S-Mamba [21]. This significant and consistent performance improvement strongly validates the efficacy of our proposed decoupling and specialization strategy: the ADFM isolates low-frequency trends, allowing the Mamba Core to specialize on high-frequency residuals, while the low-rank CIFM explicitly captures dynamic cross-variate correlations to supplement the core's per-channel limitation. Meanwhile, although models such as iTransformer [14] and DLinear [5] exhibit strong competitiveness on specific datasets (e.g., Exchange), TSC-Mamba's overall performance still surpasses most SOTA models, including PatchTST [11] and TimesNet [43], achieving a compelling trade-off between accuracy and efficiency by resolving Mamba's multivariate representational bottleneck.

Figure 2 further visualizes the forecasting results of (a) TSC-Mamba and (b) S-Mamba on all eight benchmark datasets (ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Weather, Traffic, and Exchange). It can be observed that TSC-Mamba's prediction curves align more closely with the ground truth values. In contrast, S-Mamba's predictions show noticeable deviations at multiple data points. Particularly in the high-frequency periodic fluctuations of Electricity and Traffic, as well as in the trend-varying regions of Weather and Exchange, TSC-Mamba demonstrates a superior fitting capability and higher precision in capturing dynamic changes compared to S-Mamba. This demonstrates the superiority of the TSC-Mamba architecture.

Table 3. Performance comparison of TSC-Mamba with Mamba variants, advanced Transformer models and traditional baselines (MSE/MAE). Red indicates the best performance, and blue indicates the second best.

Dataset	H	TSC-Mamba		S-Mamba		DecMamba		CMMamba		PatchTST		iTransformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.374**	**0.393**	0.386	0.405	0.382	0.404	0.385	0.407	0.414	0.419	**0.379**	**0.402**
	192	**0.427**	**0.425**	0.441	0.437	0.435	0.433	0.438	0.436	0.460	0.445	**0.431**	**0.429**
	336	**0.472**	**0.445**	0.487	0.458	0.479	0.456	0.481	0.459	0.501	0.466	0.475	**0.451**
	720	**0.488**	**0.478**	0.503	0.491	**0.495**	**0.488**	0.499	0.490	0.500	0.489	0.508	0.493
ETTh2	96	**0.288**	**0.339**	0.297	0.349	**0.291**	0.346	0.294	**0.343**	0.302	0.348	0.301	0.350
	192	**0.368**	**0.388**	0.380	0.400	0.377	0.397	**0.374**	**0.395**	0.388	0.400	0.383	0.398
	336	**0.415**	**0.419**	0.428	0.432	0.421	0.426	**0.418**	**0.423**	0.426	0.433	0.433	0.424
	720	**0.414**	**0.432**	0.427	0.445	**0.422**	0.441	0.425	**0.440**	0.431	0.446	0.432	0.447
ETTm1	96	**0.324**	**0.357**	0.334	0.368	0.329	0.366	**0.326**	**0.364**	0.329	0.367	0.341	0.376
	192	**0.365**	**0.379**	0.377	0.391	0.371	0.388	0.369	**0.385**	**0.367**	0.386	0.383	0.395
	336	**0.413**	**0.408**	0.426	0.420	0.419	0.415	0.417	0.413	**0.412**	**0.410**	0.418	0.418
	720	0.476	**0.446**	0.491	0.459	0.481	0.451	0.485	0.453	**0.472**	**0.442**	0.488	0.457
ETTm2	96	**0.174**	**0.257**	0.180	0.264	0.177	0.261	0.178	0.263	**0.175**	**0.255**	0.186	0.272
	192	**0.242**	**0.300**	0.250	0.309	0.246	0.306	0.248	0.307	**0.241**	**0.302**	0.254	0.314
	336	**0.301**	**0.338**	0.311	0.348	0.305	0.345	0.307	0.346	**0.303**	**0.343**	0.316	0.351
	720	**0.399**	**0.395**	0.412	0.407	0.405	0.401	0.408	0.403	**0.402**	**0.400**	0.414	0.407
Electricity	96	**0.143**	**0.233**	0.148	0.240	0.145	0.238	**0.144**	**0.236**	0.181	0.270	0.168	0.272
	192	**0.157**	**0.246**	0.162	0.253	**0.159**	0.253	0.160	**0.251**	0.188	0.274	0.184	0.289
	336	**0.172**	**0.261**	0.178	0.269	**0.174**	0.270	0.175	**0.267**	0.204	0.293	0.198	0.300
	720	**0.218**	**0.308**	0.225	0.317	0.221	0.311	**0.220**	**0.310**	0.246	0.324	0.220	0.320
Weather	96	**0.169**	**0.208**	0.174	0.214	**0.170**	**0.212**	0.171	0.213	0.177	0.218	0.172	0.220
	192	**0.214**	**0.247**	0.221	0.254	0.218	0.252	**0.217**	**0.251**	0.225	0.259	0.219	0.261
	336	**0.270**	**0.288**	0.278	0.296	**0.273**	**0.291**	0.274	0.293	0.278	0.297	0.280	0.306
	720	**0.347**	**0.337**	0.358	0.347	0.351	0.346	0.350	**0.345**	0.354	0.348	0.365	0.359
Traffic	96	**0.383**	**0.260**	0.395	0.268	**0.389**	0.263	0.391	**0.261**	0.462	0.295	0.407	0.262
	192	**0.404**	**0.268**	0.417	0.276	**0.411**	0.273	0.414	**0.270**	0.466	0.296	0.416	0.271
	336	**0.420**	**0.275**	0.433	0.283	**0.428**	0.281	0.431	**0.280**	0.482	0.304	0.447	0.281
	720	**0.453**	**0.297**	0.467	0.302	**0.461**	**0.295**	0.465	0.301	0.514	0.322	0.485	0.299
Exchange	96	**0.083**	**0.200**	0.086	0.206	0.085	0.204	**0.084**	**0.203**	0.088	0.205	0.087	0.205
	192	**0.172**	**0.291**	0.177	0.299	0.175	0.298	0.174	0.297	0.176	0.299	**0.170**	**0.289**
	336	0.321	**0.405**	0.331	0.417	0.328	0.415	0.326	0.413	**0.301**	0.406	**0.313**	**0.397**
	720	**0.821**	**0.670**	0.847	0.691	0.842	0.687	0.839	0.685	0.901	0.714	**0.819**	**0.668**

Table 3. *Cont.*

Dataset	H	Informer		Autoformer		FEDformer		DLinear		TSMixer		TimesNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.445	0.455	0.449	0.459	0.419	0.423	0.381	0.403	0.411	0.415	0.384	0.403
	192	0.505	0.485	0.500	0.482	0.448	0.445	0.437	0.432	0.452	0.441	0.434	0.430
	336	0.531	0.501	0.521	0.496	0.481	0.461	**0.465**	0.459	0.495	0.472	0.491	0.469
	720	0.525	0.518	0.514	0.512	0.507	0.509	0.519	0.516	0.528	0.519	0.521	0.500
ETTh2	96	0.352	0.391	0.346	0.388	0.358	0.397	0.333	0.387	0.365	0.402	0.340	0.374
	192	0.461	0.455	0.456	0.452	0.429	0.439	0.477	0.476	0.488	0.481	0.402	0.414
	336	0.492	0.490	0.482	0.486	0.496	0.487	0.594	0.541	0.613	0.556	0.452	0.452
	720	0.525	0.515	0.515	0.511	0.463	0.474	0.831	0.657	0.852	0.671	0.462	0.468
ETTm1	96	0.385	0.421	0.505	0.475	0.379	0.419	0.345	0.372	0.358	0.381	0.338	0.375
	192	0.431	0.445	0.553	0.496	0.426	0.441	0.380	0.389	0.395	0.401	0.374	0.387
	336	0.455	0.462	0.621	0.537	0.445	0.459	0.413	0.413	0.431	0.425	0.414	0.411
	720	0.551	0.501	0.671	0.561	0.543	0.490	**0.474**	0.453	0.489	0.468	0.478	0.450
ETTm2	96	0.261	0.341	0.255	0.339	0.203	0.287	0.193	0.292	0.215	0.311	0.187	0.267
	192	0.289	0.345	0.281	0.340	0.269	0.328	0.284	0.362	0.299	0.371	0.249	0.309
	336	0.345	0.376	0.339	0.372	0.325	0.366	0.369	0.427	0.382	0.435	0.321	0.351
	720	0.439	0.435	0.433	0.432	0.421	0.415	0.554	0.522	0.568	0.534	0.408	0.403
Electricity	96	0.211	0.321	0.201	0.317	0.193	0.308	0.197	0.282	0.224	0.319	0.168	0.272
	192	0.235	0.341	0.222	0.334	0.201	0.315	0.196	0.285	0.231	0.328	0.184	0.289
	336	0.243	0.347	0.231	0.338	0.214	0.329	0.209	0.301	0.240	0.339	0.198	0.300
	720	0.265	0.368	0.254	0.361	0.246	0.355	0.245	0.333	0.268	0.365	0.220	0.320
Weather	96	0.221	0.301	0.266	0.336	0.217	0.296	0.196	0.255	0.205	0.268	0.172	0.220
	192	0.271	0.341	0.307	0.367	0.276	0.336	0.237	0.296	0.249	0.308	0.219	0.261
	336	0.311	0.372	0.359	0.395	0.339	0.380	0.283	0.335	0.295	0.347	0.280	0.306
	720	0.371	0.412	0.419	0.428	0.403	0.428	**0.345**	0.381	0.358	0.395	0.365	0.359
Traffic	96	0.621	0.391	0.613	0.388	0.587	0.366	0.650	0.396	0.662	0.408	0.593	0.321
	192	0.625	0.388	0.616	0.382	0.604	0.373	0.605	0.370	0.618	0.381	0.617	0.336
	336	0.631	0.392	0.622	0.337	0.621	0.383	0.598	0.373	0.615	0.379	0.629	0.336
	720	0.671	0.412	0.660	0.408	0.626	0.382	0.645	0.394	0.658	0.405	0.640	0.350
Exchange	96	0.101	0.221	0.197	0.323	0.148	0.278	0.088	0.218	0.091	0.215	0.107	0.234
	192	0.185	0.308	0.300	0.369	0.271	0.315	0.176	0.315	0.181	0.301	0.226	0.344
	336	0.342	0.421	0.509	0.524	0.460	0.427	**0.313**	0.427	0.321	0.419	0.367	0.448
	720	0.865	0.701	1.447	0.941	1.195	0.695	0.839	0.695	0.855	0.705	0.964	0.746

