

---

# Analyzing State-Space Dynamics in Mamba for Financial Time Series Forecasting

---

Sacha Liechti<sup>\*1</sup> Arthur Windels<sup>\*1</sup>

## Abstract

We evaluate the Mamba architecture for next-day financial volatility prediction and investigate the internal dynamics of its selective state spaces. Financial time series are characterized by low signal-to-noise ratios, making robust forecasting notoriously difficult. By framing the problem around next-day log-range prediction, we demonstrate how Mamba processes noisy sequences and compare its predictive performance against strong statistical baselines. Crucially, we move beyond standard performance metrics to provide a detailed empirical analysis of Mamba’s internal selective state-space (SSM) representations. Our investigations into impulse responses, linear probing of future targets, and context warm-up reveal that early-layer states are highly reactive to volatility shocks and carry the most predictive signal, while the training objective fundamentally alters the model’s reliance on state initialization.

## 1. Introduction

Deep sequence models have achieved remarkable success across diverse domains, yet applying them to financial time series remains a formidable challenge. Equity price series are inherently noisy at short horizons and are subject to rapid regime shifts. The efficient market hypothesis posits that daily directional returns are largely unpredictable, driven primarily by unforeseen information shocks. Consequently, a constant “zero log-return” baseline is notoriously difficult to outperform in rigorous out-of-sample testing.

To conduct a meaningful evaluation of deep sequence representations, we therefore pivot from directional return forecasting to volatility forecasting. Unlike returns, volatility exhibits strong temporal persistence and clustering, offering a higher signal-to-noise ratio. We focus on predicting the

next-day log-range using Open-High-Low-Close-Volume (OHLCV) data, a task that requires a model to effectively synthesize multi-scale historical context.

Recently, the Mamba architecture (Gu & Dao, 2023), which relies on selective state-space models (SSMs), has emerged as a powerful alternative to Transformers. By updating a latent state causally, Mamba achieves linear-time complexity while its input-dependent selectivity allows it to adaptively filter noise and retain long-term dependencies. While Mamba’s computational efficiency is well-documented, its internal representational dynamics (specifically how it allocates its state capacity when confronted with noisy, financial data) remain poorly understood.

In this paper, we bridge the gap between predictive performance and internal model interpretability. We adapt Mamba for multivariate financial volatility forecasting and benchmark it against established financial models. We then analyze how the SSM states encode predictive signals.

Our main contributions are as follows:

- **Volatility Forecasting with SSMs:** We adapt Mamba for next-day log-range prediction on a broad panel of daily stock data, comparing its efficacy against strong econometric baselines like the Heterogeneous Autoregressive (HAR) model.
- **State Representation Analysis:** We probe the internal SSM dynamics, demonstrating via impulse responses and linear probing that lower-layer states are more sensitive to instantaneous volatility shocks and contain the highest linearly-decodable signal for future targets.
- **Initialization and Warm-up Dynamics:** We quantify a substantial “warm-up” effect, showing that prepending realistic historical context at evaluation significantly reduces early-position error, and that this dependence is heavily influenced by the choice of training loss masking.

## 2. Related Work

**Financial Volatility Forecasting.** Modeling the temporal dynamics of volatility is a cornerstone of quantitative

---

<sup>\*</sup>Equal contribution <sup>1</sup>Department of Computer Science, ETH Zürich, Zürich, Switzerland. Correspondence to: Sacha Liechti <mail>, Arthur Windels <awindels@student.ethz.ch>.

*Submission to the Deep Learning 2025 class., Zurich, Switzerland. 2025. Copyright 2025 by the author(s).*

finance. Classical approaches rely on autoregressive conditional heteroskedasticity, such as GARCH (Bollerslev, 1986), or simple Exponentially Weighted Moving Averages (EWMA) popularized by RiskMetrics. To capture the multi-scale, long-memory nature of volatility, the Heterogeneous Autoregressive (HAR) model (Corsi, 2009) constructs features from daily, weekly, and monthly averages. These statistical models remain exceptionally strong baselines because they explicitly encode the stylized facts of financial markets, such as volatility clustering.

