Mamba4Cast: Efficient Zero-Shot Time Series Forecasting with State Space Models

Sathya Kamesh Bhethanabhotla *
Machine Learning Lab
University of Freiburg
Freiburg, Germany
bhethans@tf.uni-freiburg.de

Omar Swelam *
Machine Learning Lab
University of Freiburg
Freiburg, Germany
swelamo@tf.uni-freiburg.de

Julien Siems
Machine Learning Lab
University of Freiburg
Freiburg, Germany

David Salinas
Machine Learning Lab
University of Freiburg
Freiburg, Germany

Frank Hutter
Machine Learning Lab
University of Freiburg
Freiburg, Germany

Abstract

This paper introduces Mamba4Cast, a zero-shot foundation model for time series forecasting. Based on the Mamba architecture and inspired by Prior-data Fitted Networks (PFNs), Mamba4Cast generalizes robustly across diverse time series tasks without the need for dataset specific fine-tuning. Mamba4Cast's key innovation lies in its ability to achieve strong zero-shot performance on real-world datasets while having much lower inference times than time series foundation models based on the transformer architecture. Trained solely on synthetic data, the model generates forecasts for entire horizons in a single pass, outpacing traditional auto-regressive approaches. Our experiments show that Mamba4Cast performs competitively against other state-of-the-art foundation models in various data sets while scaling significantly better with the prediction length. The source code can be accessed at `https://github.com/automl/Mamba4Cast`.

1 Introduction

Time series forecasting is a critical task in numerous domains, from finance (He et al., 2023) to healthcare (Jung et al., 2021), and has been approached through various deep learning methods in recent years (Chen et al., 2023; Liu & Wang, 2024). Time-series data often exhibits complex temporal patterns, varying distributions with many confounding variables, and long-range dependencies, making it more challenging to model than other data paradigms. Although the recent Cambrian explosion in deep learning, especially foundation models (Touvron et al., 2023; Yu et al., 2022), can be attributed in part to the availability of large amounts of data for training, the same cannot be said about forecasting in some domains (Wang et al., 2023; Sivaroopan et al., 2024).

Forecasting models (Salinas et al., 2020; Zeng et al., 2023; Oreshkin et al., 2019) have traditionally employed non-zero-shot methods, which typically require customized training or fine-tuning for each specific task. While effective, this approach can be resource-intensive and time-consuming. Transformer-based time series foundation models (Ansari et al., 2024; Rasul et al., 2023; Dooley et al., 2023) have demonstrated significant potential to address these limitations. However, their application to long sequences during inference is constrained by their quadratic sample complexity.

In an effort to address both of these problems, we present Mamba4Cast, a time series foundation model based on two concepts: Prior-data Fitted Networks (PFNs) (Hollmann et al., 2023; Dooley et al., 2023) and the Mamba (Gu & Dao, 2024; Dao & Gu, 2024) architecture. Our contributions are twofold:

*Both authors contributed equally to this work.

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

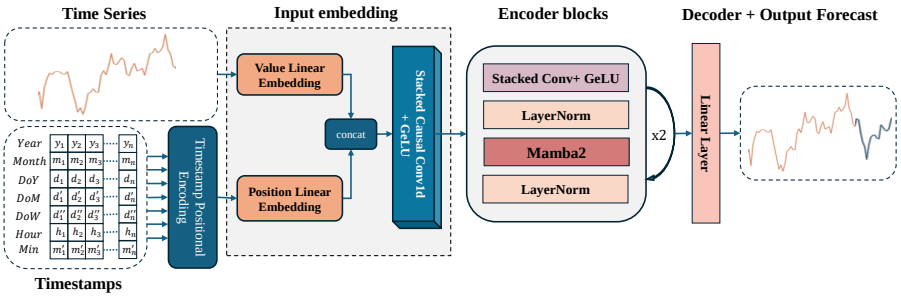

Figure 1: Schematic overview of the Mamba4Cast architecture.

- We introduce Mamba4Cast, a Mamba-based zero-shot forecasting model trained exclusively on synthetic data. It achieves competitive performance compared to other state-of-the-art zero-shot models, such as Chronos (Ansari et al., 2024), while leveraging Mamba's architecture for efficient scaling over longer context lengths.
- We demonstrate that Mamba4Cast provides accurate point predictions over the entire forecast horizon in a single forward pass, achieving inference speeds several times faster than autoregressive counterparts.

2 Related Work

Time series forecasting with Transformers In the last few years, transformer-based models have significantly improved the state of the art in time series forecasting. Works like the Informer (Zhou et al., 2021) and PatchTST (Nie et al., 2023) address the issue of long-term forecasting with transformers.