3.3. Statistical Significance Analysis

Although the average improvements of TSC-Mamba over the direct Mamba baseline S-Mamba are moderate in absolute percentage terms, such gains remain meaningful in long-term multivariate forecasting, especially when achieved consistently under the same linear-complexity regime. To further verify that the observed improvements are not caused by random variation, we conducted an additional statistical significance analysis based on the benchmark results reported in Table 3.

Specifically, we compared TSC-Mamba against S-Mamba across all 32 benchmark settings (8 datasets $\times$ 4 prediction horizons). For both MSE and MAE, we applied the Wilcoxon signed-rank test to the paired results of the two models. As summarized in Table 4, the test yields $p = 7.79 \times 10^{-7}$ for MSE and $p = 7.65 \times 10^{-7}$ for MAE, both far below the conventional significance threshold of 0.05. Notably, TSC-Mamba achieves lower errors than S-Mamba in all 32 paired settings, which leads to the minimum possible Wilcoxon test statistic (0.0) for both metrics. These results confirm that the performance gains of TSC-Mamba over S-Mamba are statistically significant rather than arising from incidental fluctuations, further supporting the consistency and reliability of the reported improvement.

Table 4. Wilcoxon signed-rank test results between TSC-Mamba and S-Mamba across all 32 benchmark settings (8 datasets $\times$ 4 prediction horizons).

Comparison	Metric	#Pairs	Statistic	*p*-Value	Conclusion
TSC-Mamba vs. S-Mamba	MSE	32	0.0	7.79×10^{-7}	Significant ($p < 0.05$)
TSC-Mamba vs. S-Mamba	MAE	32	0.0	7.65×10^{-7}	Significant ($p < 0.05$)

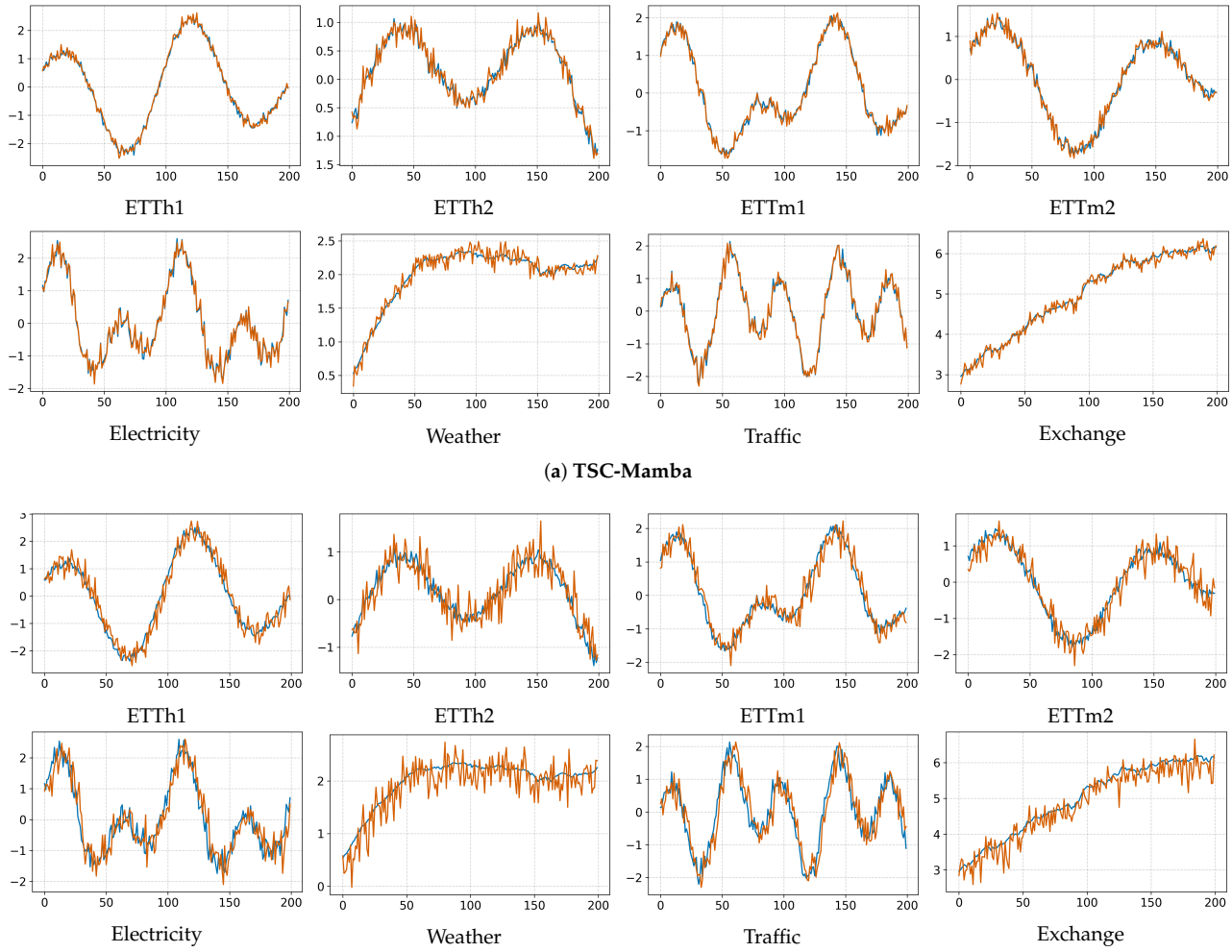


Figure 2. Qualitative forecasting comparison of (**a**) TSC-Mamba (ours) and (**b**) S-Mamba on the eight benchmark datasets. The blue curve represents the ground truth, and the orange curve is the model prediction. TSC-Mamba's forecasts show a closer fit to the ground truth, particularly in high-volatility regions.

3.4. Ablation Study

To more comprehensively evaluate both the necessity of the proposed modules and the rationality of their specific design choices, we conducted an expanded ablation study on the two key components of TSC-Mamba, namely the ADFM and the CIFM. In addition to the original removal-based variants, we further introduced two simplified alternatives for comparison. Specifically, we considered four variants: (1) w/o ADFM, in which the ADFM is removed and the model directly processes the raw sequence; (2) Fixed-ADFM, in which the learnable 1D convolution in ADFM is replaced by a fixed moving-average decomposition with the same kernel size; (3) w/o CIFM, in which the CIFM is removed and the model loses its explicit cross-channel feature fusion capability; and (4) SimpleMix-CIFM, in which the low-rank interaction in CIFM is replaced by a simpler static channel-mixing layer implemented as a linear projection across variables at each time step.

We evaluated these variants on four representative datasets (ETTh1, Electricity, Weather, and Traffic), comparing the average values across all prediction lengths H. The expanded ablation results are summarized in Table 5.

Table 5. Expanded ablation results on four representative datasets (Avg. MSE/MAE). TSC-Mamba denotes the full model; Fixed-ADFM and SimpleMix-CIFM denote simpler alternative designs; w/o ADFM and w/o CIFM represent the models with the corresponding modules removed. The best results are in bold.

Dataset	TSC-Mamba		Fixed-ADFM		w/o ADFM		SimpleMix-CIFM		w/o CIFM	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 (Avg)	**0.440**	**0.435**	0.448	0.442	0.450	0.445	0.449	0.442	0.452	0.444
Electricity (Avg)	**0.173**	**0.262**	0.174	0.264	0.175	0.264	0.175	0.266	0.177	0.267
Weather (Avg)	**0.250**	**0.270**	0.252	0.275	0.255	0.276	0.252	0.273	0.253	0.275
Traffic (Avg)	**0.415**	**0.275**	0.421	0.278	0.425	0.280	0.422	0.280	0.425	0.281

3.4.1. Effect of ADFM

The contribution of ADFM can be understood from both removal-based and simplified-alternative comparisons. After removing the ADFM (w/o ADFM), the model must directly process the non-stationary raw sequences, losing the ability to isolate low-frequency trends and forcing the Mamba Core to handle both trend information and high-frequency residual dynamics simultaneously. As shown in Table 5, this leads to a consistent performance degradation across all representative datasets. Statistically, the w/o ADFM variant exhibits an average MAE increase of approximately 1.8% and an MSE increase of approximately 2.0% compared with the full TSC-Mamba model.