**Deep Learning for Time Series.** In recent years, deep learning models have been extensively applied to time series forecasting. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks naturally handle sequential data but often struggle to capture very long-range dependencies due to vanishing gradients. Transformers largely solved this issue via self-attention, but their  $O(N^2)$  computational complexity makes them expensive for long context windows, and they are prone to overfitting on highly noisy financial data.

**State-Space Models.** Structured State-Space Models (SSMs), originating from S4, offer a mathematically rigorous way to handle long sequences with linear scaling. The introduction of Mamba (Gu & Dao, 2023) added a critical hardware-aware selective mechanism, allowing the model to dynamically filter irrelevant inputs. While recent works have begun adapting SSMs for general time series (Ma et al., 2024), there is a distinct lack of literature analyzing *how* these models internally process noisy financial data. Our work addresses this gap by directly interrogating the hidden state dynamics, connecting the model’s architecture to its empirical forecasting behavior.

### 3. Problem Setup and Methodology

To rigorously evaluate the internal representations of Mamba on financial time series, we establish a controlled volatility forecasting framework. This section details our data formulation, the adaptation of the Mamba architecture, our training protocol, and the baseline models used for comparison.

#### 3.1. Data and Feature Engineering

**Dataset.** We use a panel of *daily* stock observations (Oymak, 2025) (one time series per symbol) containing Open, High, Low, Close, and Volume (OHLCV) fields. Each training example is a fixed-length window of  $L \in \{32, 64\}$  consecutive trading days sampled from a single stock. After cleaning the dataset (removing NaNs and delisted S&P 500 components), we retain 401 stocks with daily data ranging from November 2005 to November 2025. Let  $H_{i,t}$ ,  $L_{i,t}$ ,

and  $V_{i,t}$  denote the daily high, low, and volume for stock  $i$  at time  $t$ , respectively.

**Target and Input Features.** Our target is the next-day log-range, a proxy for realized volatility. For each day  $t$  in the window, we construct a 3-dimensional feature vector  $x_{i,t} \in \mathbb{R}^3$  defined by the log-range, volume change, and log-return:

$$\text{range}_{i,t} = \log\left(\frac{H_{i,t}}{L_{i,t}}\right), \quad (1)$$

$$\text{volchg}_{i,t} = \log\left(\frac{V_{i,t}}{V_{i,t-1}}\right), \quad (2)$$

$$\text{ret}_{i,t} = \log\left(\frac{P_{i,t}^{\text{adj}}}{P_{i,t-1}^{\text{adj}}}\right), \quad (3)$$

where  $P_{i,t}^{\text{adj}}$  in Eq. 3 denotes the adjusted close price, which accounts for corporate actions such as dividend distributions and stock splits.

#### 3.2. Mamba Architecture for Forecasting

We use the Mamba architecture (Gu & Dao, 2023), a sequence model based on selective state-space models (SSMs) that maintains linear-time complexity with respect to sequence length. Our implementation adapts the standard Mamba block for multivariate time-series forecasting.

**Selective state-space dynamics.** Each Mamba block combines a local convolutional pathway with a selective SSM pathway. At time step  $t$ , the latent state  $h_t$  evolves causally according to:

$$h_{t+1} = A(x_t) h_t + B(x_t) x_t, \quad (4)$$

$$y_t = C(x_t) h_t. \quad (5)$$

Unlike standard time-invariant SSMs, the parameters  $(A, B, C)$  depend dynamically on the current input  $x_t$ . This selectivity allows the model to adaptively retain or forget information, enabling multi-timescale memory crucial for filtering financial noise. Following hyperparameter tuning on a validation split, our final model consists of two Mamba blocks with a hidden state dimension of  $d_{\text{model}} = d_{\text{state}} = 8$  and a batch size of 256.

#### 3.3. Training Supervision and Loss Design

**Supervision format.** We train the model on fixed-length rolling windows using a *many-to-many* teacher-forced regime. For a window  $X_{i,t-L+1:t}$  of length  $L$ , the model outputs a sequence of next-day volatility predictions  $\hat{Y}_{i,t-L+1:t}$ . Computation is strictly *causal*: internal states are updated using the *true input*  $x_{i,\tau}$  at each step, preventing information leakage. States are reset between windows to

ensure recurrence arises solely from within-window dynamics.