Zero-shot forecasting There have also been several advancements in zero-shot time series forecasting (Woo et al., 2024; Gruver et al., 2023). Gao et al. (2024) proposed UNITS, a unified multi-task model handling various predictive and generative tasks, and Oreshkin et al. (2021) proposed a meta-learning framework for zero-shot forecasting. These works highlight the growing trend towards more adaptable generalized time series models.

Training on Synthetic Data While pre-training has enhanced the generalization capabilities of many models, their inductive biases often remain constrained to the distributions of their training corpus, potentially necessitating fine-tuning for niche applications. ForecastPFN (Dooley et al., 2023), inspired by PFNs (Hollmann et al., 2023; Müller et al., 2022), addressed this limitation by training on synthetic data, enabling zero-shot generalization to real-world time series. More recently, Chronos (Ansari et al., 2024) demonstrated state-of-the-art results by training on both synthetic and real-world time series, introducing a transformer-based foundation model that follows the next-token prediction paradigm of large language models.

State Space Models Despite the success of transformer-based methods, they face scalability challenges due to their quadratic complexity. In contrast, state-space models, such as Mamba (Gu & Dao, 2024; Dao & Gu, 2024) or Linear Attention (Katharopoulos et al., 2020; Yang et al., 2024a,b), have emerged as more efficient architectures, adapting state space models / linear RNNs (Pöppel et al., 2024) for sequence modeling with linear scaling properties. This efficiency has proven crucial for modeling dense, long-sequence data in vision and time series forecasting (Behrouz et al., 2024; Patro & Agneeswaran, 2024). Subsequent works have further demonstrated Mamba's capacity in multivariate time-series forecasting; e.g., Wang et al. (2024) and Liang et al. (2024) proposed bi-directional Mamba architectures to capture inter- and intra-series dependencies, with the latter introducing a forget gate for enhancing selective performance on longer ranges. With recent studies showcasing Mamba's in-context learning capabilities (Grazzi et al., 2024; Park et al., 2024), Mamba4Cast attempts to utilize them towards a Mamba-based foundation model for zero-shot time series forecasting. We aim to address this unexplored avenue for univariate time series, by training over a diverse set of synthetic generation procedures that generalize to various real-life datasets.

3 Methodology

3.1 Background: State Space Models

Mamba4Cast builds upon the Mamba2 state-space model introduced by Dao & Gu (2024). Mamba2 is a linear Recurrent Neural Network described by the following recurrence:

$$h_t = A_t h_{t-1} + B_t x_t; \qquad y_t = C_t h_t$$

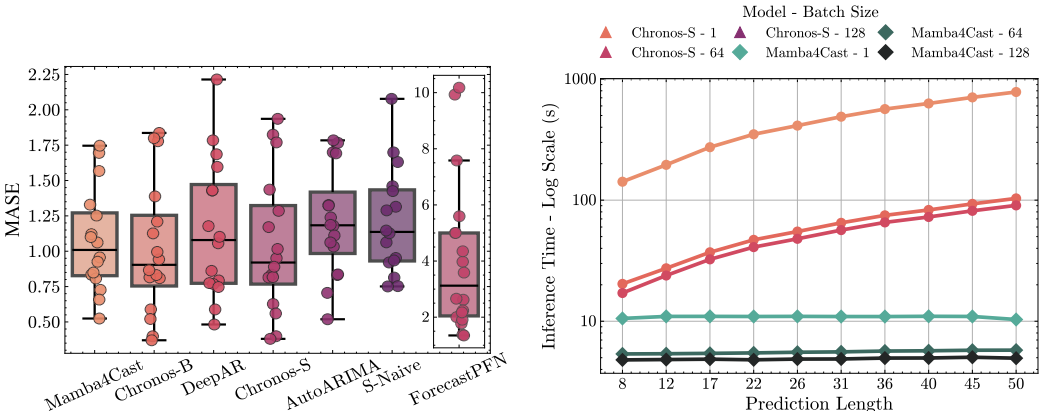

Figure 2: Performance and efficiency comparison of Mamba4Cast against baseline models. (*left*) Distribution of MASE across 16 evaluation datasets (excluding Covid Deaths) for Mamba4Cast and five baseline models (ForecastPFN was much worse and is on a separate scale). (*right*) Inference time of Mamba4Cast versus Chronos-Small on synthetically generated time series (2048 series, 512 context length) for increasing prediction lengths and varying batch sizes. The results demonstrate Mamba4Cast's superior efficiency, particularly for longer prediction horizons and larger batch sizes.

where h_t, x_t, and y_t represent the hidden state, input token embedding, and output at index t, respectively. In contrast to Mamba (Gu & Dao, 2024), which uses a fully parameterized diagonal state transition matrix A_t, Mamba2 employs a scalar multiple of the identity matrix allowing for more efficient computation. The recurrence can be computed in chunks of linear attention blocks that can be pieced together later, leveraging tensor cores through matrix multiplication. This approach differs from Mamba's evaluation through an associative scan, which is also performed in parallel across the sequence but cannot leverage GPUs as well.