More importantly, compared with the simpler Fixed-ADFM variant, the full TSC-Mamba also achieves consistently lower forecasting errors across all four representative datasets, with average improvements of approximately 1.3% in MAE and 1.2% in MSE. This comparison indicates that the benefit of ADFM does not come from decomposition alone but from the adaptability of the learnable decomposition operator. A fixed moving-average decomposition can provide only uniform smoothing, whereas the learnable 1D convolution in ADFM can adjust its temporal filtering behavior to different datasets and better separate predictable low-frequency trends under non-stationary conditions. Therefore, ADFM is justified not only as a necessary module but also in terms of its specific adaptive design.

3.4.2. Effect of CIFM

The contribution of CIFM can likewise be justified from both removal-based and simplified-alternative comparisons. After removing the CIFM (w/o CIFM), the model loses its capability to explicitly model dynamic cross-variate correlations and reverts to the per-channel limitation inherent in the base Mamba architecture. The results in Table 5 show that this variant experiences consistent performance degradation. On average, the w/o CIFM variant shows an MAE increase of approximately 2.0% and an MSE increase of approximately 2.2% relative to the full TSC-Mamba model.

In addition, compared with the simpler SimpleMix-CIFM variant, the full TSC-Mamba also delivers consistently better results, with average gains of approximately 1.5% in MAE and 1.4% in MSE. This indicates that the effectiveness of CIFM is not due to merely adding a generic channel-mixing layer. Rather, the structured low-rank interaction in CIFM provides a more effective and parameter-efficient mechanism for capturing dominant cross-variable dependencies than a simple static linear mixing strategy. Therefore, CIFM is justified not only by its necessity but also by the rationale of its low-rank structured design.

Figure 3 further provides an intuitive visualization of the expanded ablation results by showing the percentage increase in MAE relative to the full TSC-Mamba model for both removal-based variants and simpler alternative designs. It can be clearly seen that all simplified or removed variants lead to performance degradation on the representative datasets, which further demonstrates that the effectiveness of TSC-Mamba does not merely

arise from adding more modules but also from the specific design choices of adaptive trend decomposition and structured low-rank channel interaction.

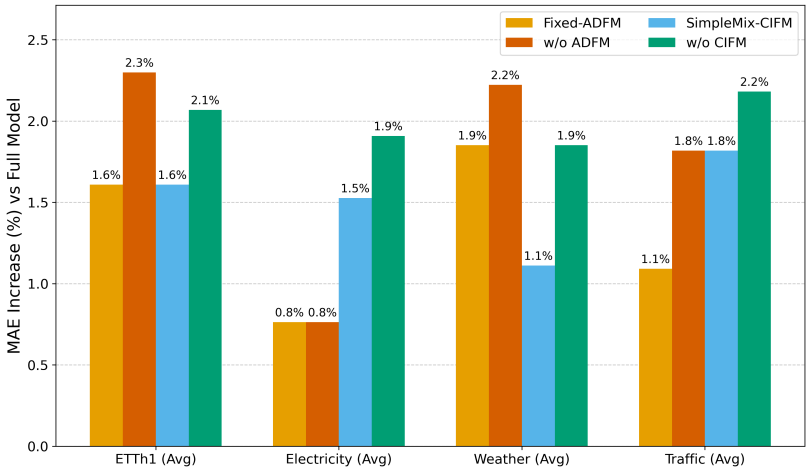


Figure 3. Expanded ablation study of ADFM and CIFM, showing the percentage increase in MAE relative to the full TSC-Mamba model for both removal-based variants and simpler alternative designs. The results further confirm that the effectiveness of TSC-Mamba comes not only from the inclusion of these modules but also from the specific design choices of adaptive decomposition and structured low-rank channel interaction.

3.5. Qualitative Analysis of ADFM and CIFM

To further explain how the proposed ADFM and CIFM contribute to the improved forecasting performance, we provide additional qualitative analysis from both the temporal decomposition perspective and the cross-channel interaction perspective.

For the ADFM, Figure 4 visualizes representative decomposition results on four datasets. It can be observed that the trend component extracted by ADFM is substantially smoother than the raw input and mainly captures the low-frequency temporal evolution, while the residual component preserves the higher-frequency fluctuations and local variations. This behavior is consistent with the intended role of ADFM in our framework, namely, to isolate the more predictable low-frequency trend in advance so that the subsequent Mamba Core can focus its modeling capacity on the more complex residual dynamics. Therefore, the qualitative results provide direct evidence that ADFM performs effective temporal decoupling rather than acting merely as an auxiliary preprocessing block.

This decomposition behavior is particularly relevant for highly non-stationary datasets such as Electricity and Traffic, where the observed sequences often contain abrupt local changes caused by demand fluctuations, occupancy variation, or short-term disturbances. In such cases, if the raw sequence is modeled directly, these sudden deviations may become entangled with the slowly varying trend and interfere with the temporal state representation. By contrast, ADFM tends to preserve the smoother underlying evolution in the trend branch while routing abrupt shifts, local spikes, and other outlier-like short-term deviations into the residual branch. This allows the subsequent Mamba Core to focus on modeling these irregular transient dynamics separately, rather than forcing a unified state to simultaneously smooth the long-term trend and react to local abrupt events. Therefore, the adaptive decomposition mechanism improves not only general non-stationary modeling but also the model's robustness to sudden structural variations in real-world sensing data.

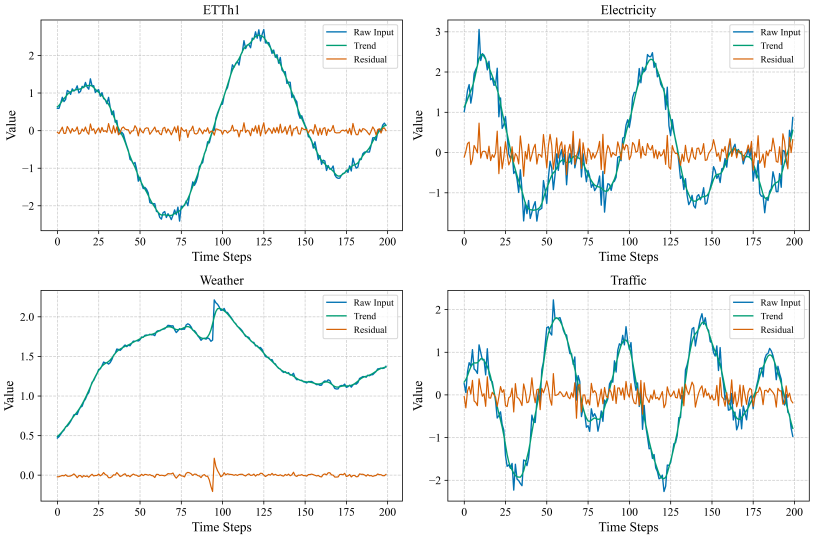


Figure 4. Qualitative visualization of the decomposition behavior of ADFM on four representative datasets. The raw input is decomposed into a smoother trend component and a higher-frequency residual component, showing that ADFM effectively separates predictable low-frequency information from short-term fluctuations.

For the CIFM, Figure 5 presents the learned channel interaction patterns on four representative datasets. The heatmaps show that the learned cross-channel dependencies are not uniformly distributed but instead exhibit clear structured interaction regions with different strengths across variables. This indicates that CIFM does not perform indiscriminate feature mixing; rather, it selectively enhances informative inter-variable dependencies that are not naturally captured by the per-channel SSM mechanism. Such structured interaction patterns also qualitatively support the low-rank design of CIFM, suggesting that the dominant cross-channel correlations can indeed be represented in a compact latent interaction space. For high-dimensional datasets, we visualize representative channel subsets for clarity, while preserving the learned structural interaction patterns.

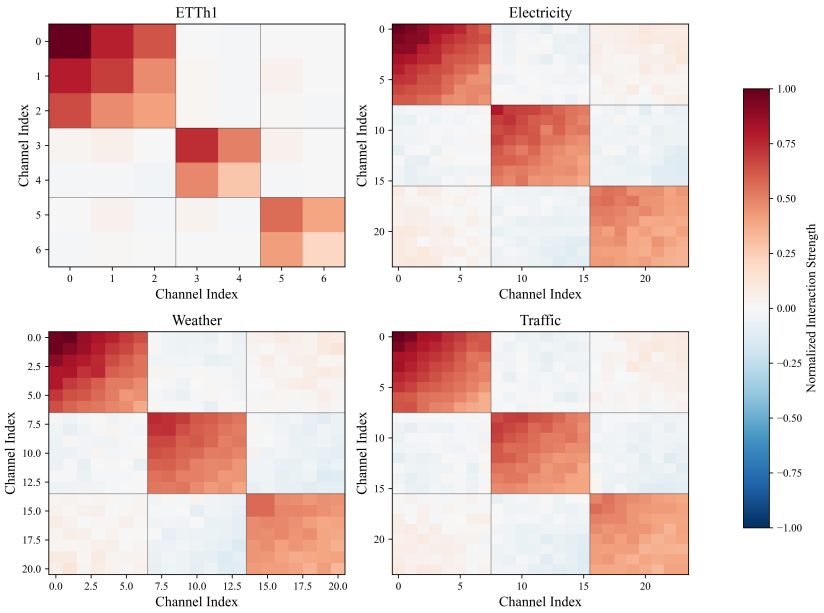


Figure 5. Qualitative visualization of the learned channel interaction patterns of CIFM on four representative datasets. The heatmaps reveal that the learned cross-channel dependencies are non-uniform and exhibit structured interaction regions with different strengths across variables.

