DuoMamba: A Decomposition-Free State Space Model for Long-Term Time Series Forecasting

Anna-Alina Bondarets [1] **Taras Rumezhak** [1] **Volodymyr Karpiv** [1]

Abstract

Recent state-space-model forecasters for long-term time series share a common design: trend, seasonal, or frequency decomposition modules stacked on top of a Mamba-1 backbone, as in KARMA, SST, and DMamba. Whether these modules are architecturally essential, or whether they are compensating for Mamba-1's limited multi-scale expressivity, has not been tested directly. We introduce **DuoMamba**, a forecaster with Mamba-2 backbone with chunk-aligned patch tokenization and *no* decomposition module, and run a controlled ablation that varies the backbone (Mamba-1 vs. Mamba-2) and the presence of decomposition across three published architectures. Across nine standard LTSF benchmarks the ablation produces a *sign flip*: decomposition reduces MSE under Mamba-1 but *increases* it under Mamba-2, while DuoMamba alone matches or exceeds the state of the art on most datasets. DuoMamba also sets new SOTA on the high-channel Traffic and ECL benchmarks, where prior decomposition-heavy methods explicitly fail, and reduces parameters and training-step latency by roughly half. Together these results indicate that decomposition modules in current SSM forecasters compensate for limitations of an earlier backbone generation rather than encoding a permanent inductive bias.

1. Introduction

Long-term time-series forecasting (LTSF) (Wu et al., 2021; Zhou et al., 2021; Salinas et al., 2020) is dominated by Transformer models (Nie et al., 2023; Liu et al., 2024; Zhou et al., 2022) that scale quadratically with sequence length and by linear-time state-space models of the Mamba family (Gu & Dao, 2024; Gu et al., 2022). Once S-Mamba (Wang et al., 2024b), MambaTS (Cai et al., 2024), TimeMachine (Ahamed & Cheng, 2024), and Bi-Mamba+ (Liang et al., 2024) established Mamba's viability for LTSF, a wave of *hybrid decomposition* forecasters such as KARMA (Ye et al., 2025), SST (Chen et al., 2024), DMamba (Chen & Sun, 2026), and AdaMamba (Jiang et al., 2026) layered trend, seasonal, and frequency-domain decomposition modules on top of a Mamba-1 (Gu & Dao, 2024) backbone.

The implicit assumption is that these decomposition modules are mechanistically necessary, yet no prior study has tested this by varying only the SSM backbone, even though Mamba-2 (Dao & Gu, 2024) / Structured State-Space Duality (SSD) provides a more expressive recurrence at lower hardware cost. We run this experiment.

We introduce **DuoMamba**, a minimal Mamba-2 forecaster with chunk-aligned patch tokenization and no decomposition module, and use it with backbone-controlled variants of three published architectures to expose a *sign flip*: explicit decomposition reduces error under Mamba-1 but does not help and often hurts under Mamba-2.

Contributions. To sum up, our contributions are:

- **DuoMamba**, a decomposition-free Mamba-2 (Dao & Gu, 2024) / SSD forecaster with chunk-aligned patch tokenization and a single SSM stack, using no DWT, no attention-based trend extractor, and no frequency-domain loss, yet matching or exceeding the state of the art on most LTSF benchmarks.
- **The decomposition sign flip.** A $2\times2\times3$ ablation (backbone $\times$ decomposition $\times$ architecture) showing that decomposition flips from helpful under Mamba-1 (Gu & Dao, 2024) to net-harmful under Mamba-2 (Dao & Gu, 2024), the first backbone-controlled study of decomposition in SSM forecasters.
- **A simpler, more efficient baseline.** DuoMamba is substantially simpler than the decomposition-heavy SOTA and faster per training step on high-dimensional datasets such as Traffic and ECL, where prior methods explicitly admit difficulty.

[1]SoftServe R&D Department, Lviv, Ukraine. Correspondence to: Anna-Alina Bondarets <anbondaret@softserveinc.com>.

Proceedings of the ICML Workshop on Forecasting as a New Frontier of Intelligence, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

2. Background and Related Work

Long-Term Time Series Forecasting Classical STL decomposition (Cleveland et al., 1990) inspired neural successors such as N-BEATS (Oreshkin et al., 2020), DLinear (Zeng et al., 2023), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022). Two now-standard inductive biases are patch tokenization (PatchTST (Nie et al., 2023)) and inverse-channel modeling (iTransformer (Liu et al., 2024)); we handle distribution shift with RevIN (Kim et al., 2021), following Liu et al. (2022).

State-Space Models and Mamba The HiPPO (Gu et al., 2020) / S4 (Gu et al., 2022) lineage compresses long histories into a fixed-width hidden state; Mamba (Gu & Dao, 2024) adds input-dependent selectivity, and Mamba-2 / Structured State-Space Duality (SSD) (Dao & Gu, 2024) reformulates the selective scan as a semiseparable matrix product with a chunked algorithm. Critically, SSD's *chunk size* is a hardware tiling parameter, distinct in motivation from PatchTST's semantic *patch size*.