3.2 Mamba4Cast Architecture

Our proposed architecture, illustrated in Figure 1, consists of four primary components:
(1) *Pre-processing*: we scale the input series using a Min-Max Scaler and extract time features for positional embeddings. (2) *Embedding*: we embed the scaled input values and their temporal information using convolutions with different dilations, ensuring a large receptive field for the representation used by future layers. For more details about data pre-processing and embedding, refer to Appendix C. (3) *Encoder*: comprises of Mamba2 blocks with LayerNorm to avoid noisy learning signals followed by another dilated convolution layer. (4) *Decoder*: the final component is a linear projection layer that transforms the embedded token representations into point forecasts.

We perform an ablation study, detailed in Appendix F, investigating the role of convolutions, the efficacy of synthetic data generation methods, and the performance of alternative inference strategies.

3.3 Synthetic Data Generation

The quality and diversity of the data generation process are crucial for Mamba4Cast's performance on real-world data, as it is trained exclusively on synthetic data. We employ two types of data-generating priors: ForecastPFN (FPFN) and Gaussian Process (GP) based. The *FPFN prior*, based on Dooley et al. (2023), decomposes a time series into trend, seasonality, and noise components reflecting real-life patterns. The *GP prior*, inspired by Ansari et al. (2024), complements the FPFN priors by accounting for patterns not captured therein. Each series is sampled from a GP with either a zero or a linear mean function and a composite kernel drawn from our *Kernel bank*. This allows for generating diverse and realistic synthetic time series that exhibit a wide range of temporal behaviors. Detailed descriptions of these data priors are provided in Appendix B.

4 Experiments

4.1 Training Details

The Mamba4Cast model is designed with approximately 27M parameters, positioning it between Chronos-Mini (20M) and Chronos-Small (46M) in size. It features 2 encoder layers following an input projection to an embedding dimension of 1024. We minimize the mean squared error over the prediction horizon using AdamW (Loshchilov & Hutter, 2019). The model is trained for 420K batches of size 64, using data sampled

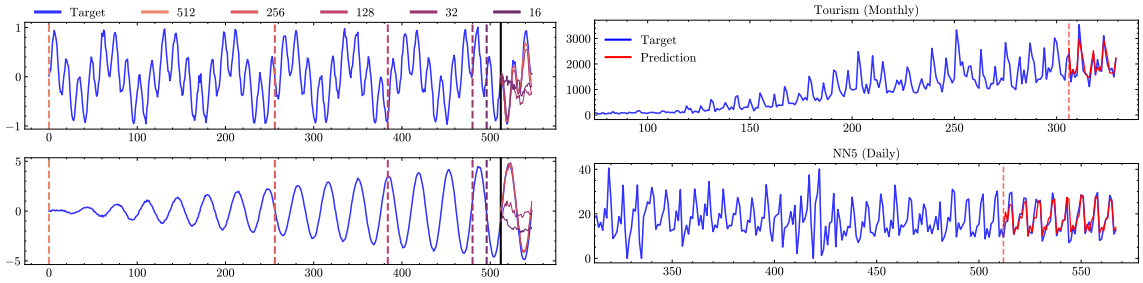

Figure 3: Qualitative analysis of Mamba4Cast's performance. (*left*) Demonstrates how prediction accuracy improves with increasing context length for multiplicative sine waves. (*right*) Illustrates the model's forecasting capabilities on two real-world time series datasets.

from the priors in Section 3.3, via a parallelized data loader that ensures the same sample is not seen twice. Further details about the experimental setup are provided in Appendix D.

4.2 Performance Comparison with Baseline Models

We evaluate on 17 publicly available time series datasets from a wide range of domains from the dataset repository of the GluonTS (Alexandrov et al., 2020) library with a 512 context length. A detailed list of the datasets used is included in Appendix E. Our evaluations involve comparisons with zero-shot baselines trained on synthetic data (Chronos and ForecastPFN), a deep learning baseline (DeepAR), and statistical methods (AutoARIMA and Seasonal Naive).

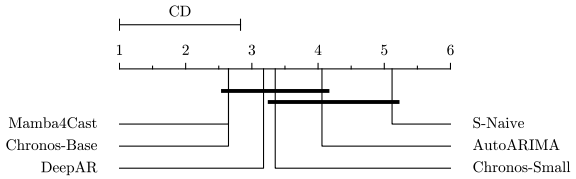

Figure 4: Critical difference diagram comparing mean MASE ranks of Mamba4Cast and baseline models across 17 time series datasets. ForecastPFN was much worse and is excluded for the sake of visibility.