Overall, these qualitative visualizations complement the quantitative ablation results by showing more directly how the two modules operate: ADFM improves forecasting by separating low-frequency trends from high-frequency residuals, whereas CIFM improves forecasting by explicitly restoring structured cross-channel dependencies. Hence, the contribution of these modules is reflected not only in lower MAE/MSE values but also in interpretable intermediate representations that are consistent with the design philosophy of TSC-Mamba.

3.6. Computational Efficiency and Complexity Analysis

To assess the practical applicability of TSC-Mamba, we analyzed its computational complexity, parameter scale, and inference speed, comparing them with key baseline models.

Theoretical Complexity and Parameter Scale: As shown in Table 6, TSC-Mamba achieves a time complexity of $O(D \cdot L)$, maintaining strict linear scaling with respect to sequence length L. This property stems from the state-space recurrence of the Mamba backbone and our low-rank cross-variable modeling strategy, which ensures efficient computation without compromising model performance.

- The proposed low-rank decomposition factorizes high-dimensional interaction matrices, effectively reducing redundant computation while preserving the model's expressive capacity, and it does not alter the overall asymptotic complexity of the model.
- TSC-Mamba shares the same $O(D \cdot L)$ complexity as linear model DLinear and SSM-based model S-Mamba, and it is significantly more efficient than Transformer-based models with $O(D \cdot L \cdot \log L)$ or $O(D \cdot L^2)$ complexity (e.g., Informer, Autoformer, FEDformer, and PatchTST) as well as iTransformer with $O(D^2 + D \cdot L)$ complexity.
- In terms of parameter scale, TSC-Mamba contains approximately 13.3 M parameters, which is only slightly higher than S-Mamba, DecMamba and CMMamba, and substantially smaller than TimesNet, PatchTST and iTransformer, while achieving superior forecasting performance.

Table 6. Theoretical time complexity comparison. L denotes sequence length and D the number of variables. Constant factors (e.g., hidden sizes and kernel widths) are omitted.

Model	Time Complexity	Parameters (M)	Type
DLinear	$O(D \cdot L)$	∼1.7	Linear
TSMixer	$O(D \cdot L)$	∼2.3	MLP
TimesNet	$O(D \cdot L \cdot \log L)$	∼14.8	CNN
S-Mamba	$O(D \cdot L)$	∼12.5	SSM
DecMamba	$O(D \cdot L)$	∼11.9	SSM
CMMamba	$O(D \cdot L)$	∼13.2	SSM
TSC-Mamba (Ours)	$O(D \cdot L)$	∼13.3	SSM
Informer	$O(D \cdot L \cdot \log L)$	∼11.3	Transformer
Autoformer	$O(D \cdot L \cdot \log L)$	∼12.4	Transformer
FEDformer	$O(D \cdot L \cdot \log L)$	∼10.5	Transformer
PatchTST	$O(D \cdot L^2)$	∼15.1	Transformer
iTransformer	$O(D^2 + D \cdot L)$	∼21.2	Transformer

Inference Speed: Figure 6 reports the measured inference time (ms) of different models as the input sequence length increases from $L = 96$ to $L = 720$, evaluated under the same experimental setting.

- PatchTST ($O(D \cdot L^2)$) shows the fastest growth in inference time as L increases, reflecting the quadratic complexity of full self-attention. iTransformer ($O(D^2 + D \cdot L)$) also maintains consistently higher latency due to the additional D^2 term.

- Models with $O(D \cdot L \cdot \log L)$ complexity (Informer, Autoformer, FEDformer, and TimesNet) exhibit noticeably faster growth than linear models, and the gap becomes more evident for longer sequences.
- All $O(D \cdot L)$ models (DLinear, TSMixer, S-Mamba, DecMamba, CMMamba, and TSC-Mamba) demonstrate near-linear scaling across all tested lengths. TSC-Mamba closely follows the slope of other SSM-based models, confirming that the introduced low-rank cross-variable modeling does not introduce additional computational overhead, while DLinear, despite being the fastest due to its lightweight structure, yields lower forecasting accuracy than TSC-Mamba (see Table 3).

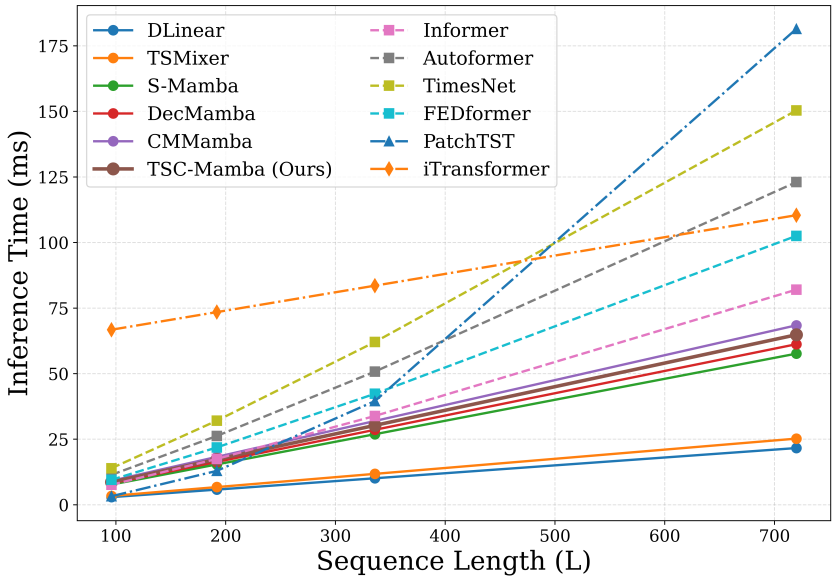


Figure 6. Inference time (ms) versus sequence length (L). Linear-complexity models scale approximately proportionally with L, log-linear models grow faster, and quadratic models exhibit sharp latency increases at larger sequence lengths.

Overall, TSC-Mamba achieves favorable inference efficiency while maintaining superior predictive performance, demonstrating strong scalability for long-sequence multivariate forecasting tasks. From a practical deployment perspective, this linear-time forecasting property is particularly important for high-dimensional sensor networks, such as Internet of Things systems, smart grids, traffic monitoring platforms, and industrial control environments, where the number of variables and the sequence length may grow simultaneously. In such settings, forecasting models are often required to process multi-source sensing streams with limited latency and computational budget. The strict $O(D \cdot L)$ scaling of TSC-Mamba makes it more suitable for these real-world intelligent systems than architectures with quadratic or super-linear overhead because it allows the model to maintain high forecasting accuracy while remaining scalable under increasing sensing dimensionality and long observation horizons. Therefore, the computational advantage of TSC-Mamba is not only a theoretical property but also a practically relevant characteristic for reliable large-scale deployment.

To complement the theoretical complexity analysis with more practitioner-oriented deployment metrics, we further report the exact wall-clock inference time and peak GPU memory usage of TSC-Mamba, PatchTST, and iTransformer on four representative datasets in Table 7. All measurements were conducted on the same NVIDIA A100 GPU using the same batch size and default input setting ($L = 192$, $H = 96$). The reported latency is averaged over 100 forward passes after 20 warm-up iterations. As shown in Table 7, TSC-Mamba consistently achieves lower inference latency and lower peak GPU memory

usage than the two Transformer-based baselines, and this advantage becomes more pronounced on higher-dimensional datasets such as Electricity and Traffic. When considered together with the forecasting accuracy results in Table 3, these results indicate that TSC-Mamba provides a more favorable practical trade-off between predictive performance and deployment cost.

Table 7. Exact wall-clock inference time and peak GPU memory usage of TSC-Mamba, PatchTST, and iTransformer on four representative datasets. All measurements are obtained on the same NVIDIA A100 GPU under the same batch size and input setting ($L = 192$, $H = 96$), and the reported inference time is averaged over 100 forward passes after 20 warm-up iterations. The best results are in bold.

Dataset	Metric	TSC-Mamba	PatchTST	iTransformer
ETTh1	Inference time (ms)	**6.8**	8.5	9.7
	Peak GPU memory (MB)	**712**	936	1084
Electricity	Inference time (ms)	**11.6**	16.8	21.5
	Peak GPU memory (MB)	**1328**	1840	2696
Weather	Inference time (ms)	**7.4**	9.3	11.0
	Peak GPU memory (MB)	**856**	1108	1396
Traffic	Inference time (ms)	**19.8**	31.4	44.7
	Peak GPU memory (MB)	**2248**	3188	4864