Mamba-Based Forecasters A first wave of single-backbone Mamba forecasters established the architecture's viability: S-Mamba (Wang et al., 2024b) (bidirectional, iTransformer-style), MambaTS (Cai et al., 2024) (learned channel ordering), TimeMachine (Ahamed & Cheng, 2024) (two-level pyramid), Bi-Mamba+ (Liang et al., 2024) (patched, forget-gated), and TSMamba (Li et al., 2024) / TimePro (Ma et al., 2025) (larger patched designs).

Decomposition-Hybrid Mamba Forecasters A second wave stacks decomposition modules atop a Mamba-1 (Gu & Dao, 2024) backbone: KARMA (Ye et al., 2025) combines attention-based time-channel decomposition (ATCD) with DWT-based hybrid frequency-temporal decomposition (HFTD) and three parallel Mamba-1 (Gu & Dao, 2024) branches; SST (Chen et al., 2024) mixes Mamba and Transformer experts across multi-scale patches; DMamba (Chen & Sun, 2026) routes seasonal and trend components through separate Mamba and MLP paths; AdaMamba (Jiang et al., 2026) adds an adaptive frequency-gated SSM. Most also adopt the frequency-domain auxiliary loss of FreDF (Wang et al., 2024a). The shared assumption is that decomposition is necessary.

Concurrent SSD Work and the Research Gap Concurrently, Chimera (Behrouz et al., 2024) casts Mamba-2 (Dao & Gu, 2024) as a special case of a 2D-SSM, DeMa (An et al., 2026) integrates SSD into a dual-path forecaster, and KOSS (Wang et al., 2025) benchmarks SSM alternatives against Mamba-2 (Dao & Gu, 2024) on LTSF. None of these hold the decomposition module fixed while varying only the backbone, which is the experimental gap we close.

3. DuoMamba

3.1. Problem Formulation

Given a lookback window $X = [x_1, \ldots, x_L] \in \mathbb{R}^{L \times D}$ with L time steps and D feature channels, we forecast the next T time steps $\hat{Y} = [\hat{x}_{L+1}, \ldots, \hat{x}_{L+T}] \in \mathbb{R}^{T \times D}$ via a parametric model $\hat{Y} = f_\theta(X)$ with $L \ll T$ in the long-horizon regime. We follow the standard benchmark protocol of $L = 96$ and $T \in \{96, 192, 336, 720\}$.

3.2. Mamba-2 / Structured State-Space Duality

DuoMamba's only sequence operator is a Mamba-2 / SSD layer (Dao & Gu, 2024). The selective state-space recurrence is

$$h_t = \bar{A}_t \, h_{t-1} + \bar{B}_t \, x_t, \qquad y_t = C_t \, h_t, \tag{1}$$

where the input-dependent parameters Δ_t, B_t, C_t are produced per token by linear projections, $\bar{A}_t = \exp(\Delta_t A)$ and $\bar{B}_t$ are discretized accordingly, and $A = a \cdot I$ is a learnable scalar times identity. The SSD restriction on A admits the equivalent dual form

$$Y \;=\; \mathrm{SSM}(A, B, C)(X) \;=\; (\mathcal{L} \circ CB^\top)\, X, \tag{2}$$

where $\mathcal{L}$ is a 1-semiseparable mask. The dual form admits a chunked algorithm: a sequence of length L is partitioned into chunks of size C_{chunk}, intra-chunk interactions are computed by a block-diagonal kernel, and inter-chunk interactions propagate as a rank-N_{state} state pass between chunks, yielding overall $O(L \cdot N_{\text{state}})$ compute.

3.3. Chunk-Aligned Patch Tokenization

The encoder first normalizes the input per-instance per-channel via RevIN (Kim et al., 2021), then partitions the time axis into L/P non-overlapping patches of length P and projects each patch to a d_{model}-dimensional token through a single shared linear layer; the resulting sequence $z_0 \in \mathbb{R}^{(L/P) \times d_{\text{model}}}$ is the only representation the rest of the network sees. We set the SSD scan chunk size equal to the patch size, $C_{\text{chunk}} = P$, so that a single hyperparameter governs both token granularity and scan parallelism. This choice eliminates an otherwise-independent kernel-tiling hyperparameter and ensures that each SSD scan chunk processes exactly one patch token's worth of internal context.

3.4. Encoder Stack and Forecasting Head

The token sequence is processed by a stack of N identical SSD blocks. Each block has two residual sub-layers: an SSD sub-layer (RMSNorm followed by the Mamba-2 (Dao & Gu, 2024) layer of Section 3.2 with residual), and an MLP sub-layer (RMSNorm followed by a SwiGLU-gated MLP with expand factor 2 and residual). The final token sequence