**Burn-in loss masking.** To reduce sensitivity to arbitrary state initializations (i.e., zero-initialized hidden states) and delay credit assignment, we apply *burn-in masking*. By ignoring the initial  $b$  steps of a window  $T$ , the masked Mean Squared Error (MSE) loss is:

$$\mathcal{L}_{\text{burn-in}} = \frac{1}{T-b} \sum_{\tau=b+1}^T (\hat{y}_\tau - y_\tau)^2. \quad (6)$$

We experiment with  $b \in \{0, \lfloor T/2 \rfloor, T-1\}$ , corresponding to full supervision, half-window burn-in, and last-step-only supervision.

### 3.4. Evaluation Metrics and Baselines

To ensure meaningful representation analysis, we benchmark Mamba against strong statistical baselines evaluated under identical time-based splits and horizon constraints.

**Baselines.** We compare against: (i) a **Last-value** heuristic ( $\widehat{RV}_{i,t+1|t} = RV_{i,t}$ ); (ii) **Exponentially Weighted Moving Averages (EWMA)**, including a short-horizon span (5 days) and a RiskMetrics-style decay ( $\lambda = 0.94$ ) (J.P. Morgan, 1996); and (iii) the **Heterogeneous Autoregressive (HAR)** model (Corsi, 2009), which captures multi-scale long-memory effects.

**Evaluation metric.** Let  $y_{s,t}$  denote the target variance for symbol  $s$  at time  $t$ , and  $\hat{y}_{s,t}$  the prediction. To prevent highly volatile stocks from dominating the evaluation, we report a balanced Mean Squared Error with *equal per-symbol weighting*:

$$\text{MSE}_{\text{balanced}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{1}{N_s} \sum_{t:(s,t) \in \mathcal{D}_s} (\hat{y}_{s,t} - y_{s,t})^2, \quad (7)$$

where  $\mathcal{S}$  is the set of test symbols and  $N_s$  is the number of evaluated points for symbol  $s$ .

## 4. Experiments and Predictive Performance

In this section, we evaluate Mamba’s predictive capabilities on the next-day log-range forecasting task. We measure the balanced Mean Squared Error ( $\text{MSE}_{\text{balanced}}$ ) of various Mamba configurations against the statistical baselines outlined in Section 3.

**Main Results.** Table 1 presents a condensed comparison of Mamba’s performance under different sequence lengths and training loss masks (the comprehensive results table containing all configurations is available in Appendix A).

Mamba consistently outperforms naive heuristics (Persist) and short-memory models (EWMA). Most notably, under full-sequence supervision (`seq32_full` and `seq64_full`), Mamba achieves predictive accuracy that surpasses the strong, long-memory HAR baseline. This confirms that the selective state-space architecture can effectively capture the complex, multi-scale volatility clustering inherent to financial markets.

Table 1. Summarized Test MSE for next-day log-range prediction. A positive  $\Delta\%$  indicates an improvement over the respective HAR baseline for that specific evaluation window. Full results are provided in Appendix A.

MODEL CONFIG	$\text{MSE}_{\text{balanced}}$	$\Delta\%$ vs HAR
<b>BASELINES (<math>L = 32</math>)</b>		
PERSIST	$1.847 \times 10^{-4}$	−45.3%
EWMA ( $\lambda = 0.94$ )	$1.603 \times 10^{-4}$	−26.1%
HAR (1,5,22)	$1.271 \times 10^{-4}$	—
<b>MAMBA (<math>L = 32</math>)</b>		
SEQ32_FULL	<b><math>1.209 \times 10^{-4}</math></b>	<b>+4.9%</b>
SEQ32_BURNIN16	$1.284 \times 10^{-4}$	−1.1%
SEQ32_BURNIN31	$2.252 \times 10^{-4}$	−77.2%
<b>MAMBA (<math>L = 64</math>)</b>		
SEQ64_FULL	<b><math>1.193 \times 10^{-4}</math></b>	<b>+4.7%</b>

**The Impact of Training Objectives.** While Mamba demonstrates state-of-the-art predictive capabilities, Table 1 also highlights a critical vulnerability: the model’s performance is highly sensitive to the training objective. When aggressive burn-in masking is applied—such as in `seq32_burnin31`, where only the final step of the window is supervised—performance degrades severely, underperforming even the simplest baselines.