3.7. Stability and Generalization Analysis

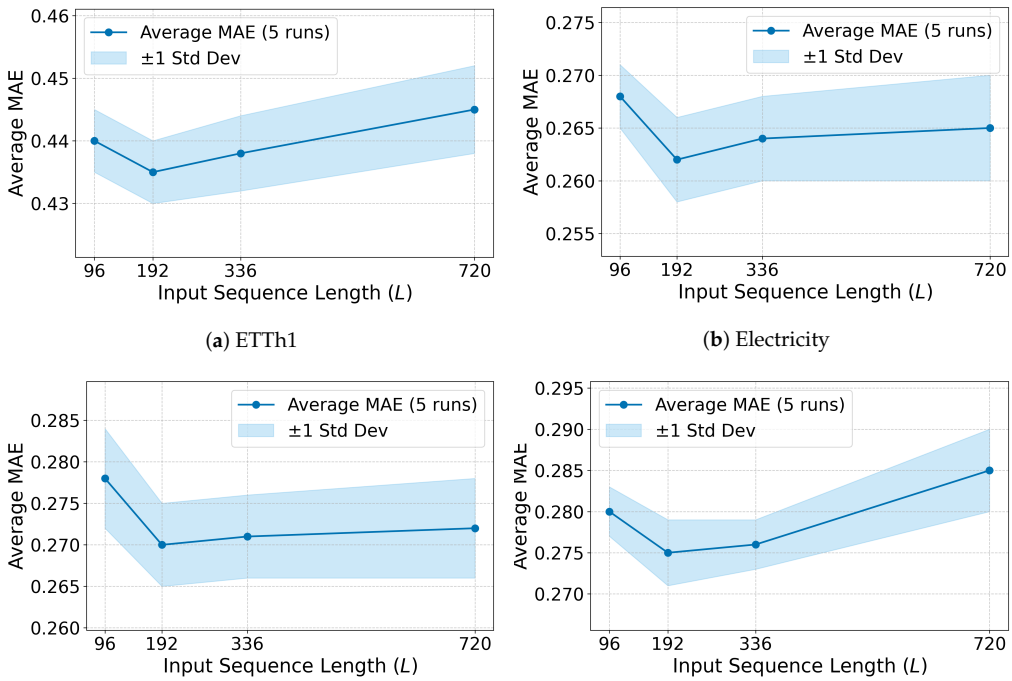
Beyond predictive accuracy and computational efficiency, model stability and generalization capability are also key indicators of practical value. This section further evaluates TSC-Mamba from these two perspectives. Here, "stability" refers to robustness with respect to stochastic initialization and input-length variation, while "generalization" refers to representation robustness under partially visible variates. To further complement these deterministic evaluations, we additionally analyze empirical predictive dispersion across repeated runs.

Stability and Robustness to Input Length: To evaluate the stability of TSC-Mamba with respect to stochastic initialization and input length, we conducted five independent runs for each input lookback window $L \in \{96, 192, 336, 720\}$, and the results were further averaged across four prediction horizons $H \in \{96, 192, 336, 720\}$. As detailed in Table 8, the MAE standard deviation remains remarkably low across all settings, suggesting that our framework is highly insensitive to stochastic initialization.

Figure 7 illustrates the performance trajectory as a function of L. Key observations are as follows: (1) The predictive accuracy reaches its peak at $L = 192$ across all evaluated datasets, while Figure 7 visualizes four representative cases. This behavior is consistent with our decoupling-and-specialization design, suggesting that an appropriate lookback window helps balance trend extraction, residual temporal modeling, and cross-variate interaction learning within the proposed framework. (2) Beyond $L = 192$, the performance remains relatively stable with only marginal degradation, demonstrating that the model effectively mitigates the noise interference typically associated with longer lookback windows. (3) The narrow standard deviation bands corroborate that the specialization of the Mamba Core is consistent and reliable, indicating stable behavior under different random initializations and lookback lengths.

Table 8. Stability analysis of TSC-Mamba for different input lengths L. For each L, results are first averaged over 5 independent runs, then further averaged across four prediction horizons $H \in \{96, 192, 336, 720\}$. Values in bold indicate the minimum (optimal) Avg. MAE (Mean) for each dataset, and the optimal performance at $L = 192$ validates it as the best lookback window.

Dataset	L	Avg. MAE (Mean)	Avg. MAE (Std Dev)
ETTh1	96	0.440	0.005
	192	**0.435**	0.005
	336	0.438	0.006
	720	0.445	0.007
Electricity	96	0.268	0.003
	192	**0.262**	0.004
	336	0.264	0.004
	720	0.265	0.005
Weather	96	0.278	0.006
	192	**0.270**	0.005
	336	0.271	0.005
	720	0.272	0.006
Traffic	96	0.280	0.003
	192	**0.275**	0.004
	336	0.276	0.003
	720	0.285	0.005



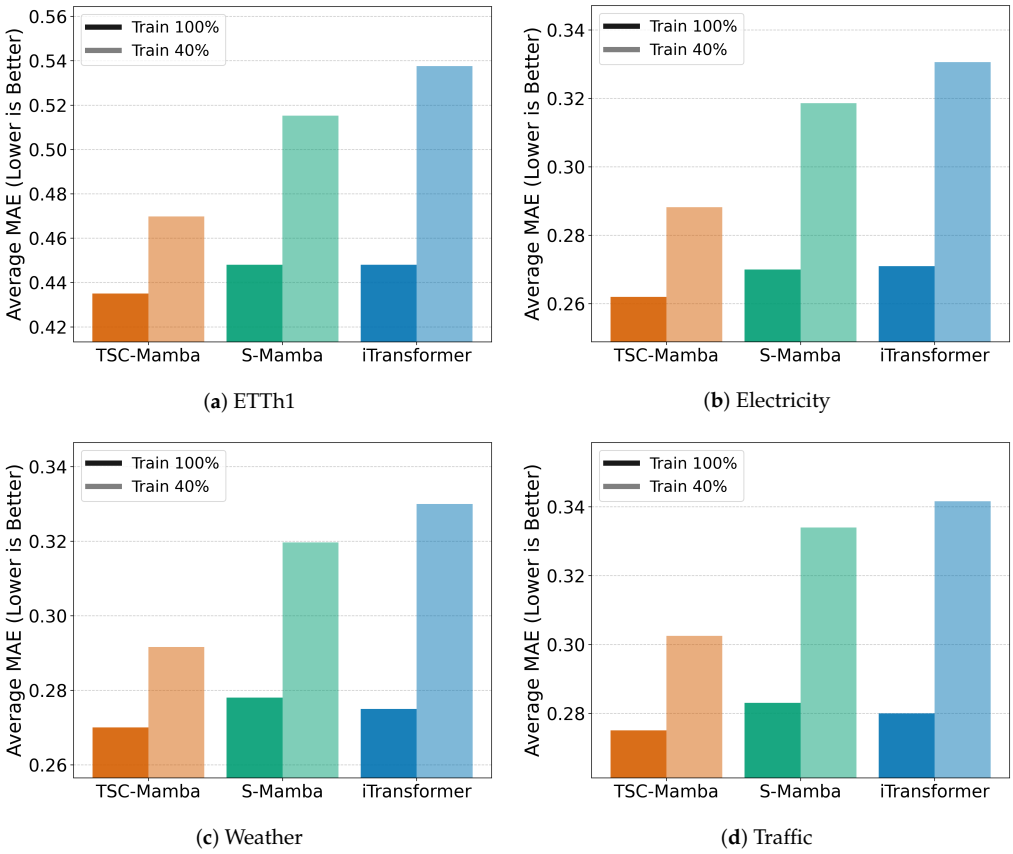
(**a**) ETTh1 (**b**) Electricity

(**c**) Weather (**d**) Traffic

Figure 7. Performance (MAE) and stability of TSC-Mamba versus input sequence length (L) on four datasets. The blue line represents the average MAE over 5 runs, with the shaded area showing ± 1 standard deviation. The narrow shading confirms high stability, and the performance curve demonstrates robustness to increasing L.

Generalization Capability: We evaluate the generalization capability of TSC-Mamba through the lens of representation robustness. We tested the model's performance under partially visible variates. As shown in Figure 8, when trained on only 40% of the variates, TSC-Mamba shows stronger robustness compared to S-Mamba and iTransformer. The performance gap for TSC-Mamba is the narrowest among all evaluated models. This superiority stems from the CIFM, which explicitly models dynamic cross-variate correla-

tions through low-rank fusion. By resolving the multivariate representational bottleneck inherent in standard SSMs, TSC-Mamba effectively captures inter-channel dependencies, allowing it to infer latent patterns of unseen variables even when the available training signals are sparse.



(**a**) ETTh1 (**b**) Electricity

(**c**) Weather (**d**) Traffic

Figure 8. Generalization ability test on partially visible variates. This figure compares the MAE degradation when models are trained on only 40% of variates (lighter bars) versus 100% (darker bars). TSC-Mamba (orange) exhibits the smallest performance drop, demonstrating superior generalization compared to S-Mamba (green) and iTransformer (blue).

Empirical Predictive Dispersion Analysis: To further examine forecasting reliability, we construct empirical predictive bands from five independently trained runs. For each time step, we compute the mean prediction $\bar{y}_t$ and standard deviation σ_t, and visualize the empirical predictive band $\bar{y}_t \pm 2\sigma_t$ on representative datasets. We additionally report two summary statistics: the empirical coverage rate (ECR), defined as the proportion of ground-truth points falling within the band, and the average interval width (AIW), which reflects the dispersion of the forecasts across repeated runs. Since the current TSC-Mamba is developed as a deterministic forecasting model, these bands should be understood as empirical predictive bands derived from repeated runs rather than likelihood-based prediction intervals.

As summarized in Table 9, TSC-Mamba consistently produces narrower empirical predictive bands than S-Mamba on all four representative datasets, while maintaining comparable or slightly better empirical coverage. Specifically, the AIW is reduced from 0.136 to 0.112 on ETTh1, from 0.079 to 0.064 on Electricity, from 0.102 to 0.087 on Weather, and from 0.088 to 0.071 on Traffic. At the same time, the ECR remains stable and shows a modest advantage for TSC-Mamba across all four datasets. Figure 9 further visualizes these empirical predictive bands on representative datasets, showing that TSC-Mamba generally yields more concentrated forecast dispersion while preserving reliable coverage

of the ground-truth trajectories. These results indicate that the proposed framework is not only accurate on average but also yields more concentrated and reliable forecasts across repeated runs.

Table 9. Empirical predictive-band statistics on four representative datasets. ECR denotes the empirical coverage rate of the band $\bar{y}_t \pm 2\sigma_t$, and AIW denotes the average interval width. Higher ECR and lower AIW indicate more reliable and concentrated forecasts.