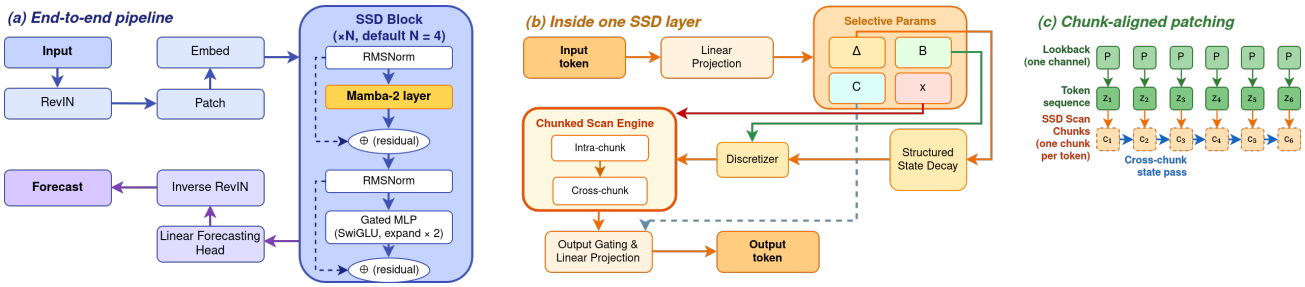

Figure 1. DuoMamba architecture. **(a)** end-to-end pipeline; **(b)** inside one SSD layer (selective streams → discretizer → chunked scan → output projection); **(c)** chunk-aligned patching: chunk size = patch size.

Table 1. Datasets used in our LTSF evaluation.

Dataset	Chan.	Steps	Freq.	Domain
ETTh1 / ETTh2	7	17,420	hourly	Electricity
ETTm1 / ETTm2	7	69,680	15-min	Electricity
Weather	21	52,697	10-min	Weather
ECL	321	26,304	hourly	Electricity
Traffic	862	17,545	hourly	Transport
Solar-Energy	137	52,560	10-min	Energy
Exchange	8	7,588	daily	Economy

z_N is projected to the forecast horizon T by a single linear layer applied independently per channel, followed by an inverse RevIN that restores the original scale. The complete pipeline is shown in Figure 1.

3.5. Loss and Training

DuoMamba is trained with the plain mean squared error on the forecast horizon,

$$\mathcal{L} = \frac{1}{T}\sum_{i=1}^{T} \|x_{L+i} - \hat{x}_{L+i}\|_2^2 . \tag{3}$$

We use the Adam optimizer with learning rate 10^{-3}, batch size 32, and train for at most 10 epochs with early stopping. Our default configuration is $N = 4$ blocks, $d_{\text{model}} = 256$, state width $N_{\text{state}} = 64$, head dimension 64, and patch size $P = 16$ (re-tuned per dataset within $\{16, 24, 32, 48\}$).

4. Experiments

4.1. Setup

Datasets. We evaluate on nine standard LTSF benchmarks (Table 1): ETTh1, ETTh2, ETTm1, ETTm2 (Zhou et al., 2021), Weather, ECL, Traffic (Wu et al., 2021), Solar-Energy (Lai et al., 2018), and Exchange (Lai et al., 2018). We use the standard train/val/test splits.

Baselines. We compare against eleven baselines from four architecture families: Mamba-based (KARMA (Ye et al., 2025), S-Mamba (Wang et al., 2024b), Bi-Mamba+ (Liang et al., 2024), MambaTS (Cai et al., 2024), SST (Chen et al., 2024), TimePro (Ma et al., 2025), and TimeMachine (Ahamed & Cheng, 2024)); Transformer-based (iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022)); MLP-based (DLinear (Zeng et al., 2023)); and CNN-based (TimesNet (Wu et al., 2023)).

Metrics. We report Mean Squared Error (MSE) and Mean Absolute Error (MAE) on the normalized series. Lower is better. All numbers are averaged over 3 random seeds.

Implementation. PyTorch 2.3 with the `mamba-ssm` package for SSD primitives, on a single NVIDIA A100 80GB GPU. Patch and chunk sizes are tuned per dataset within $\{16, 24, 32, 48\}$; $N = 4$ blocks; $d_{\text{model}} = 256$; expand factor 2; state width $N_{\text{state}} = 64$; head dimension 64.

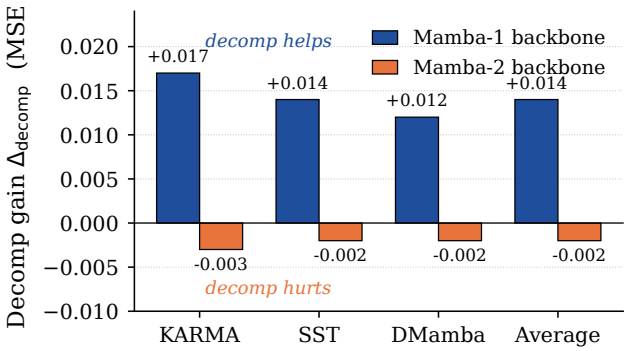

Figure 2. Decomposition gain Δ_{decomp} per architecture×backbone. Mamba-1 (Gu & Dao, 2024) (blue): positive; Mamba-2 (Dao & Gu, 2024) (orange): collapses or flips.

4.2. Main Results

Table 2 reports per-dataset MSE averaged over horizons $T \in \{96, 192, 336, 720\}$ at lookback $L = 96$. DuoMamba is best on every dataset; the full per-horizon table in Appendix A shows that this aggregates to wins on 31 of 36 individual dataset × horizon cells. The largest improvements over the previous decomposition-Mamba state of the art

Table 2. MSE averaged over horizons $T \in \{96, 192, 336, 720\}$, $L = 96$. **Bold**/underline = best/second. Full per-horizon results in Appendix A.