This divergence raises a fundamental question about how Mamba utilizes its internal capacity to process financial data. Is the model learning robust representations, or is it highly dependent on how supervision shapes its state initializations? To contextualize these forecasting results and understand the mechanisms driving Mamba’s predictions, we transition from standard performance metrics to a rigorous empirical analysis of its selective state-space dynamics.

## 5. Analysis of State Dynamics

The predictive results in Section 4 show that Mamba is a capable volatility forecaster, but its performance is highly sensitive to the training objective (specifically, burn-in masking). To understand this sensitivity and how Mamba processes financial noise, we explicitly unroll the model under teacher-forced inputs to analyze the internal SSM states, which carry the model’s long-term memory.

Let  $h_{t,\ell} \in \mathbb{R}^d$  denote the flattened SSM state at time  $t$  and

layer  $\ell$ .

### 5.1. Impulse Response of SSM States

**Motivation.** Financial volatility is characterized by rapid shocks that gradually decay. To see if Mamba’s internal representations reflect this financial reality, we analyze how an artificial shock to the input propagates through the hidden states over time.

**Experiment.** For a fixed feature index  $j$  (here: `log_range`), we perturb a single time step  $\tau$  in a window:

$$x_t^{(\epsilon)} = \begin{cases} x_t + \epsilon e_j, & t = \tau, \\ x_t, & \text{otherwise,} \end{cases}$$

and measure the (per- $\epsilon$ ) relative change in SSM states for subsequent lags  $k \geq 0$ :

$$\text{IR}_{\ell,\tau}(k) = \mathbb{E} \left[ \frac{\|h_{\tau+k,\ell}^{(\epsilon)} - h_{\tau+k,\ell}\|_2}{\|h_{\tau+k,\ell}\|_2 + \epsilon} \right] \frac{1}{|\epsilon|}. \quad (8)$$

We average (8) over sampled test windows and visualize  $\log_{10}(\mathbb{E}[\text{IR}_{\ell,\tau}(k)])$ .

**Result.** Figure 1 shows that layer  $L0$  exhibits substantially higher sensitivity to a `log_range` impulse than layer  $L1$  across all measured lags (approximately a  $2.5\text{--}3\times$  difference in linear scale). This indicates a clear temporal hierarchy: lower layers act as high-frequency filters that strongly retain instantaneous volatility shocks, while higher layers abstract these shocks into a more stable, less reactive representation.

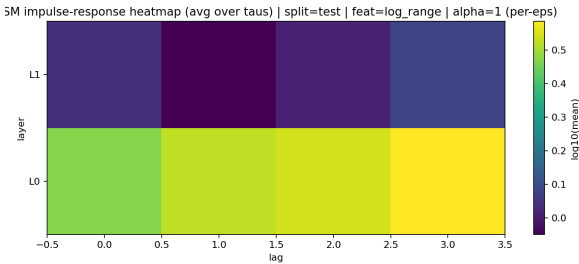


Figure 1. SSM impulse-response heatmap on the test split for feature `log_range`. Lower layers ( $L0$ ) are significantly more sensitive to instantaneous shocks.

### 5.2. Linear Probing of Future Targets

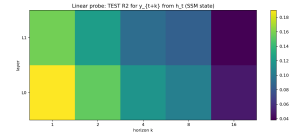
**Motivation.** If lower layers are more reactive to shocks, where is the actual predictive signal stored? We test whether  $h_{t,\ell}$  contains linearly-decodable information about future volatility.