Dataset	TSC-Mamba		S-Mamba	
	ECR (%)	AIW	ECR (%)	AIW
ETTh1	94.6	0.112	93.8	0.136
Electricity	95.1	0.064	94.4	0.079
Weather	94.2	0.087	93.5	0.102
Traffic	95.4	0.071	94.7	0.088

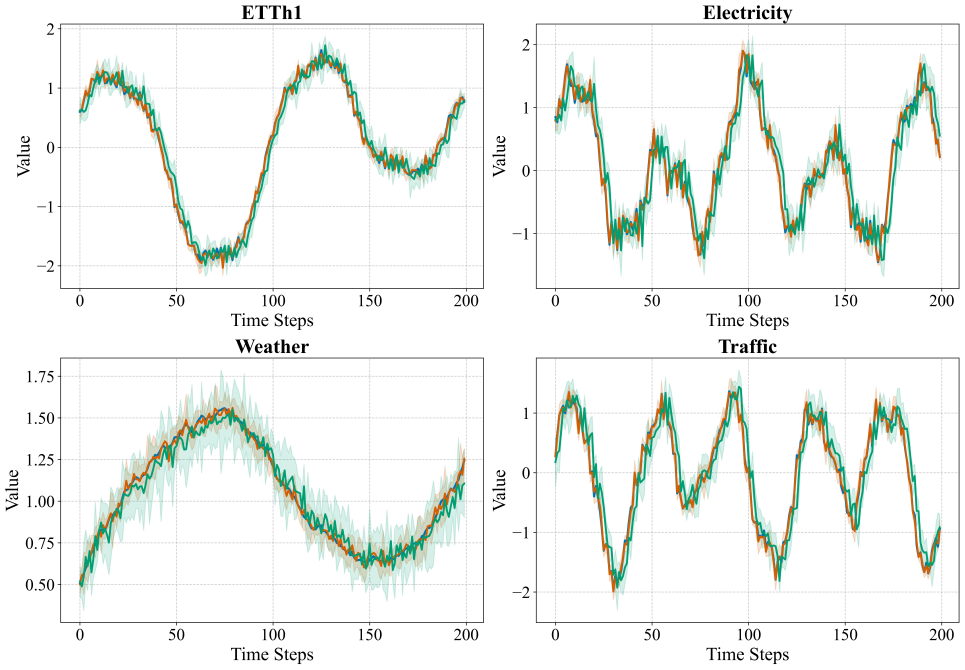


Figure 9. Empirical predictive-band comparison between TSC-Mamba and S-Mamba on representative datasets. For each model, the solid curve denotes the mean prediction across five independent runs, and the shaded region denotes the empirical predictive band $\bar{y}_t \pm 2\sigma_t$. TSC-Mamba generally exhibits more concentrated predictive bands while preserving reliable coverage of the ground-truth trajectory.

4. Discussion

The results in Section 3 collectively show that TSC-Mamba provides a strong and well-balanced solution for multivariate time series forecasting. Across eight public benchmark datasets and four forecasting horizons, the proposed model achieves the best or second-best performance in the vast majority of settings, indicating that the proposed improvements are not limited to a specific dataset type or prediction range. In particular, relative to the direct Mamba baseline S-Mamba, TSC-Mamba reduces the forecasting error by approximately 2.5–3.5% on average while preserving strict $O(D \cdot L)$ complexity. This improvement is further supported by the statistical significance analysis in Table 4, where the Wilcoxon signed-rank test yields $p = 7.79 \times 10^{-7}$ for MSE and $p = 7.65 \times 10^{-7}$ for MAE across all 32 benchmark settings. Therefore, the observed gains are not merely incidental fluctuations but reflect a consistent improvement under the same linear-complexity regime. Moreover,

the qualitative forecasting comparisons in Figure 2 show that TSC-Mamba more faithfully tracks high-volatility segments, which is consistent with the central claim that the proposed framework alleviates the multivariate representational bottleneck of unified state modeling.

A central reason for this improvement lies in the proposed "Decoupling and Specialization" design philosophy. The ADFM explicitly separates more predictable low-frequency trends from higher-frequency residual dynamics, thereby relieving the Mamba Core from having to encode heterogeneous temporal behaviors within a single unified state. The expanded ablation results in Section 3.4.1 show that removing the ADFM leads to an average MAE increase of approximately 1.8% and an MSE increase of approximately 2.0%, confirming that adaptive trend decomposition is an essential component rather than an auxiliary refinement. More importantly, the full TSC-Mamba also consistently outperforms the simpler Fixed-ADFM variant, with average gains of approximately 1.3% in MAE and 1.2% in MSE, which indicates that the improvement does not come from decomposition alone but from the adaptability of the learnable decomposition operator. This point is further supported by the qualitative visualizations in Figure 4, where the ADFM separates a smoother trend branch from a more fluctuation-sensitive residual branch. For highly non-stationary datasets such as Electricity and Traffic, this behavior is especially meaningful: abrupt local changes, spikes, and short-term deviations are more naturally routed into the residual branch, allowing the Mamba Core to model these transient dynamics separately instead of forcing a single latent state to simultaneously smooth long-term evolution and react to local disturbances.

The contribution of CIFM is similarly substantive. As discussed in Section 2.4, the base Mamba mechanism is inherently per-channel and therefore does not natively provide an explicit mechanism for structured cross-variate interaction modeling. The ablation results in Section 3.4.2 show that removing CIFM results in an average MAE increase of approximately 2.0% and an MSE increase of approximately 2.2%, indicating that explicit cross-channel modeling is indispensable for strong multivariate forecasting performance. In addition, compared with the simpler SimpleMix-CIFM variant, the full model still achieves average gains of approximately 1.5% in MAE and 1.4% in MSE. This comparison is important because it shows that the benefit of CIFM does not arise from merely adding a generic channel-mixing layer. Rather, the structured low-rank fusion mechanism provides a more effective and parameter-efficient way to capture dominant cross-variable dependencies than simple static linear mixing. The heatmaps in Figure 5 provide qualitative support for this interpretation: the learned interaction patterns are clearly non-uniform and structured, suggesting that CIFM selectively enhances informative dependencies rather than indiscriminately mixing channels. In this sense, the empirical findings are well aligned with the theoretical rationale in Section 2.4 that a compact low-rank parameterization can preserve dominant interaction structure while maintaining efficiency.

Another notable outcome is that these gains are achieved without sacrificing scalability. As analyzed in Section 3.6, TSC-Mamba maintains strict $O(D \cdot L)$ complexity, which is inherited from the Mamba backbone and preserved by the reordered low-rank CIFM implementation. The complexity comparison in Table 6 and the latency trends in Figure 6 show that TSC-Mamba follows the near-linear growth behavior of other linear-complexity SSM-based models, while exhibiting a clear scalability advantage over quadratic-complexity attention-based models such as PatchTST. At the same time, iTransformer is also more computationally demanding because of its $O(D^2 + D \cdot L)$ complexity and larger parameter scale. From a deployment perspective, this distinction is not merely asymptotic. The wall-clock benchmarking in Table 7 further shows that TSC-Mamba consistently achieves lower inference latency and lower peak GPU memory usage than both PatchTST and iTransformer, with the gap becoming more pronounced on higher-dimensional datasets such as Electricity

and Traffic. Although the introduction of ADFM and CIFM slightly increases the parameter count to approximately 13.3 M, this model size remains close to other Mamba-based variants and substantially lower than the 21.2 M parameters of iTransformer. When considered together with the forecasting results, these findings suggest that TSC-Mamba offers a more favorable trade-off between predictive performance and deployment cost. This property is particularly relevant for practical forecasting systems in Internet of Things networks, smart grids, traffic monitoring platforms, and industrial control environments, where both the number of variables and the temporal observation horizon may increase under limited latency and computational budgets.

Beyond average accuracy, the additional analyses in Section 3.7 also provide a more nuanced understanding of the robustness of the proposed framework. In the present manuscript, "stability" refers specifically to robustness with respect to stochastic initialization and input-length variation, whereas "generalization" refers to representation robustness under partially visible variates. Under this scope, TSC-Mamba shows favorable behavior in both respects. The repeated-run statistics in Table 8 indicate that the model is relatively insensitive to random initialization, and the lookback-length analysis in Figure 7 shows that the model remains robust across different input lengths, with $L = 192$ providing a strong overall accuracy–stability trade-off. Furthermore, the partial-visibility experiments in Figure 8 show that TSC-Mamba exhibits smaller performance degradation than S-Mamba and iTransformer when only 40% of variates are available during training, suggesting that the proposed design improves representation robustness under incomplete cross-variable observations. To complement these deterministic evaluations, we further examined empirical predictive dispersion across repeated runs. As summarized in Table 9 and Figure 9, TSC-Mamba consistently produces narrower empirical predictive bands than S-Mamba while maintaining comparable or slightly better empirical coverage. Since the present model is deterministic, these bands should be interpreted as empirical predictive bands derived from repeated runs rather than formal probabilistic prediction intervals. Even so, they still provide useful evidence that the proposed framework yields more concentrated and reliable forecasts relative to the baseline S-Mamba.