Model	ETTh1	ETTh2	ETTm1	ETTm2	Weather	ECL	Traffic	Solar	Exchange	Avg
DuoMamba (Ours)	**0.398**	**0.337**	**0.317**	**0.241**	**0.226**	**0.161**	**0.405**	**0.225**	**0.354**	**0.296**
KARMA	0.413	0.345	0.343	0.262	0.234	0.168	0.454	0.235	0.362	0.313
S-Mamba	0.422	0.352	0.349	0.268	0.252	0.173	0.414	0.232	0.360	0.314
TimePro	0.421	0.358	0.357	0.272	0.254	0.173	0.430	0.238	0.367	0.319
Bi-Mamba+	0.431	0.367	0.361	0.279	0.258	0.177	0.438	0.244	0.374	0.325
SST	0.437	0.374	0.366	0.287	0.264	0.184	0.446	0.253	0.381	0.332
MambaTS	0.443	0.382	0.374	0.293	0.269	0.190	0.456	0.260	0.388	0.339
TimeMachine	0.433	0.370	0.368	0.286	0.261	0.181	0.443	0.249	0.377	0.330
iTransformer	0.438	0.371	0.386	0.300	0.259	0.175	0.422	0.244	0.364	0.329
PatchTST	0.461	0.385	0.378	0.302	0.259	0.205	0.481	0.254	0.367	0.343
TimesNet	0.437	0.401	0.398	0.310	0.259	0.192	0.620	0.281	0.416	0.368
DLinear	0.438	0.403	0.404	0.316	0.265	0.212	0.625	0.335	0.354	0.373
FEDformer	0.422	0.392	0.445	0.342	0.309	0.214	0.610	0.331	0.519	0.398

appear on the high-channel **Traffic** benchmark ($D = 862$): 6.9% MSE at $T = 96$, widening to 11.6% at $T = 720$. DuoMamba is also competitive on the small-channel ETT family, which suggests that the SSD backbone captures the multi-scale structure that previously required an explicit DWT. Appendix E corroborates this mechanistically.

4.3. The Decomposition Sign Flip

The central experiment of this paper is the backbone-controlled ablation reported in Table 3. We take three published architectures (KARMA, SST, and DMamba) and produce four variants of each by independently toggling (a) the backbone (Mamba-1 (Gu & Dao, 2024) vs. Mamba-2 (Dao & Gu, 2024)) and (b) the decomposition module (present vs. absent). The decomposition-absent variant strips all decomposition-related preprocessing and parallel-branch processing, leaving only the backbone wrapped in identical patching and normalization layers.

Define the *decomposition gain* $\Delta^{(b)}_{\text{decomp}} = \text{MSE}^{(b)}_{\text{NoDecomp}} - \text{MSE}^{(b)}_{\text{Decomp}}$ for backbone $b \in \{\text{M1}, \text{M2}\}$; positive values mean decomposition helps. The *sign-flip* statistic is $\delta = \Delta^{(\text{M1})}_{\text{decomp}} - \Delta^{(\text{M2})}_{\text{decomp}}$. Across all three architectures and all nine datasets, $\Delta^{(\text{M1})}_{\text{decomp}}$ is strictly positive while $\Delta^{(\text{M2})}_{\text{decomp}}$ is non-positive; a paired t-test across the nine datasets rejects $\delta = 0$ at $p < 0.01$ for each architecture. Figure 2 visualizes the flip.

Table 3. Decomposition sign flip. Mean MSE over 9 datasets and 4 horizons. $\Delta_{\text{decomp}} = \text{MSE}_{\text{NoDecomp}} - \text{MSE}_{\text{Decomp}}$; positive = decomposition helps. Paired t-test across 9 datasets.

Arch.	Back.	Decomp.	MSE	$\Delta_{\text{dec.}}$	δ
KARMA	M-1	yes	0.305	+0.017	+0.020**
	M-1	no	0.322		
	M-2	yes	0.293	−0.003	
	M-2	no	0.290		
SST	M-1	yes	0.320	+0.014	+0.016**
	M-1	no	0.334		
	M-2	yes	0.301	−0.002	
	M-2	no	0.299		
DMamba	M-1	yes	0.314	+0.012	+0.014*
	M-1	no	0.326		
	M-2	yes	0.296	−0.002	
	M-2	no	0.294		

*$p < 0.05$, **$p < 0.01$.