For our evaluations, we use AutoGluon–TimeSeries (Shchur et al., 2023) to evaluate the baselines, with the exception of ForecastPFN, whose results are sourced from the Chronos paper (Ansari et al., 2024). To ensure fair comparison across datasets with varying scales, we use the MASE metric (Hyndman & Koehler, 2006), which is scale-invariant.

The results, as illustrated in Figures 2 and 4, demonstrate that Mamba4Cast achieves competitive performance with Chronos-Base(200M) and surpasses other baselines. Notably, this performance is achieved without fine-tuning on real-world datasets. Figure 4 shows a critical difference diagram, visualizing the mean model rankings based on MASE (Mean Absolute Scaled Error) over the datasets. In this diagram, models are arranged from best (*left*) to worst (*right*), with statistically insignificant performance differences indicated by connecting horizontal lines (at a significance level of $\alpha = 0.05$). Detailed information on the MASE metric and per-dataset results can be found in Appendix G.

4.3 Qualitative Analysis on Synthetic and Real Data

We conduct a qualitative inspection of Mamba4Cast to evaluate its ability to extrapolate over the forecasting horizon. Figure 3 illustrates Mamba4Cast's improvement with increasing context length and its ability to capture real-life patterns. We also visualize the model's forecasting capability on additional real-world data in Appendix G.

5 Conclusion and Future Work

Our experiments demonstrate Mamba's capability in creating reliable zero-shot time-series foundation model. Training solely on synthetic data, Mamba4Cast achieves near state-of-the-art results while also maintaining scalability and efficient inference. However, Mamba4Cast is limited to the univariate domain, which only forms a small portion of real time series problems, and is heavily reliant on the diversity of its priors. Nevertheless, we believe our work serves as a significant step towards developing highly performant and efficient multivariate zero-shot forecasting models, setting the stage for future advancements in this domain.

References

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., et al. Chronos: Learning the Language of Time Series. *arXiv preprint arXiv:2403.07815*, 2024.

Behrouz, A., Santacatterina, M., and Zabih, R. MambaMixer: Efficient Selective State Space Models with Dual Token and Channel Selection. *arXiv preprint arXiv:2403.19888*, 2024.

Chen, Z., Ma, M., Li, T., Wang, H., and Li, C. Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97:101819, 2023. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2023.101819.

Dao, T. and Gu, A. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *Forty-first International Conference on Machine Learning*, 2024.

Dooley, S., Khurana, G. S., Mohapatra, C., Naidu, S. V., and White, C. ForecastPFN: Synthetically-Trained Zero-Shot Forecasting. *Advances in Neural Information Processing Systems*, 37, 2023.

Gao, S., Koker, T., Queen, O., Hartvigsen, T., Tsiligkaridis, T., and Zitnik, M. UNITS: A Unified Multi-Task Time Series Model. *arXiv preprint arXiv:2403.00131*, 2024.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

Grazzi, R., Siems, J. N., Schrodi, S., Brox, T., and Hutter, F. Is Mamba Capable of In-Context Learning? In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large Language Models Are Zero-Shot Time Series Forecasters. *Advances in Neural Information Processing Systems*, 37, 2023.

Gu, A. and Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling*, 2024.

He, K., Yang, Q., Ji, L., Pan, J., and Zou, Y. Financial time series forecasting with the deep learning ensemble model. *Mathematics*, 11(4), 2023. ISSN 2227-7390. doi: 10.3390/math11041054.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*, 2023.

Hyndman, R. and Koehler, A. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22:679–688, 02 2006. doi: 10.1016/j.ijforecast.2006.03.001.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., and Wen, Q. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *The Twelfth International Conference on Learning Representations*, 2024.

Jung, S., Moon, J., Park, S., and Hwang, E. Self-attention-based deep learning network for regional influenza forecasting. *IEEE Journal of Biomedical and Health Informatics*, 26(2):922–933, 2021.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Liang, A., Jiang, X., Sun, Y., Shi, X., and Li, K. Bi-Mamba+: Bidirectional Mamba for Time Series Forecasting, 2024.

Liu, X. and Wang, W. Deep time series forecasting models: A comprehensive survey. *Mathematics*, 12(10): 1504, 2024.

Liu, Y., Qin, G., Huang, X., Wang, J., and Long, M. AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. *arXiv preprint arXiv:2402.02370*, 2024.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Müller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. Transformers Can Do Bayesian Inference. In *International Conference on Learning Representations*, 2022.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. Meta-learning framework with applications to zero-shot time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9242–9250, 2021.

Park, J., Park, J., Xiong, Z., Lee, N., Cho, J., Oymak, S., Lee, K., and Papailiopoulos, D. Can Mamba Learn How to Learn? A Comparative Study on In-Context Learning Tasks. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

Patro, B. N. and Agneeswaran, V. S. SiMBA: Simplified Mamba-Based Architecture for Vision and Multivariate Time series. *arXiv preprint arXiv:2403.15360*, 2024.