**Experiment.** We fit ridge-regression probes to predict the target at horizon  $k$ :

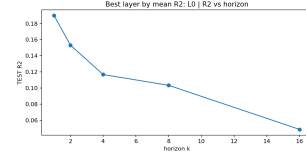
$$\hat{y}_{t+k} = w_{\ell,k}^\top h_{t,\ell} + b_{\ell,k}, \quad (9)$$

reporting the test  $R^2$  for horizons  $k \in \{1, 2, 4, 8, 16\}$ .

**Result.** Figure 2 shows two trends. First, predictability decreases with horizon, which is expected for causal sequence models. Second, and more importantly, the lower layer ( $L0$ ) consistently outperforms  $L1$  at *all* horizons. This suggests that for financial volatility, the raw, reactive traces of recent shocks (stored in  $L0$ ) are more linearly useful for forecasting than the deeper abstractions formed in  $L1$ . This aligns with classical financial theory, where recent realized variance is the strongest predictor of near-term future variance.



(a) Test  $R^2$  heatmap across layers and horizons.



(b) Best layer ( $L0$ ) test  $R^2$  vs. horizon.

Figure 2. Linear probes predicting  $y_{t+k}$  from  $h_{t,\ell}$ . The most accessible predictive signal is concentrated in the earlier-layer SSM state.

### 5.3. Initialization and Warm-Up Effects

**Motivation.** During training, windows are processed with zero-initialized states. However, in a continuous financial time series, the “true” state is a running accumulation of historical context. We quantify how this artificial reset affects performance and whether providing historical context (“warm-up”) alters the state manifold.

**Experiment.** We evaluate the trained model on test windows while *prepending* an additional chronological context of length  $K$  from the same ticker, scoring only the last  $T$  predictions. We report the per-position MSE within the scored window.

**Result.** Figure 3 shows a monotone trend: larger context lengths  $K$  yield uniformly lower MSE, with the largest gains at early timesteps. We confirm this is a state-manifold

effect by tracking the deviation of warmed-up states against zero-initialized states (Figure 4), showing that context drives the SSM away from the zero-init transient toward a more predictive steady-state distribution.

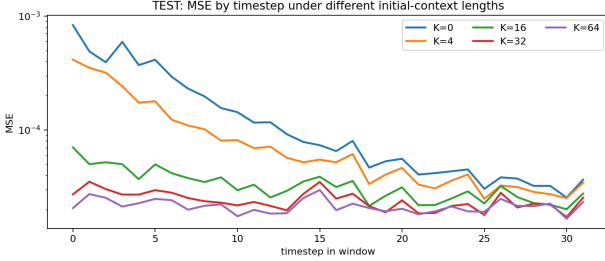


Figure 3. Test MSE as a function of position within the scored window for different prepended context lengths  $K$ . Larger contexts reduce early-timestep error.

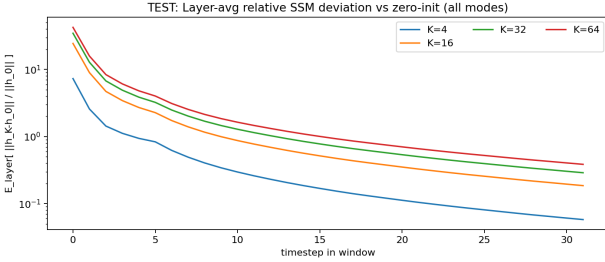


Figure 4. Layer-averaged relative deviation between SSM states with context length  $K$  and the zero-initialized trajectory ( $K=0$ ).

**Addressing the Data Leakage Hypothesis.** One might argue that providing the actual chronological history ( $x_{t-K}, \dots, x_{t-1}$ ) improves performance simply because it provides highly correlated, recent data (a form of test-time leakage), rather than genuinely “warming up” the recurrent state dynamics. However, if this were purely a leakage effect, the improvement would be uniform regardless of how the model was trained. As shown in Table 2, models trained with aggressive burn-in (where early steps are ignored) show massive sensitivity to the lack of context (Ratio = 9.77), whereas fully-supervised models are relatively robust (Ratio = 1.21). This confirms that the warm-up effect is deeply tied to how the training objective forces the SSM to utilize its hidden state, rather than just naive data leakage.

