Re-imagining Time Series Foundation Models: Metadata and State-Space Model Perspectives

Pengrui Quan
UCLA
prquan@g.ucla.edu

Ozan Baris Mulayim
CMU
omulayim@andrew.cmu.edu

Liying Han
UCLA
liying98@ucla.edu

Dezhi Hong*
Amazon
hondezhi@amazon.com

Mario Bergés †
CMU
marioberges@cmu.edu

Mani Srivastava †
UCLA
mbs@ucla.edu

Abstract

The success of foundation models in natural language processing has sparked a growing interest in developing analogous models for time series (TS) analysis. These time series foundation models (TSFM), pre-trained on vast amounts of TS data, demonstrate capabilities of zero-shot and few-shot inference on unseen datasets. However, the intrinsic heterogeneity of TS data presents unique challenges: accurate inference often necessitates a deep understanding of the underlying data-generating process and the sensing apparatus, which cannot be readily inferred from the raw data alone. Furthermore, recent advances in state-space models raise the question of whether they may offer advantages over transformer-based architectures for TS analysis.

This paper investigates these questions in two key areas: (a) a fair comparison of methods for integrating metadata into TSFMs and (b) the comparative effectiveness of state-space models (SSM) versus transformer models for TS forecasting. Our results, based on experiments across 11 datasets, suggest advantages for SSM building blocks as well as for incorporating the notion of real-world timestamps. More specifically, on our curated in-domain and out-of-domain datasets, an SSM approach incorporating timestamps outperforms three existing TSFMs on forecasting tasks while using 6,000$\times$ fewer trainable parameters and 10$\times$ less training data. The paper aims to highlight the potential for SSM building blocks and general directions for future TSFM research.

1 Introduction

Recent advancements in natural language processing have inspired the development of Time Series Foundation Models (TSFMs) capable of zero-shot and few-shot inference on unseen data [11, 12, 26]. These models aim to generalize forecasting across diverse time series (TS) tasks, similar to how language models handle various tasks without fine-tuning. However, unlike natural language (NL), which has clear syntactic and semantic structures, TS data is often less structured, lacks explicit context, and relies heavily on domain-specific attributes. This makes directly applying methods developed for NL to TS a non-trivial task. Moreover, existing TSFM approaches primarily treat TS forecasting as a value prediction problem [2, 4, 17], neglecting the contextual elements that could

*Work unrelated to Amazon.

†Authors hold concurrent appointments as Amazon Scholars, and as Professors at their respective universities, but work in this paper is not associated with Amazon.

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

enrich model understanding. Therefore, this paper explores whether integrating language-based metadata and utilizing timestamps as additional input channels can improve TSFM performance.

Most TSFM approaches rely on transformer (TM) architectures for handling sequential data. Recent advances in structured state-space models (SSMs) [27, 7] provide an alternative by modeling latent states and capturing long-range dependencies. However, SSMs have mainly been tested on their training datasets, limiting comparisons with TM architectures in TSFM. To address this, we evaluated the two architectures on curated in-domain and out-of-domain scenarios.

Our contributions are summarized as follows: (1) We compare the performance of SSM-based against TM-based TSFMs, highlighting the potential of using SSM as a building block. (2) We show that by incorporating timestamps as side channels, SSM models outperform existing TSFMs with 6,000$\times$ fewer trainable parameters and requiring 10$\times$ less training data. Though these results were obtained from our curated datasets, we anticipate generalization to broader applications. (3) We demonstrate that simple approaches for combining language-based metadata into TSFMs are insufficient, and discuss potential further research directions for the broader TSFM community.

2 Related works

Existing TSFMs. TSFMs [5, 4, 2, 20, 24] are a rapidly evolving area creating foundation models capable of generalizing to diverse, unseen TS datasets. Unlike language models, which are trained on self-contained textual data, TSFMs face unique challenges due to the need to incorporate additional contextual information. For instance, many TS analysis tasks often require incorporating additional covariates and the notion of time to provide context [8, 22, 28, 25].