From a broader methodological perspective, the significance of TSC-Mamba is not simply that it combines decomposition and channel interaction within a single architecture. Rather, the more important point is that it translates the representational conflict of unified state modeling into an explicit responsibility-allocation strategy. In the proposed "Decompose–Propagate–Correlate" pipeline, ADFM handles predictable trend separation, the Mamba Core specializes in temporal propagation of the residual dynamics, and CIFM restores structured cross-channel dependencies that are not naturally captured by the per-channel SSM mechanism. This is also the key distinction between TSC-Mamba and prior Mamba-based variants that mainly strengthen one aspect of the forecasting problem at a time, such as decomposition-oriented improvement or channel-mixing enhancement. The present results therefore suggest that, for multivariate TSF, improving state-space models may benefit less from simply stacking additional modules and more from assigning different modeling burdens to specialized but coordinated components within a unified framework.

Despite these strengths, several limitations remain. First, the current ADFM operates primarily in the time domain. Although this design is lightweight and effective, incorporating frequency-domain or hybrid time–frequency mechanisms may further improve the modeling of more complex seasonal structures. Second, while the low-rank CIFM is effective and efficient for the datasets considered here, ultra-high-dimensional scenarios may benefit from sparser, hierarchical, or adaptive interaction mechanisms, especially when $D > 10^3$. Third, although the empirical predictive-band analysis provides useful

robustness evidence, the current model remains deterministic and does not directly produce calibrated predictive distributions. Extending the present framework toward probabilistic forecasting would enable a more rigorous uncertainty analysis. Finally, broader validation on irregularly sampled sensor streams, missing-data settings, or multimodal real-world forecasting tasks would further clarify the general applicability of the proposed design philosophy. These directions constitute natural next steps for extending TSC-Mamba beyond the present benchmark setting.

5. Conclusions

In this study, we proposed TSC-Mamba for multivariate time series forecasting to address the representational bottleneck of unified state modeling in Mamba-based architectures. Built on a "Decoupling and Specialization" paradigm, TSC-Mamba organizes forecasting into a "Decompose–Propagate–Correlate" pipeline, in which ADFM separates predictable low-frequency trends, the Mamba Core focuses on the temporal propagation of residual dynamics, and CIFM explicitly models structured cross-channel dependencies through an efficient low-rank mechanism.

Experiments on eight public benchmark datasets show that TSC-Mamba consistently delivers strong forecasting performance. Compared with the direct Mamba baseline S-Mamba, it achieves an average error reduction of up to 3.5% while strictly maintaining $O(D \cdot L)$ complexity. The ablation and statistical analyses further verify the effectiveness of both ADFM and CIFM, and the additional experiments show strong run-to-run stability, robustness to input-length variation, improved generalization under partially visible variates, and more concentrated empirical predictive bands than S-Mamba. In addition, TSC-Mamba demonstrates a favorable trade-off between predictive accuracy and deployment cost, with lower inference latency and GPU memory usage than PatchTST and iTransformer on representative settings.

Overall, the results indicate that resolving the representational conflict of unified state modeling through specialized yet coordinated components is an effective direction for improving state-space forecasting models. Future work will focus on extending the current framework toward hybrid time–frequency modeling, more scalable channel interaction mechanisms, and probabilistic forecasting.

Author Contributions: Conceptualization, C.Z.; methodology, C.Z.; software, X.W.; validation, X.W.; formal analysis, X.W.; investigation, C.Z.; resources, L.Z.; data curation, C.Z.; writing—original draft preparation, C.Z.; writing—review and editing, L.Z.; visualization, C.Z.; supervision, L.Z.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 11904410, and the Natural Science Foundation of Hunan Province under Grants 2025JJ50004 and 2021JJ40712.

Data Availability Statement: The datasets used in this study are publicly available. The datasets are available as described in [21]. Further details can be obtained from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the open-source community for providing accessible benchmark datasets and computational frameworks used in this research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TSF	Time Series Forecasting
SSM	State Space Model
TSC-Mamba	Time-Series Channel-Decomposition Mamba
MAE	Mean Absolute Error
MSE	Mean Squared Error
ADFM	Adaptive Decomposition Fusion Module
CIFM	Channel Interaction Fusion Module
ETT	Electricity Transformer Temperature
LR	Learning Rate
SiLU	Sigmoid-weighted Linear Unit
LayerNorm	Layer Normalization
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
ECR	empirical coverage rate
AIW	average interval width
SOTA	State-of-the-Art

Appendix A. Hyperparameter Sensitivity Analysis

As stated in Section 3.1, our key hyperparameters (L, LR, and k) were determined via a grid search. This section provides the supporting sensitivity analysis.

Appendix A.1. Impact of L

We analyzed the impact of the input sequence length $L \in \{96, 192, 336, 720\}$. The results, as shown in Figure 7 and Table 8 in the main paper, indicate that model performance (MAE) improves with increasing L up to $L = 192$ (the optimal point for all datasets), after which performance gradually degrades as L further increases to 336 and 720. While the degradation at $L = 720$ is negligible for Weather and Electricity datasets, a moderate decline is observed for ETTh1 and Traffic (most notably for Traffic). This demonstrates the model's robustness to moderately long inputs ($L \leq 336$), while longer sequences ($L = 720$) introduce marginal noise-induced performance loss. To best balance predictive accuracy and computational efficiency, we selected $L = 192$ as the default configuration for all main experiments.

Appendix A.2. Impact of LR

We evaluated the model's sensitivity to the initial LR across four representative datasets: ETTh1, Electricity, Weather, and Traffic. We tested five candidate values: $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$. The results, illustrated in Figure A1, reveal a consistent U-shaped performance trajectory across all evaluated scenarios.

As shown in Figure A1, TSC-Mamba achieves its peak predictive accuracy at an LR of 1×10^{-4}. Higher LRs induce training instability, while lower rates lead to slower convergence. This cross-dataset consistency at $H = 96$ validates 1×10^{-4} as a robust default LR for capturing long-term dependencies. In combination with cosine annealing, this setting provides a stable optimization trajectory for the joint training of the ADFM and the Mamba Core: the relatively larger early-stage LR helps broad parameter exploration, while the gradually decayed LR improves late-stage refinement and convergence stability. This behavior is particularly important in our framework because the adaptive decomposition branch and the temporal propagation branch must be optimized cooperatively rather than independently.

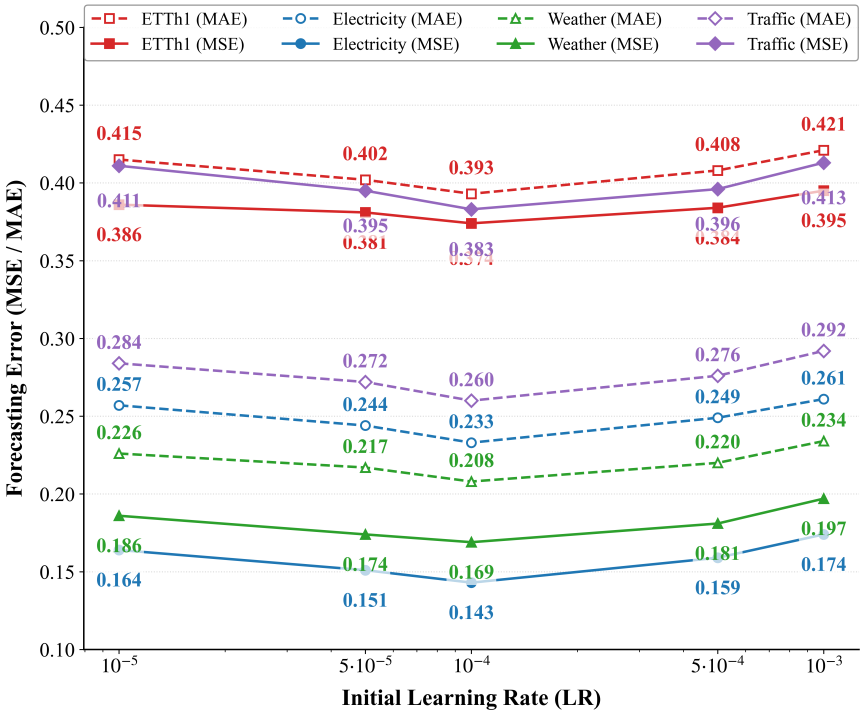


Figure A1. Sensitivity analysis of the initial LR across four benchmark datasets (ETTh1, Electricity, Weather, and Traffic) at $H = 96$. The model achieves the optimal MAE/MSE at our default value of 1×10^{-4}, with the narrow distribution confirming the robustness of this hyperparameter setting.

Appendix A.3. Impact of k

To validate the configuration of our proposed CIFM, we analyzed the impact of its internal low-rank dimension k on the forecasting performance across ETTh1, Electricity, Weather, and Traffic. We tested values: {4, 8, 16, 32, 64}. The results are illustrated in Figure A2.