## 6. Conclusion and Future Work

**Conclusion.** In this paper, we evaluated the Mamba architecture for next-day financial volatility forecasting, demonstrating that structured state-space models can effectively

Table 2. Warm-up sensitivity at the first scored position ( $\tau=0$ ) for  $L = 32$ . Ratio =  $\text{MSE}_{K=0}/\text{MSE}_{K=64}$ .

TRAIN LOSS	$\text{MSE}_{K=0}$	$\text{MSE}_{K=64}$	RATIO
SEQ32_FULL	$9.864 \times 10^{-5}$	$8.181 \times 10^{-5}$	1.21
SEQ32_BURNIN16	$2.275 \times 10^{-4}$	$7.811 \times 10^{-5}$	2.91
SEQ32_BURNIN31	$7.926 \times 10^{-4}$	$8.113 \times 10^{-5}$	9.77

navigate the low signal-to-noise ratio of financial markets. When trained with full-sequence supervision, Mamba outperformed strong traditional baselines, including the long-memory HAR model and EWMA heuristics.

Beyond predictive accuracy, our primary contribution lies in opening the “black box” of the Mamba architecture to analyze its internal selective state-space dynamics. Through impulse-response experiments and linear probing, we established a clear representational hierarchy: lower-layer states act as high-frequency filters that react strongly to instantaneous volatility shocks and contain the most accessible predictive signal, while higher layers abstract this information. Furthermore, we identified a substantial context warm-up effect. Prepending realistic historical context during evaluation drastically reduces early-position errors by shifting the SSM states from an artificial zero-initialized transient to a predictive steady-state manifold. Crucially, we showed that the magnitude of this warm-up dependence is dictated by the training objective, with aggressive burn-in masking leading to fragile initializations.

**Limitations.** Our experiments utilize a single historical period with a chronological split to prevent look-ahead bias. Because financial markets are inherently non-stationary and subject to sudden exogenous shocks, findings regarding hidden-state dynamics are conditional on this chosen period. Additionally, while we controlled for data leakage in our warm-up experiments, the true generalization of these state dynamics under severe distribution shift remains an open question.

**Future Work.** These findings open several promising directions for future research. First, while we focused on daily data, Mamba’s linear-time complexity makes it uniquely suited for ultra-high-frequency (tick-level) financial data, where longer sequence lengths might fully exploit the SSM’s long-range memory capabilities. Second, our warm-up analysis suggests that alternative state-initialization strategies—such as learning a steady-state prior rather than defaulting to zero-initialization—could significantly improve out-of-sample robustness. Finally, future work should investigate how Mamba’s input-dependent selectivity parameters ( $A, B, C$ ) dynamically adapt during macro-economic regime shifts, potentially offering a new mechanism for detecting concept drift in financial markets.