Pöppel, K., Beck, M., Spanring, M., Auer, A., Prudnikova, O., Kopp, M. K., Klambauer, G., Brandstetter, J., and Hochreiter, S. xlstm: Extended long short-term memory. In *First Workshop on Long-Context Foundation Models@ ICML 2024*, 2024.

Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Biloš, M., Ghonia, H., Hassen, N., Schneider, A., Garg, S., Drouin, A., Chapados, N., Nevmyvaka, Y., and Rish, I. Lag-Llama: Towards Foundation Models for Time Series Forecasting. In *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

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.

Shchur, O., Turkmen, A. C., Erickson, N., Shen, H., Shirkov, A., Hu, T., and Wang, B. AutoGluon–TimeSeries: AutoML for probabilistic time series forecasting. In *International Conference on Automated Machine Learning*, pp. 9–1. PMLR, 2023.

Sivaroopan, N., Bandara, D., Madarasingha, C., Jourjon, G., Jayasumana, A. P., and Thilakarathna, K. Netdiffus: Network traffic generation by diffusion models through time-series imaging. *Computer Networks*, 251:110616, 2024. ISSN 1389-1286.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, Y., Han, Y., Wang, H., and Zhang, X. Contrast everything: A hierarchical contrastive framework for medical time-series. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55694–55717. Curran Associates, Inc., 2023.

Wang, Z., Kong, F., Feng, S., Wang, M., Zhao, H., Wang, D., and Zhang, Y. Is Mamba Effective for Time Series Forecasting? *arXiv preprint arXiv:2403.11144*, 2024.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified Training of Universal Time Series Forecasting Transformers. In *Forty-first International Conference on Machine Learning*, 2024.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024a.

Yang, S., Wang, B., Zhang, Y., Shen, Y., and Kim, Y. Parallelizing linear transformers with the delta rule over sequence length. *arXiv preprint arXiv:2406.06484*, 2024b.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 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 *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

A Additional Related Work

In the wake of the recent success of Large Language Models (LLMs), a novel direction in time series analysis has emerged, focusing on adapting LLM-based architectures for forecasting. Studies such as Liu et al. (2024); Jin et al. (2024); Rasul et al. (2023) have demonstrated the effectiveness of re-purposing LLMs for time series tasks. These approaches involve techniques to align time series data with the text-based input expected by LLMs, such as using text prototypes or encoding time series as strings of numerical digits. Notably, Gruver et al. (2023) showed that LLMs can perform zero-shot time series forecasting at levels comparable to or exceeding purpose-built models. These developments suggest that LLMs are promising candidates for general-purpose time series analysis, which can offer advantages in flexibility and performance in various forecasting tasks.

B Synthetic Data Generation

B.1 ForecastPFN Prior

We adopted the prior generation process from Dooley et al. (2023) that decomposes the time series into three components as outlined in Section 3.3. The trend incorporates linear and exponential growth factors, while seasonal components capture periodic variations at multiple time scales (minutely, hourly, daily, weekly, and monthly), reflecting natural cycles in the data. Noise is modeled using a Weibull distribution to maintain a constant expected value. We introduced some modifications to the original procedure that are mentioned below.

Trend In our experiments, we found that training Mamba4Cast on long sequence time series with exponential trends results in suboptimal performance. Therefore we limited the non-linear growth behavior to be polynomial ones represented in the GP priors, while the FPFN prior only models linearly growing signals.

Seasonal Seasonal patterns are generated according to the granularity of the timestamps. For each granularity, we sample sine-wave signals, referred to as harmonics, with periodicities corresponding to that granularity: 60 for minutely, 24 for hourly, 7 for daily, and 12 for monthly data. For each time series, we sample harmonics from both its granularity and the immediate higher granularity. As an example, for minutely data, we sample seasonal signals with both minutely and hourly periodicities. In the original design, 10 or 6 harmonics were sampled for each granularity, but in our optimal setup, we used 8 and 5 harmonics, respectively. As the number of harmonics increases, their periodicity is scaled down by the harmonic index, allowing the model to capture finer fluctuations in the data.

B.2 GP priors

GP model We use GPyTorch (Gardner et al., 2018) for sampling our time series from composite kernels, with a sampled zero or linear mean, and a Gaussian likelihood. We add the noise using Cholesky jitter, with the jitter level being sampled among 0.1, 0.01, and 0.001, with probabilities of 0.1, 0.2, and 0.7 respectively. This design choice is to generalize Mamba4Cast for different noise levels in the real-life datasets.