In contrast, the majority of TSFMs reduce the TS forecasting task to value prediction similar to NL data [2, 4, 17], disregarding external factors and the dependency on integrating a time dimension. However, many real signals like temperature, electricity demand, or stock prices exhibit different statistical properties based on the time of day, week, or season [16, 8]. This `non-stationarity` necessitates models that dynamically adapt to temporal variations using absolute timestamps. Although some works [20, 24] incorporate timestamps as covariates, there are no direct comparisons on the usage of timestamps in TSFMs, making it difficult to draw deterministic conclusions.

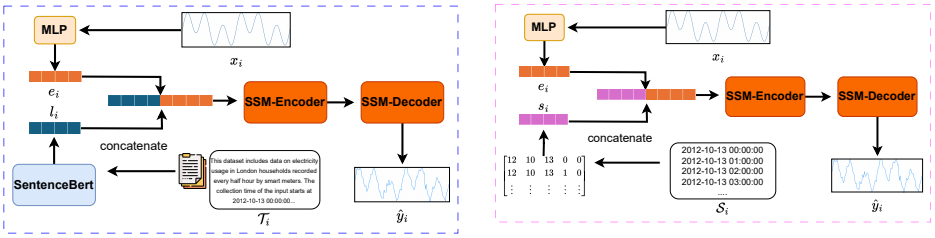
Large Language Models for TS. Recent works have explored adapting Large Language Models (LLMs) for TS forecasting by reprogramming pre-trained models [11, 9]. While these efforts incorporate NL metadata and TS data, the models' capability of general TS analysis without the need for retraining on new datasets has not been shown. On the other hand, [6] demonstrated that LLMs could perform zero-shot TS forecasting using Byte Pair Encoding (BPE) tokenization with inserted spaces between digits. However, LLMs are not inherently designed for TS tasks, so their performance in zero-shot inference remains inconsistent, and they fail to understand TS effectively [18, 15]. The above limitations suggest a combination of the flexibility of LLMs and the specificity of TSFMs as a way forward.

3 Methodology

In this section, we describe our proposed model architectures. Let N be the dataset size, $i \in \{1, 2, ..., N\}$ be the index of a TS. Let $x_i \in \mathbb{R}^c$ be the C historical observations, let $\mathcal{S}_i$ be the list of timestamps with length C, let $\hat{y}_i \in \mathbb{R}^H$ be the forecast for the future H observations, and $y_i \in \mathbb{R}^H$ be the ground truth for the future H observations. The objective of the training is to minimize the Mean Square Error (MSE) loss: $L = \frac{1}{N}\sum_{i=1}^{N} ||y_i - \hat{y}_i||_2^2$.

Structured State-space model (SSM). We follow the design of SSM architectures [27] to capture long-term temporal dependencies. A time series x_i is first mapped to a high dimensional embedding e_i with an MLP layer. e_i is further input to the SSM encoder-decoder model with companion matrix parameterization. The design of the companion matrices is to capture autoregressive processes, effectively expressing time series models such as ARIMA and exponential smoothing.

Transformer-based model (TM). We employ the decoder-only GPT-2 architecture [19] after the time series x_i is mapped to a high dimensional embedding e_i. The model is trained to adapt to variable context and horizon lengths, with the training objective of predicting the value of the next timestamps. Following the convention in language modeling [23], the ground truth value at each



(a) LA-SSM (b) TS-SSM

Figure 1: Model architectures.