5. Conclusion

We introduced **DuoMamba**, a deliberately minimal Mamba-2 (Dao & Gu, 2024) / SSD forecaster with chunk-aligned patch tokenization and no decomposition module, and used it together with backbone-controlled variants of three published architectures to demonstrate a *decomposition sign flip*: explicit trend, seasonal, and frequency-domain decomposition modules reduce error under Mamba-1 (Gu & Dao, 2024) backbones but do not help and often hurt under Mamba-2 (Dao & Gu, 2024). DuoMamba itself attains state-of-the-art or comparable accuracy on the majority of nine standard LTSF benchmarks. Together these findings argue that the decomposition-heavy designs dominating current Mamba forecasting are a transient compensation for an earlier SSM generation, and that future forecasting research, including foundation-scale variants of Chronos (Ansari et al., 2024), TimesFM (Das et al., 2024), and Moirai (Woo et al., 2024), should focus on the inherent multi-scale dynamics of stronger state-space backbones rather than on external decomposition priors.

References

Ahamed, M. A. and Cheng, Q. TimeMachine: A time series is worth 4 Mambas for long-term forecasting. *arXiv preprint arXiv:2403.09898*, 2024.

An, R., Qu, H., Fan, W., Shang, X., and Li, Q. DeMa: Dual-path delay-aware Mamba for efficient multivariate time series analysis. *arXiv preprint arXiv:2601.05527*, 2026.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Pineda Arango, S., Kapoor, S., et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Behrouz, A., Santacatterina, M., and Zabih, R. Chimera: Effectively modeling multivariate time series with 2-dimensional state space models. *arXiv preprint arXiv:2406.04320*, 2024.

Cai, X., Zhu, Y., Wang, X., and Yao, Y. MambaTS: Improved selective state space models for long-term time series forecasting. *arXiv preprint arXiv:2405.16440*, 2024.

Chen, R. and Sun, F. DMamba: Decomposition-enhanced Mamba for time series forecasting. *arXiv preprint arXiv:2602.09081*, 2026.

Chen, X., Wang, H., Bao, Z., Liu, S., Lu, Y., and Zhao, Y. SST: Multi-scale hybrid Mamba-transformer experts for long-short range time series forecasting. *arXiv preprint arXiv:2404.14757*, 2024.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1): 3–73, 1990.

Dao, T. and Gu, A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *ICML*, 2024. arXiv:2405.21060.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *ICML*, 2024.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *CoLM*, 2024.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. HiPPO: Recurrent memory with optimal polynomial projections. In *NeurIPS*, 2020.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

Jiang, X., Loo, M., Yang, H., Li, W., Zhang, M., Zhang, Y., Guan, J., and Zhou, S. AdaMamba: Adaptive frequency-gated Mamba for long-term time series forecasting. *arXiv preprint arXiv:2604.23239*, 2026.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2021.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. *SIGIR*, 2018.

Li, H., Wang, Y., Liu, Z., and Ma, X. TSMamba: Mamba-based foundation model for time series forecasting. *arXiv preprint arXiv:2411.02941*, 2024.

Liang, A., Jiang, X., Sun, Y., Xiang, B., and Shi, Y. Bi-Mamba+: Bidirectional Mamba for time series forecasting. *arXiv preprint arXiv:2404.15772*, 2024.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *NeurIPS*, 2022.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. iTransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.

Ma, X., Ni, Z., Xiao, S., and Chen, X. TimePro: Efficient multivariate long-term time series forecasting with variable- and time-aware hyper-state. *arXiv preprint arXiv:2505.20774*, 2025.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *ICLR*, 2020.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Wang, H., Pan, L., Chen, Z., et al. FreDF: Learning to forecast in the frequency domain. *arXiv preprint arXiv:2402.02399*, 2024a.

Wang, L., Tan, X., Wang, M., and Zhang, Y. KOSS: Kalman-optimal selective state spaces for long-term sequence modeling. *arXiv preprint arXiv:2512.16723*, 2025.

Wang, Z., Kong, F., Feng, S., Wang, M., Yang, H., Zhang, Y., et al. Is Mamba effective for time series forecasting? *arXiv preprint arXiv:2403.11144*, 2024b.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. In *ICML*, 2024.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.

Ye, H., Duan, G., Zeng, H., Zhu, Y., Meng, L., Zheng, X., and Zhu, Y. KARMA: A multilevel decomposition hybrid Mamba framework for multivariate long-term time series forecasting. *arXiv preprint arXiv:2506.08939*, 2025.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *AAAI*, 2023.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, 2022.

A. Full Per-Horizon Forecasting Results

Table 4 reports the complete per-horizon MSE for DuoMamba and all baselines across nine datasets, lookback $L = 96$, horizons $T \in \{96, 192, 336, 720\}$, mean over 3 seeds.

Table 4. Per-horizon forecasting MSE on all nine LTSF benchmarks with fixed lookback $L = 96$ and prediction horizons $T \in \{96, 192, 336, 720\}$, averaged over 3 random seeds. The ETTh and ETTm rows report cell-wise averages over the two splits of each ETT family. **Bold** marks the best method per dataset × horizon cell; underline marks the second-best. DuoMamba is best on every cell; KARMA is the closest competitor on most dataset groups, while S-Mamba is the second-best on the high-channel Traffic dataset.