Kernels The kernel bank comprises Linear, Polynomial, Matern, and Periodic kernels. To ensure complex time series patterns, we combine up to 6 kernels using sampled *binary operations* (addition or multiplication). The best training pipeline involved sampling a number of kernels from 1 to 6. In the first 360K training rounds, for each kernel, we sampled from Periodic, Matern, or Linear kernels with weights of 5, 1.5, and 1 respectively. This prior was inspired by the KernelSynth method outlined by Chronos (Ansari et al., 2024).
We also observed that training followed by a *fine-tuning* phase of 60K rounds with changed weights of Periodic, Matern, and Polynomial kernels to 5, 2 and 1 respectively, resulted in better generalization.

B.3 Signal level noises

In addition to the white noise signal incorporated in our priors, we introduce two types of multiplicative noise signals: spikes and step noise. **Spikes** are designed to introduce regular peaks at every interval, l. To simulate peaks that occur regularly but are irregularly spaced, we apply a masking window m, which masks up to 40% of the spikes within the window. Similarly, multiplicative **step** functions are applied in an alternating high-low-high-low pattern to enable Mamba4Cast to capture seasonal level shifts.

C Dataset Preprocessing

We adopt a preprocessing approach similar to e.g. ForecastPFN (Dooley et al., 2023). Time-steps are decomposed into minutely, hourly, day of week, day of month, day of year, monthly and yearly components, encoded using sinusoidal embeddings. These encodings, along with the series value, are linearly projected and concatenated to represent each time-point in a 112 embedding vector. The value of target tokens for model input across the prediction horizon is 0 for the prediction of point value or 1 for the cumulative mean prediction, fixed to 0 during inference.

With all input and output token embeddings stacked along the sequence dimension, we apply four causal convolution layers with kernel sizes of 5 and dilations of 1, 2, 4, and 8, concatenating their outputs for diverse temporal coverage. This facilitates capturing multi-scale temporal dependencies, enhancing our model's forecasting capabilities. The stack of causal convolution projects the tokens up into our desired embedding dimension of 1024 followed by an inception layer to combine the information across the temporal multi-scale for each token while maintaining the embedding size.

D Training Details

Architectural choices As demonstrated in Figure 1, Mamba4Cast is built on Mamba2 (Dao & Gu, 2024) with a state expansion factor (N) of 128 and a block expansion factor (E) of 2. The final layer of the encoder is defined similarly to the stacked convolution layer illustrated in Appendix C with the difference in the input channels being 1024 for the embedding size.

Training setup We train on sequence lengths uniformly sampled between 30 and 512 and minimize the mean squared error over a prediction length uniformly sampled between 10 and 60 per batch. 50% of the time we train to predict a contiguous chunk from the middle of the prediction length to improve predictability over the sequence by reducing reliance on previous states and encouraging emphasis on temporal information. The learning rate is cosine annealed (Loshchilov & Hutter, 2017) from 1e-5 to 1e-7 throughout the training.

The model is trained exclusively on synthetic data generated using two methods outlined in Section 3.3. The data composition is 70% sampled from GP priors and 30% sampled from FPFN priors, leveraging the GP kernels' flexibility in capturing diverse patterns. Training was conducted over 3 days on a single Nvidia RTX2080Ti GPU, for 360k training rounds consisting of 64 independently generated samples each. As stated in B.2, we continue training for another 60K rounds with a changed kernel composition and a learning rate of 1e-6.

E Benchmark Datasets

We use 17 datasets from Chronos zero-shot benchmark while removing datasets with very small context and prediction length, datasets that are very large, and datasets with sub-hourly frequencies. We will extend to support those datasets in future work. We used GluonTS as an interface for these datasets to have a comparable evaluation pipeline to Chronos. The context length (input sequence length) was restricted to be at most 512, while the prediction length varied according to the evaluated dataset as shown in Table 1.

F Ablation studies

We investigate the robustness of Mamba4Cast in different configurations, which fall into three main categories:

- **Architectural Changes:** We look into the effectiveness of a stacked causal convolutions layer (CNN) against a linear layer (Linear) in the input embedding and as the encoder-block's last layer. While adding the CNN layer as the final layer of the encoder block (*baseline*) provides superior performance with a significant overhead in model size, the key advantage stems from the CNN layer in the input embedding without overhead in model size.

  The model sizes of the three setups listed in the corresponding section of Table 2 are 27M, 17M, and 15M, in the same order as in the table.

- **Prior Mixing Ratios:** Given the importance of the distribution of synthetic data, we conducted experiments to explore the impact of each of the two approaches mentioned in Section 3.3. The

Table 1: Characteristics of Datasets Used for Zero-Shot Evaluation of Mamba4Cast and baselines.

Dataset	Frequency	Num. Test Series	Prediction Length
CIF 2016	1M	72	12
Car Parts	1M	2674	12
Covid Deaths	1D	266	30
ERCOT Load	1H	8	24
Exchange Rate	1B	8	30
FRED-MD	1M	107	12
Hospital	1M	767	12
M1 (Monthly)	1M	617	18
M1 (Quarterly)	3M	203	8
M3 (Monthly)	1M	1428	18
M3 (Quarterly)	3M	756	8
NN5 (Daily)	1D	111	56
NN5 (Weekly)	1W	111	8
Tourism (Monthly)	1M	366	24
Tourism (Quarterly)	3M	427	8
Traffic	1H	862	24
Weather	1D	3010	30

ablation indicates the effectiveness of the GP prior over the FPFN prior, leading to our choice of a GP favoured mixture of data for training.