Dataset	Domain	SSM	TS-SSM	LA-SSM	TM	TS-TM	LA-TM	Moment*	Chronos*	Moirai*
# param (M)	-	0.052	0.058	0.091	0.050	0.051	0.091	385	710	311
elecdemand	energy	0.446	0.443	1.619	0.688	0.584	2.044	0.748	0.55	**0.317**
subseasonal	weather	18.407	**16.266**	25.54	19.055	20.111	19.008	17.432	17.598	16.813
pems04	traffic	47.398	**39.516**	117.517	64.285	78.387	53.443	39.82	43.814	40.0
covid	healthcare	469.921	394.01	1296.571	750.06	866.71	693.272	487.717	**147.721**	9.57e8
rlp	traffic	3.843	8.664	13.596	4.004	3.879	3.847	**3.491**	3.786	98.602
loop_seattle	traffic	2.609	2.542	2.518	2.15	2.22	3.133	**1.882**	2.258	1.998
ECL	energy	108.338	104.863	360.632	191.275	153.534	219.78	194.311	**56.154**	59.874
smart*	energy	0.627	**0.607**	1.582	0.668	0.67	0.722	0.639	0.699	27.395
ID ranking	-	4.13	**3.0**	8.13	5.63	5.88	6.75	3.5	3.38	4.63
Ecobee	temperature	1.535	1.357	1.999	1.743	1.534	2.079	1.729	1.61	**1.257**
restaurant	sale	14.368	13.286	17.055	14.102	14.161	13.844	**13.25**	15.195	13.437
air	nature	22.989	**20.669**	73.081	21.988	22.306	24.631	23.067	25.056	21.81
OOD ranking	-	5.33	**1.67**	8.67	5.0	4.33	6.67	4.33	7.0	2.0
AVG ranking	-	4.45	**2.64**	8.27	5.45	5.45	7.72	3.72	4.36	3.91

*Moirai has seen a majority of the datasets during training. Moment and Chronos have seen the ECL and covid datasets during training.

Table 1: Performance of models across domains. The best results are **bold**, and the second best are underscored. Our models were not trained in temperature, sale, and nature domains.

decoding step t is used as the input to be conditioned for step $t+1$ during training time, where $t \in \{1, 2, ..., H-1\}$.

Language-augmented SSM (LA-SSM). Following [10], we generate a text description for a TS x_i containing dataset-related metadata, such as context, sampling rate, and data collection time. To inject language variations into metadata, we use GPT-4o-mini[1] to generate 50 versions of metadata for each dataset. Finally, we combine the metadata with sample-related data (minimum, maximum, median, and lag) to produce the description $\mathcal{T}_i$. We then use SentenceBert [21] to generate the embeddings of the above text description $\mathcal{T}_i$ and calculate the average embedding l_i, which is input to a learnable MLP. We concatenate the embeddings produced by an MLP layer e_i to l_i to generate the embedding $[l_i, e_i]$ for the SSM blocks, shown in Fig. 1(a).

Timestamp-encoded SSM (TS-SSM). Similarly, in Fig. 1(b), we encode timestamps $\mathcal{S}_i$ for the i-th TS into a five-dimensional vector s_i containing year, month, date, hour, and minute values. We concatenate the embeddings e_i produced by the MLP to s_i to generate the embedding $[s_i, e_i]$ for the SSM blocks.

LA-TM and TS-TM. Using the same configuration in Fig. 1, we also design the Language-augmented- and Timestamp-encoded-TMs. The only differences are that we replace the encoder-decoder-SSM blocks with the GPT-2 decoder blocks.

4 Experiments

For model training, we use 0.3% of the training datasets from Moirai[24], resulting in a training dataset with 98M observations (see Table 3 in the Appendix). We test models on new datasets for zero-shot forecasting (we exclude the training datasets). We also test existing TSFMs, including Chronos[2], Moirai[24], and Moment[5] (we used largest versions). Table 2 summarizes the evaluation datasets and their visibility to TSFMs' training. While Moirai, Moment, and Chronos have different levels of familiarity with the datasets in Table 2, our models have not been exposed to them. Training and inference use standardized normalization [2] for zero mean and unit variance. Following [27], we use AdamW [13] with a 0.001 learning rate, cosine scheduler with warmup, batch size 256, and train for 4 epochs on 4× H100-96G GPUs with FP32. We use $C = 192$ and $H = 48$ for both phases. [3]

[3]Code and model weights are available at: `https://github.com/nesl/Reimagine_TSFM`.

4.1 Zero-shot evaluation on in-domain and out-of-domain datasets

In-domain (ID) evaluation. We select new datasets from the domains used during training (refer to Table 2) and subsample them for ID evaluation. The first section of Table 1 shows the results of ID evaluation and the models' rankings across all ID datasets (the lower, the better). TS-SSM achieves the best performance among all models, even though the model has fewer trainable parameters and is trained on less data than existing TSFMs. Besides, by incorporating timestamps as a side channel, SSM has a larger gain compared to TM. Additionally, SSM models can consistently achieve superior performance compared to TM, showing the potential of using SSM in TSFM development. Finally, both LA-SSM and LA-TM perform below baseline, indicating limited effectiveness in incorporating language-based metadata.