## References

- Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986. doi: 10.1016/0304-4076(86)90063-1. URL [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1).
- Corsi, F. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009. doi: 10.1093/jjfinec/nbp001. URL <https://doi.org/10.1093/jjfinec/nbp001>.
- Engle, R. F. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982. doi: 10.2307/1912773. URL <https://doi.org/10.2307/1912773>.
- Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2023. URL <https://arxiv.org/abs/2312.00752>.
- J.P. Morgan. Riskmetrics™ — technical document. Technical report, J.P. Morgan, 1996. URL <https://www.mscl.com/documents/10199/5915b101-4206-4ba0-aee2-3449d5c7e95a>. Introduces the EWMA volatility model commonly used with daily decay  $\lambda = 0.94$ .
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks, 2020. URL <https://arxiv.org/abs/2005.11401>.
- Ma, H., Chen, Y., Zhao, W., Yang, J., Ji, Y., Xu, X., Liu, X., Jing, H., Liu, S., and Yang, G. A mamba foundation model for time series forecasting, 2024. URL <https://arxiv.org/abs/2411.02941>.
- Oymak, G. sp500.csv. Hugging Face Datasets, 2025. URL [https://huggingface.co/datasets/guloyy/sp500\\_csv](https://huggingface.co/datasets/guloyy/sp500_csv). Version: main (commit 0189082); accessed 2026-01-19.
- Parnichkun, R. N., Tumma, N., Thomas, A. W., Moro, A., An, Q., Suzuki, T., Yamashita, A., Poli, M., and Massaroli, S. Quantifying memory utilization with effective state-size. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, 2025. URL <https://icml.cc/virtual/2025/poster/44915>. Poster and preprint.
- Póro, M., Wołczyk, M., Pascanu, R., von Oswald, J., and Sacramento, J. State soup: In-context skill learning, retrieval and mixing, 2024. URL <https://arxiv.org/abs/2406.08423>.
- Siems, J., Carstensen, T., Zela, A., Hutter, F., Pontil, M., and Grazi, R. Deltaproduct: Improving state-tracking in linear rnns via householder products, 2025. URL <https://arxiv.org/abs/2502.10297>.
- Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to transformer for large language models, 2023. URL <https://arxiv.org/abs/2307.08621>.
- Sun, Y., Li, X., Dalal, K., Xu, J., Vikram, A., Zhang, G., Dubois, Y., Chen, X., Wang, X., Koyejo, S., Hashimoto, T., and Guestrin, C. Learning to (learn at test time): RNNs with expressive hidden states, 2025. URL <https://arxiv.org/abs/2407.04620>.
- Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Tamkin, A., Durmus, E., Hume, T., Mosconi, F., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from CLIP. *Transformer Circuits*, 2024. URL <https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html>. Accessed 2026-01-15.

## A. Comprehensive Experimental Results

Table 3: All runs: Test MSE (lower is better). Positive  $\Delta\%$  means Mamba improves vs the baseline. Mamba training configuration is taken from the folder name (train tag).