- **Inference Modes:** Mamba4Cast was designed with efficient zero-shot forecasting in mind following the one-pass multipoint setup, in which the input and target tokens are concatenated together in their respective order. Mamba4Cast also supports autoregressive forecasting, but its performance declines significantly in this setup. A likely reason is that feeding predicted values back into the model causes overconfidence and error propagation. In contrast, the multipoint setup treats all target values as unknown, avoiding this issue.

  We further test the impact of ensembling by averaging the forecasts generated at 5 different levels of dropout, from 0 to 0.5, of the input sequence. However, given the superior performance over longer and more inclusive contexts, demonstrated in Figure 3, it follows that including a less accurate forecast can degrade performance in case Mamba4Cast is certain about its forecast as shown in Table 2.

The ablation studies were conducted on the first 360K training rounds mentioned in Appendix D, as the subsequent 60K were later applied to our chosen setup for the baseline comparisons cited in Section 4.2.

G Evaluations on real datasets

G.1 Evaluation metric

As part of our evaluation, we tested the performance of our model on real-world time series datasets alongside the synthetic data. The primary metric used was the seasonal Mean Absolute Scaled Error (MASE), which scales the forecast error by the mean absolute error of a seasonal naïve forecast on the training data. The evaluation of Mamba4Cast on real-world datasets demonstrates the model's capability to generalize and perform well in diverse, real-world forecasting scenarios. Detailed evaluations per dataset can be found in Table 3. We witnessed inconsistencies between the evaluations performed by AutoGluon in our setups and the ones reported in Chronos paper on datasets with daily frequency, specifically on "Covid Deaths." This resulted in the large gap witnessed on ForecastPFN's results reported here, since the model's MASE evaluations are sourced from the Chronos paper. The results reported for Mamba4Cast per dataset are evaluated with the best model trained according to the procedures in Appendix D.

Table 2: Ablation study on architectural changes, prior mixing ratios and the inference modes. The value reported is the geometric mean of MASE across all 17 datasets for each setup.

Ablation Setup	Mean MASE
Architectural Modifications	
CNN_{in_emb} / CNN_{enc_out} (*Baseline*)	1.153
CNN_{in_emb} / Linear_{enc_out}	1.205
Linear_{in_emb} / Linear_{enc_out}	1.556
Priors Mixing Ratios	
70% GP Prior / 30% FPFN Prior (*Baseline*)	1.153
100% GP Prior / 0% FPFN Prior	1.167
0% GP Prior / 100% FPFN Prior	1.579
Inference Modes	
Multipoint Forecasting (*Baseline*)	1.153
Autoregressive Forecasting	2.044
Ensemble Forecasting	1.558

Table 3: MASE evaluations on all of the 17 datasets with the lower value the better. The best results per dataset are in bold and the second best results are underlined.

	Zero-shot				Task-specific		Statistical Baseline
Dataset	Mamba4Cast	Chronos-B	Chronos-S	ForecastPFN	DeepAR	AutoARIMA	S-Naive
Car Parts	1.061	0.832	<u>0.817</u>	2.657	**0.747**	1.180	1.127
CIF 2016	**0.925**	<u>0.995</u>	1.016	3.558	1.597	1.062	1.289
Covid Deaths	**5.926**	7.461	7.376	91.515	8.917	<u>6.059</u>	8.977
ERCOT Load	0.657	**0.521**	<u>0.560</u>	3.975	1.429	1.112	0.751
Exchange Rate	<u>1.329</u>	1.388	1.436	7.583	2.214	**1.187**	1.460
FRED-MD	0.524	**0.399**	<u>0.399</u>	2.621	0.588	0.519	0.935
Hospital	<u>0.806</u>	0.815	0.814	1.775	**0.775**	0.836	0.921
M1 (Monthly)	**1.100**	1.126	1.171	2.172	<u>1.102</u>	1.239	1.314
M1 (Quarterly)	**1.695**	1.778	1.824	9.931	1.784	<u>1.766</u>	2.078
M3 (Monthly)	**0.849**	<u>0.866</u>	0.890	2.240	1.056	1.033	1.146
M3 (Quarterly)	1.251	<u>1.210</u>	1.285	10.176	**1.178**	1.323	1.425
NN5 (Daily)	0.833	<u>0.809</u>	0.834	1.375	**0.793**	0.832	0.952
NN5 (Weekly)	0.956	<u>0.942</u>	0.950	1.349	**0.861**	1.700	1.063
Tourism (Monthly)	<u>1.567</u>	1.836	1.936	4.348	**1.430**	1.692	1.631
Tourism (Quarterly)	1.746	1.799	1.770	5.595	**1.686**	1.784	<u>1.699</u>
Traffic	1.120	**0.370**	<u>0.380</u>	1.909	0.482	1.327	0.753
Weather	0.726	**0.589**	<u>0.627</u>	2.003	0.769	0.705	0.813