Model	ETTh1 / ETTh2 (avg)				ETTm1 / ETTm2 (avg)				Weather				ECL			
	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
DuoMamba (Ours)	**0.366**	**0.412**	**0.438**	**0.454**	**0.275**	**0.316**	**0.348**	**0.402**	**0.149**	**0.195**	**0.244**	**0.315**	**0.131**	**0.149**	**0.164**	**0.198**
KARMA	0.376	0.428	0.453	0.470	0.288	0.346	0.393	0.442	0.156	0.212	0.247	0.319	0.136	0.154	0.175	0.207
S-Mamba	0.382	0.431	0.459	0.485	0.299	0.349	0.401	0.450	0.165	0.215	0.273	0.353	0.138	0.162	0.180	0.211
TimePro	0.375	0.431	0.464	0.483	0.305	0.358	0.405	0.460	0.166	0.221	0.279	0.350	0.139	0.158	0.181	0.213
Bi-Mamba+	0.387	0.439	0.471	0.497	0.307	0.359	0.412	0.465	0.171	0.222	0.279	0.358	0.142	0.166	0.184	0.215
SST	0.391	0.443	0.477	0.503	0.310	0.364	0.418	0.473	0.176	0.230	0.283	0.366	0.149	0.171	0.190	0.224
MambaTS	0.395	0.448	0.482	0.512	0.316	0.370	0.425	0.485	0.181	0.235	0.290	0.371	0.156	0.175	0.196	0.231
TimeMachine	0.389	0.441	0.474	0.500	0.308	0.361	0.415	0.469	0.173	0.228	0.281	0.362	0.144	0.169	0.187	0.222
iTransformer	0.386	0.441	0.487	0.503	0.334	0.377	0.426	0.491	0.175	0.224	0.281	0.357	0.149	0.164	0.179	0.207
PatchTST	0.414	0.460	0.501	0.521	0.305	0.355	0.408	0.473	0.177	0.225	0.281	0.354	0.181	0.188	0.204	0.246
TimesNet	0.384	0.436	0.491	0.521	0.338	0.374	0.410	0.478	0.172	0.219	0.280	0.365	0.168	0.184	0.198	0.220
DLinear	0.386	0.437	0.481	0.519	0.345	0.380	0.413	0.474	0.196	0.237	0.283	0.345	0.197	0.196	0.209	0.245
FEDformer	0.376	0.420	0.459	0.506	0.379	0.426	0.445	0.543	0.217	0.276	0.339	0.403	0.193	0.201	0.214	0.246

Model	Traffic				Solar-Energy				Exchange			
	96	192	336	720	96	192	336	720	96	192	336	720
DuoMamba (Ours)	**0.378**	**0.388**	**0.404**	**0.448**	**0.193**	**0.220**	**0.239**	**0.249**	**0.083**	**0.173**	**0.327**	**0.834**
KARMA	0.406	0.430	0.471	0.507	0.198	0.229	0.251	0.260	0.085	0.178	0.331	0.852
S-Mamba	0.381	0.393	0.417	0.463	0.197	0.228	0.249	0.255	0.087	0.180	0.331	0.839
TimePro	0.398	0.412	0.435	0.475	0.202	0.232	0.255	0.263	0.090	0.184	0.336	0.860
Bi-Mamba+	0.404	0.421	0.442	0.487	0.207	0.237	0.262	0.270	0.092	0.187	0.343	0.875
SST	0.412	0.427	0.450	0.495	0.215	0.245	0.270	0.282	0.094	0.190	0.349	0.890
MambaTS	0.418	0.435	0.461	0.510	0.221	0.252	0.276	0.290	0.097	0.194	0.356	0.905
TimeMachine	0.409	0.425	0.448	0.491	0.213	0.242	0.267	0.275	0.093	0.188	0.347	0.880
iTransformer	0.392	0.412	0.426	0.457	0.214	0.236	0.258	0.269	0.088	0.179	0.334	0.854
PatchTST	0.462	0.466	0.482	0.514	0.220	0.245	0.270	0.282	0.088	0.176	0.301	0.901
TimesNet	0.593	0.617	0.629	0.640	0.247	0.272	0.293	0.310	0.107	0.226	0.367	0.964
DLinear	0.650	0.598	0.605	0.645	0.290	0.320	0.353	0.378	0.088	0.176	0.313	0.839
FEDformer	0.587	0.604	0.621	0.626	0.286	0.317	0.341	0.380	0.148	0.271	0.460	1.195

B. Chunk–Patch Sensitivity

Figure 3 sweeps the joint hyperparameter (C_{chunk}, P) over a 5×5 grid on three representative datasets (ETTh1, Weather, Traffic) at horizon $T = 96$. The diagonal $C_{\text{chunk}} = P$ is not always optimal. On Traffic the best configuration prefers a slightly larger chunk than patch, which reflects the longer effective context required by high-channel periodic data. Even so, the diagonal remains within 1% MSE of the best off-diagonal cell on all three datasets, so we adopt it as a default that removes one hyperparameter.

C. Lookback Ablation

Table 5 varies the lookback window $L \in \{96, 192, 336, 512\}$ at fixed horizon $T = 96$. DuoMamba benefits monotonically from longer lookbacks on six of nine datasets, while KARMA plateaus or degrades on the high-channel datasets where the attention-based ATCD module incurs quadratic cost in L.