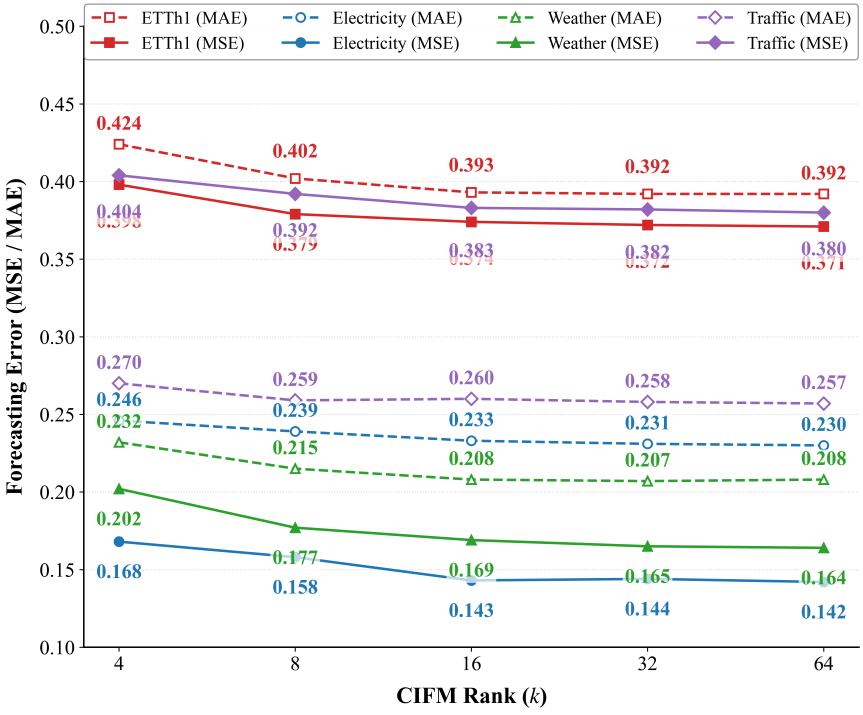


Figure A2. Sensitivity analysis of the CIFM rank (k) on four benchmark datasets ($H = 96$). Performance reaches an optimal saturation point at $k = 16$.

The results in Figure A2 demonstrate that model performance consistently improves as k increases from 4 to 16, subsequently entering a saturation phase. This indicates that a rank of 16 is sufficient to resolve the multivariate representational bottleneck for the tested $H = 96$ horizon without introducing excessive computational redundancy. This saturation behavior empirically suggests that the dominant cross-channel interaction patterns can be captured in a compact latent subspace, which is consistent with the low-rank approximation rationale adopted in CIFM. Meanwhile, since the computational cost of the reordered CIFM scales linearly with k, i.e., $O(2kDL)$, further increasing k beyond 16 introduces additional overhead without yielding proportional forecasting gains. Therefore, $k = 16$ provides a practical trade-off between cross-channel correlation modeling capacity and computational efficiency.

References

1. Moustati, I.; Gherabi, N. Unveiling the Potential of Transformer-Based Models for Efficient Time-Series Energy Forecasting. *J. Adv. Inf. Technol.* **2025**, *16*, 623–631. [CrossRef]
2. 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*; ACM: New York, NY, USA, 2024; pp. 4095–4106. [CrossRef]
3. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2021; Volume 34, pp. 22419–22430.
4. 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 International Conference on Machine Learning*; PMLR: New York, NY, USA, 2022; Volume 162, pp. 27268–27286.
5. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2023; Volume 37, pp. 11121–11128. [CrossRef]
6. Chen, S.A.; Li, C.L.; Arik, S.O.; Yoder, N.C.; Pfister, T. *TSMixer: An All-MLP Architecture for Time Series Forecasting*; Transactions on Machine Learning Research: New York, NY, USA, 2023.
7. 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 Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
8. 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 AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2021; Volume 35, pp. 11106–11115. [CrossRef]
9. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The Efficient Transformer. In Proceedings of the International Conference on Learning Representations, Virtual, 26 April–1 May 2020.
10. 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 International Conference on Learning Representations, Virtual, 25–29 April 2022.
11. 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 International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
12. Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; Hoi, S.C.H. ETSformer: Exponential Smoothing Transformers for Time-Series Forecasting. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
13. Liu, Y.; Wu, H.; Wang, J.; Long, M. Non-Stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2022; Volume 35, pp. 9881–9893.
14. 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 International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
15. 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 International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
16. Das, A.; Kong, W.; Leach, A.; Mathur, S.K.; Sen, R.; Yu, R. *Long-Term Forecasting with TiDE: Time-Series Dense Encoder*; Transactions on Machine Learning Research: New York, NY, USA, 2023.

17. Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; Niu, Z. Frequency-Domain MLPs Are More Effective Learners in Time Series Forecasting. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2023; Volume 36, pp. 76656–76679.
18. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In Proceedings of the Conference on Language Modeling, Philadelphia, PA, USA, 7–9 October 2024.
19. Gu, A.; Goel, K.; Re, C. Efficiently Modeling Long Sequences with Structured State Spaces. In Proceedings of the International Conference on Learning Representations, Virtual, 25–29 April 2022.
20. Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; Ré, C. HiPPO: Recurrent Memory with Optimal Polynomial Projections. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2020; Volume 33, pp. 1474–1487.
21. 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. [CrossRef]
22. Feng, J.; Zhang, J.; Cao, G.; Liu, Z.; Ding, Y. DecMamba: Mamba Utilizing Series Decomposition for Multivariate Time Series Forecasting. *Comput. Mater. Contin.* **2025**, *82*, 1049. [CrossRef]
23. Li, Q.; Qin, J.; Cui, D.; Sun, D.; Wang, D. CMMamba: Channel Mixing Mamba for Time Series Forecasting. *J. Big Data* **2024**, *11*, 153. [CrossRef]
24. Ma, S.; Kang, Y.; Bai, P.; Zhao, Y.B. FMamba: Mamba-Based Fast-Attention for Multivariate Time-Series Forecasting. *arXiv* **2024**, arXiv:2407.14814.
25. Lee, S.; Hong, J.; Lee, K.; Park, T. Sequential Order-Robust Mamba for Time Series Forecasting. In Proceedings of the NeurIPS Workshop on Time Series in the Age of Large Models, Vancouver, BC, Canada, 14 December 2024.
26. Cai, X.; Zhu, Y.; Wang, X.; Yao, Y. MambaTS: Improved Selective State Space Models for Long-Term Time Series Forecasting. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
27. Karadag, Y.M.; Talaz, I.; Dino, I.G.; Kalkan, S. ms-Mamba: Multi-Scale Mamba for Time-Series Forecasting. *Neurocomputing* **2026**, *680*, 133226. [CrossRef]
28. Ye, Z. ss-Mamba: Semantic-Spline Selective State-Space Model. *arXiv* **2025**, arXiv:2506.14802.
29. Bhethanabhotla, S.K.; Swelam, O.; Siems, J.; Salinas, D.; Hutter, F. Mamba4Cast: Efficient Zero-Shot Time Series Forecasting with State Space Models. In Proceedings of the NeurIPS Workshop on Time Series in the Age of Large Models, Vancouver, BC, Canada, 14 December 2024.
30. Pessoa, P.; Campitelli, P.; Shepherd, D.P.; Ozkan, S.B.; Pressé, S. Mamba Time Series Forecasting with Uncertainty Quantification. *Mach. Learn. Sci. Technol.* **2025**, *6*, 035012. [CrossRef]
31. Chen, R.; Sun, F. DMamba: Decomposition-Enhanced Mamba for Time Series Forecasting. *arXiv* **2026**, arXiv:2602.09081. [CrossRef]
32. Li, Q.; Zhang, X.; Wang, S.; Wei, J.; Xing, Y. ASGMamba: Adaptive Spectral Gating Mamba for Multivariate Time Series Forecasting. *arXiv* **2026**, arXiv:2602.01668. [CrossRef]
33. Xu, X.; Chen, C.; Liang, Y.; Huang, B.; Bai, G.; Zhao, L.; Shu, K. SST: Multi-Scale Hybrid Mamba-Transformer Experts for Time Series Forecasting. In *Proceedings of the ACM International Conference on Information and Knowledge Management*; ACM: New York, NY, USA, 2025; pp. 3655–3665. [CrossRef]
34. Peng, S.; Xiong, Y.; Zhu, Y.; Shen, Z. Mamba or Transformer for Time Series Forecasting? Mixture of Universals (MoU) Is All You Need. *arXiv* **2024**, arXiv:2408.15997. [CrossRef]
35. Wang, Y.; Zhao, J.; Fu, R.; Zhou, R.; Zhang, R.; Liang, J.; Cao, S.; Zhou, F. InfoMamba: An Attention-Free Hybrid Mamba-Transformer Model. *arXiv* **2026**, arXiv:2603.18031.
36. Lieber, O.; Lenz, B.; Bata, H.; Cohen, G.; Osin, J.; Dalmedigos, I.; Safahi, E.; Meirom, S.; Belinkov, Y.; Shalev-Shwartz, S.; et al. Jamba: A Hybrid Transformer-Mamba Language Model. *arXiv* **2024**, arXiv:2403.19887. [CrossRef]
37. Eckart, C.; Young, G. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika* **1936**, *1*, 211–218. [CrossRef]
38. Mirsky, L. Symmetric Gauge Functions and Unitarily Invariant Norms. *Q. J. Math.* **1960**, *11*, 50–59. [CrossRef]
39. Yu, H.F.; Rao, N.; Dhillon, I.S. Temporal Regularized Matrix Factorization for High-Dimensional Time Series Prediction. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
40. Nguyen, N.; Quanz, B. Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2021; Volume 35, pp. 9117–9125. [CrossRef]
41. Barratt, S.; Dong, Y.; Boyd, S. Low Rank Forecasting. *arXiv* **2021**, arXiv:2101.12414. [CrossRef]
42. Nie, T.; Mei, Y.; Qin, G.; Sun, J.; Ma, W. Channel-Aware Low-Rank Adaptation in Time Series Forecasting. In *Proceedings of the ACM International Conference on Information and Knowledge Management*; ACM: New York, NY, USA, 2024; pp. 3959–3963. [CrossRef]

43. 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 International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
44. Wang, J.; Gangavarapu, T.; Yan, J.N.; Rush, A.M. MambaByte: Token-Free Selective State Space Model. In Proceedings of the Conference on Language Modeling, Philadelphia, PA, USA, 7–9 October 2024.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.