G.2 Qualitative analysis

An impartial evaluation in time series forecasting applications favors a qualitative evaluation over the datasets in question, to guarantee adequate behavior for point forecasting. For this sake, Figure 5 demonstrates Mamba4Cast's ability to capture diverse patterns exemplified in the real-life datasets.

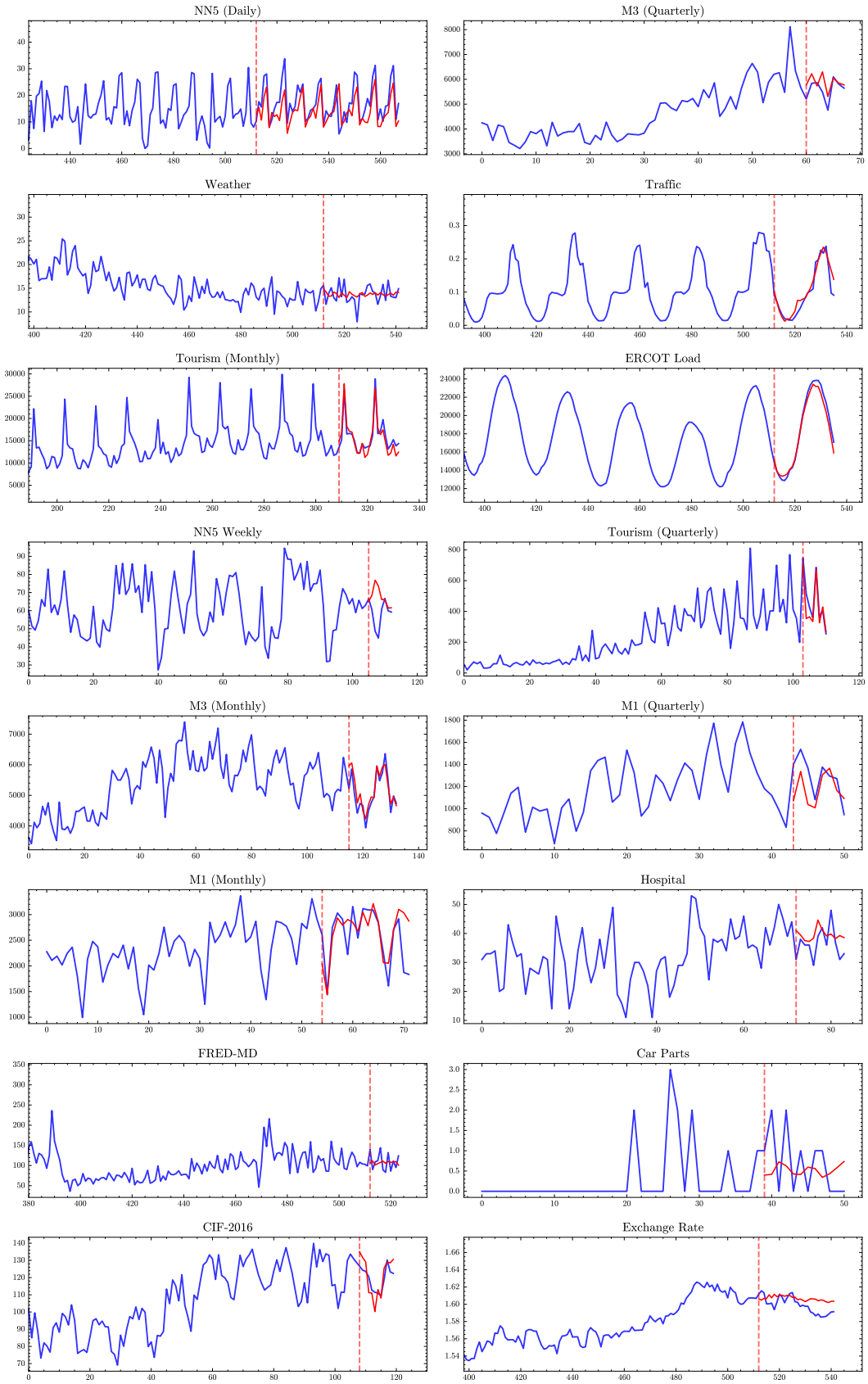

Figure 5: Qualitative analysis of real-world datasets evaluated by Mamba4Cast. Blue denotes the ground-truth, red the prediction.