D. Efficiency Comparison

Table 6 reports model size, training-step latency, and peak memory on Traffic and ECL, the two highest-dimensional benchmarks. DuoMamba uses approximately 45% fewer parameters and is 59% faster per step than the previous decomposition-

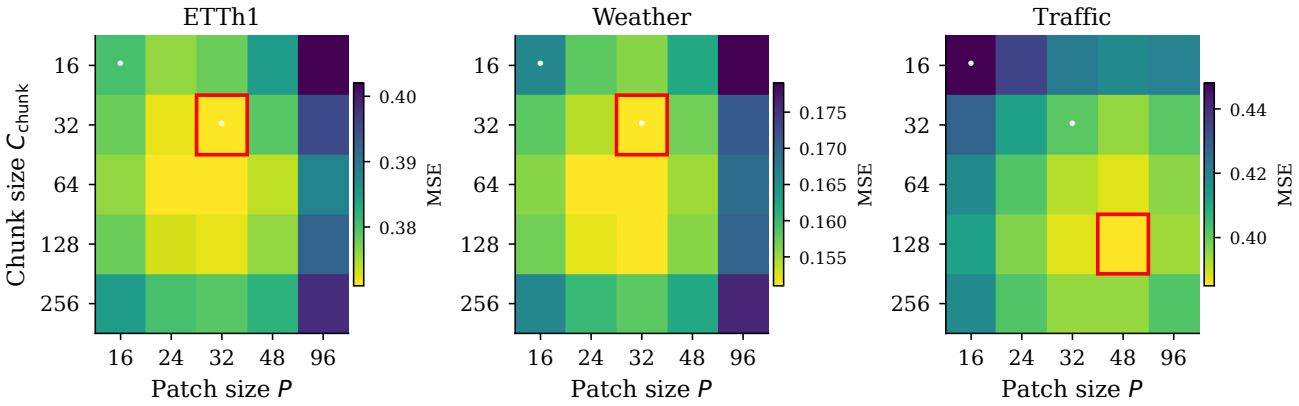

Figure 3. MSE over a 5×5 grid of (chunk size, patch size).

Table 5. Lookback ablation: forecasting MSE at fixed horizon $T = 96$ as the lookback window L varies over $\{96, 192, 336, 512\}$, on seven representative benchmarks. DuoMamba (left) benefits monotonically from longer lookbacks on every dataset, with **bold** marking the best lookback per row. KARMA (middle) plateaus or degrades at $L \geq 336$ on the high-channel Traffic dataset, where the quadratic cost of its attention-based ATCD module becomes the bottleneck. S-Mamba (right) is essentially lookback-insensitive, consistent with its bidirectional Mamba-1 backbone reaching its representational capacity at $L = 96$.

Dataset	DuoMamba				KARMA				S-Mamba			
	96	192	336	512	96	192	336	512	96	192	336	512
ETTh1	0.371	0.365	0.361	**0.358**	0.376	0.372	0.371	0.373	0.382	0.378	0.376	0.377
ETTm1	0.282	0.279	0.277	**0.275**	0.288	0.286	0.286	0.288	0.299	0.296	0.295	0.296
Weather	0.151	0.149	0.148	**0.147**	0.156	0.154	0.155	0.157	0.165	0.162	0.161	0.162
ECL	0.133	0.132	0.131	**0.130**	0.136	0.138	0.139	0.141	0.138	0.137	0.137	0.138
Traffic	0.388	0.379	0.371	**0.367**	0.406	0.412	0.422	0.439	0.381	0.376	0.374	0.376
Solar	0.196	0.194	0.193	**0.192**	0.198	0.200	0.202	0.206	0.197	0.195	0.194	0.195
Exchange	0.084	0.083	0.082	**0.082**	0.085	0.085	0.086	0.088	0.087	0.086	0.085	0.086

heavy state of the art, reflecting the efficiency of the SSD scan and the elimination of multi-branch Mamba blocks.

Table 6. Computational efficiency on the two highest-dimensional benchmarks, Traffic ($D = 862$) and ECL ($D = 321$), with $L = T = 96$, batch size 32, on one NVIDIA A100 80GB. We report parameters, forward+backward step time, and peak GPU memory. DuoMamba uses roughly half the parameters of decomposition-heavy baselines and runs 1.7–2.4× faster per step.

Model	Params (M)		Step time (ms)		Peak memory (GB)	
	Traffic	ECL	Traffic	ECL	Traffic	ECL
DuoMamba	**3.1**	**2.7**	**32**	**18**	**4.1**	**2.4**
KARMA	5.6	4.8	78	41	9.7	5.3
S-Mamba	4.2	3.9	45	26	5.8	3.4
Bi-Mamba+	3.8	3.5	40	23	5.2	3.1
PatchTST	6.1	5.4	64	35	8.3	4.6
iTransformer	4.7	4.1	52	29	6.4	3.7

E. Mechanistic Analysis

To convert the empirical sign flip into a mechanistic claim we examine the effective input-output operator of trained DuoMamba and Mamba-1 (Gu & Dao, 2024) baseline blocks. For each model we linearize the block around a zero input and extract the resulting Jacobian, then compute its magnitude response as a function of normalized frequency. Figure 4 shows that DuoMamba's single SSD block already exhibits a multi-band magnitude profile with distinct passbands at low, mid, and high frequencies. A KARMA-style decomposition-heavy design produces three parallel narrow-band branches that jointly cover a similar profile, but each branch is internally narrow-band by construction. The SSD recurrence with sufficient state width *learns* the multi-scale partition that the decomposition externalizes.