Train tag	Eval mask	Method	MSE <sub>balanced</sub>	$\Delta\%$
seq32_burnin16	full	<b>Mamba</b> (seq32_burnin16)	<b>1.284e-04</b>	–
seq32_burnin16	full	Persist	1.847e-04	+30.5%
seq32_burnin16	full	EWMA ( $\lambda = 0.2$ )	1.598e-04	+19.6%
seq32_burnin16	full	EWMA ( $\lambda = 0.94$ )	1.603e-04	+19.9%
seq32_burnin16	full	HAR (1,5,22)	1.271e-04	-1.1%
seq32_burnin16	trainmask	<b>Mamba</b> (seq32_burnin16)	<b>1.192e-04</b>	–
seq32_burnin16	trainmask	Persist	1.857e-04	+35.8%
seq32_burnin16	trainmask	EWMA ( $\lambda = 0.2$ )	1.599e-04	+25.4%
seq32_burnin16	trainmask	EWMA ( $\lambda = 0.94$ )	1.408e-04	+15.3%
seq32_burnin16	trainmask	HAR (1,5,22)	1.245e-04	+4.2%
seq32_burnin31	full	<b>Mamba</b> (seq32_burnin31)	<b>2.252e-04</b>	–
seq32_burnin31	full	Persist	1.847e-04	-22.0%
seq32_burnin31	full	EWMA ( $\lambda = 0.2$ )	1.598e-04	-40.9%
seq32_burnin31	full	EWMA ( $\lambda = 0.94$ )	1.603e-04	-40.5%
seq32_burnin31	full	HAR (1,5,22)	1.271e-04	-77.2%
seq32_burnin31	trainmask	<b>Mamba</b> (seq32_burnin31)	<b>1.189e-04</b>	–
seq32_burnin31	trainmask	Persist	1.865e-04	+36.3%
seq32_burnin31	trainmask	EWMA ( $\lambda = 0.2$ )	1.606e-04	+26.0%
seq32_burnin31	trainmask	EWMA ( $\lambda = 0.94$ )	1.361e-04	+12.6%
seq32_burnin31	trainmask	HAR (1,5,22)	1.248e-04	+4.7%
seq32_full	full	<b>Mamba</b> (seq32_full)	<b>1.209e-04</b>	–
seq32_full	full	Persist	1.847e-04	+34.5%
seq32_full	full	EWMA ( $\lambda = 0.2$ )	1.598e-04	+24.4%
seq32_full	full	EWMA ( $\lambda = 0.94$ )	1.603e-04	+24.6%
seq32_full	full	HAR (1,5,22)	1.271e-04	+4.9%
seq32_full	trainmask	<b>Mamba</b> (seq32_full)	<b>1.209e-04</b>	–
seq32_full	trainmask	Persist	1.847e-04	+34.5%
seq32_full	trainmask	EWMA ( $\lambda = 0.2$ )	1.598e-04	+24.4%
seq32_full	trainmask	EWMA ( $\lambda = 0.94$ )	1.603e-04	+24.6%
seq32_full	trainmask	HAR (1,5,22)	1.271e-04	+4.9%
seq64_burnin32	full	<b>Mamba</b> (seq64_burnin32)	<b>1.539e-04</b>	–
seq64_burnin32	full	Persist	1.836e-04	+16.1%
seq64_burnin32	full	EWMA ( $\lambda = 0.2$ )	1.586e-04	+2.9%
seq64_burnin32	full	EWMA ( $\lambda = 0.94$ )	1.470e-04	-4.7%
seq64_burnin32	full	HAR (1,5,22)	1.253e-04	-22.9%
seq64_burnin32	trainmask	<b>Mamba</b> (seq64_burnin32)	<b>1.195e-04</b>	–
seq64_burnin32	trainmask	Persist	1.852e-04	+35.5%
seq64_burnin32	trainmask	EWMA ( $\lambda = 0.2$ )	1.595e-04	+25.1%
seq64_burnin32	trainmask	EWMA ( $\lambda = 0.94$ )	1.344e-04	+11.1%
seq64_burnin32	trainmask	HAR (1,5,22)	1.244e-04	+4.0%
seq64_burnin63	full	<b>Mamba</b> (seq64_burnin63)	<b>1.452e-04</b>	–
seq64_burnin63	full	Persist	1.836e-04	+20.9%
seq64_burnin63	full	EWMA ( $\lambda = 0.2$ )	1.586e-04	+8.4%
seq64_burnin63	full	EWMA ( $\lambda = 0.94$ )	1.470e-04	+1.2%
seq64_burnin63	full	HAR (1,5,22)	1.253e-04	-16.0%
seq64_burnin63	trainmask	<b>Mamba</b> (seq64_burnin63)	<b>1.195e-04</b>	–
seq64_burnin63	trainmask	Persist	1.868e-04	+36.0%
seq64_burnin63	trainmask	EWMA ( $\lambda = 0.2$ )	1.609e-04	+25.7%
seq64_burnin63	trainmask	EWMA ( $\lambda = 0.94$ )	1.345e-04	+11.1%
seq64_burnin63	trainmask	HAR (1,5,22)	1.255e-04	+4.8%
seq64_full	full	<b>Mamba</b> (seq64_full)	<b>1.193e-04</b>	–

Continued on next page

---

**Analyzing State-Space Dynamics in Mamba for Financial Time Series Forecasting**

---

Train tag	Eval mask	Method	MSE <sub>balanced</sub>	$\Delta\%$
seq64_full	full	Persist	1.836e-04	+35.0%
seq64_full	full	EWMA ( $\lambda = 0.2$ )	1.586e-04	+24.7%
seq64_full	full	EWMA ( $\lambda = 0.94$ )	1.470e-04	+18.8%
seq64_full	full	HAR (1,5,22)	1.253e-04	+4.7%
seq64_full	trainmask	<b>Mamba</b> (seq64_full)	<b>1.193e-04</b>	–
seq64_full	trainmask	Persist	1.836e-04	+35.0%
seq64_full	trainmask	EWMA ( $\lambda = 0.2$ )	1.586e-04	+24.7%
seq64_full	trainmask	EWMA ( $\lambda = 0.94$ )	1.470e-04	+18.8%
seq64_full	trainmask	HAR (1,5,22)	1.253e-04	+4.7%

---