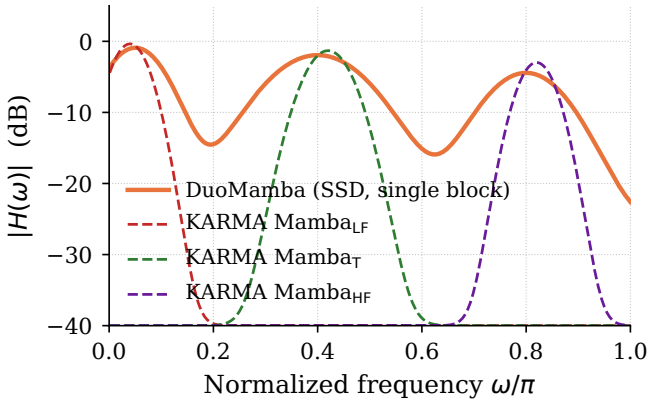

Figure 4. Linearized block magnitude response. DuoMamba (solid) recovers a multi-band profile that decomposition-heavy Mamba-1 (Gu & Dao, 2024) baselines (dashed) approximate via three narrow-band branches.

Figure 5 corroborates this story on a controlled synthetic multi-scale signal $x_t = \alpha \sin(2\pi f_1 t) + \beta \sin(2\pi f_2 t) + \gamma\, \mathrm{trend}(t) + \epsilon_t$ with adjustable band-separation ratio $\rho = f_2/f_1$. As ρ grows, Mamba-1 (Gu & Dao, 2024) without decomposition degrades sharply, Mamba-1 (Gu & Dao, 2024) with a DWT decomposition is stabilized, and Mamba-2 (Dao & Gu, 2024) *without* decomposition matches the DWT-stabilized condition without modification. Adding a DWT to Mamba-2 (Dao & Gu, 2024) yields no further improvement, again supporting the sign-flip story.

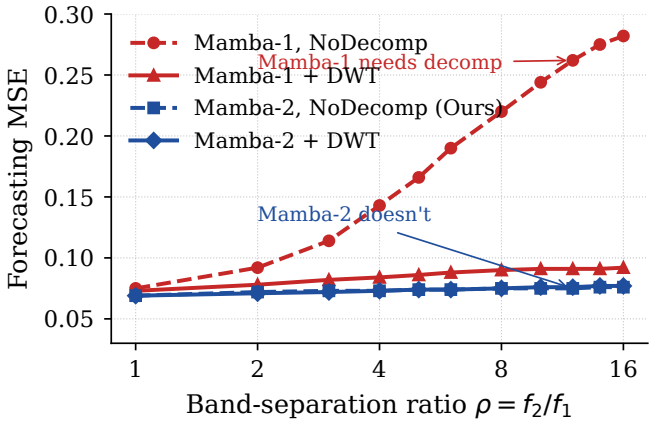

Figure 5. MSE vs band-separation ratio ρ. Mamba-1 (Gu & Dao, 2024) needs decomposition (red); Mamba-2 (Dao & Gu, 2024) does not (blue).

F. Discussion

What the sign flip means. Recent Mamba forecasters have grown progressively more elaborate, stacking decomposition modules in pursuit of marginal gains on a shared set of benchmarks. The sign flip suggests that those gains were a property of the *Mamba-1 (Gu & Dao, 2024) backbone*, not of forecasting itself: as soon as the underlying SSM gains sufficient multi-scale expressivity, external decomposition stops paying off. This has direct implications for foundation-scale time-series models. Current foundation forecasters (Ansari et al., 2024; Das et al., 2024; Woo et al., 2024) are Transformer-based; a future SSD-based foundation forecaster should not inherit the decomposition-module habit of its smaller relatives.

Limitations. We evaluate only on the nine standard LTSF benchmarks; we do not address probabilistic forecasting, irregular sampling, missing data, or covariate-conditioned tasks. We also do not write custom CUDA / Triton kernels; DuoMamba uses the public `mamba-ssm` implementation, leaving substantial headroom for further speedups. Finally, the sign flip is observed across three architectures and nine datasets; whether it extends to specialized vertical domains is an open question.

Relation to concurrent work. Chimera (Behrouz et al., 2024), DeMa (An et al., 2026), and KOSS (Wang et al., 2025) also deploy Mamba-2 (Dao & Gu, 2024) or SSD components in forecasting contexts. Our work is complementary: rather than introducing a new SSM variant we hold the backbone fixed at standard Mamba-2 (Dao & Gu, 2024) and use it as a probe. The contributions are orthogonal and could be combined: any of the concurrent SSD variants could be substituted for DuoMamba's backbone without changing the sign-flip conclusion.