Out-of-domain (OOD) evaluation. We evaluate model generalizability on datasets from unseen domains, as shown in the second part of Table 1. TS-SSM achieves the best ranking among all models, even outperforming Moirai, which was trained on these datasets. Incorporating timestamps improves both SSM and TM performance, while language metadata reduces model effectiveness.

Overall, despite not being trained on the datasets or on data from the same domain, SSM models can achieve comparable performance to existing TSFMs, with 6,000 to 12,000$\times$ fewer parameters and 10 to 850$\times$ less training data (details in Table. 4 in Appendix). More importantly, by incorporating timestamps as side channels, TS-SSM can achieve either the best or second-best results for all datasets except rlp, loop_seattle, and ECL.

Testing Dataset	Domain	Frequency	# Time Series	Moment	Chronos	Moirai	Ours
elecdemand	energy	6H	10	No	No	Yes	No
subseasonal	weather	D	200	No	No	Yes	No
pems04	traffic	5T	200	No	No	Yes	No
covid	healthcare	D	200	Yes	Yes	Yes	No
rlp (Residential Load Power)	traffic	30T	200	No	No	Yes	No
Loop Seattle	traffic	5T	200	No	No	Yes	No
ECL	energy	H	200	Yes	Yes	Yes	No
Smart*[3]	energy	H	200	No	No	No[1]	No
ecobee [14]	temperature	H	200	No	No	No	No
restaurant	sale	D	200	No	No	Yes	No
air (China air quality)	nature	H	200	No	No	Yes	No

[1]Moirai is trained on the *Home dataset* in Smart*, while we use the *Apartment dataset* in Smart* for testing.

Table 2: Testing datasets and their visibility during models' training. Yes/No indicates whether the dataset has been incorporated into the models' training dataset. The first part of the table is used for ID evaluation, and the second part is used for OOD evaluation.

5 Conclusion

In this work, we demonstrated the effectiveness of SSM building blocks in TSFMs, showing their superiority in forecasting over TM models. With timestamps as a side channel, TS-SSM outperformed existing transformer-based TSFMs despite having orders of magnitude fewer parameters.

Limitations & Future work. However, our study also highlights several limitations that open avenues for future research. First, our initial attempt to integrate language-based metadata into the model did not yield performance gains. The current incapability of our models to incorporate language-based metadata could be attributed to the limited model size and approach to properly learning the language modality.

Additionally, our experiments are limited in scope and dataset size compared to broader TSFM studies, and our baselines were not exhaustive. Future research should expand dataset diversity and include more TSFM baselines for a comprehensive comparison.

6 Acknowledgment

This research was sponsored in part by the AFOSR award #FA95502210193, the DEVCOM ARL award #W911NF1720196, the NSF award #CNS-23091241, the NIH award #1P41EB028242, and the Pennsylvania Infrastructure Technology Alliance.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[3] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, Jeannie Albrecht, et al. Smart*: An open data set and tools for enabling research in sustainable homes. *SustKDD, August*, 111(112):108, 2012.

[4] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, February 2024. URL `http://arxiv.org/abs/2310.10688`. arXiv:2310.10688 [cs].

[5] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A Family of Open Time-series Foundation Models, February 2024. URL `http://arxiv.org/abs/2402.03885`. arXiv:2402.03885 [cs].

[6] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[7] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, December 2023. URL `http://arxiv.org/abs/2312.00752`. arXiv:2312.00752 [cs].

[8] RJ Hyndman. *Forecasting: principles and practice*. OTexts, 2018.

[9] Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23343–23351, 2024.

[10] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[11] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models, January 2024. URL `http://arxiv.org/abs/2310.01728`. arXiv:2310.01728 [cs].

[12] Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. UniTime: A Language-Empowered Unified Model for Cross-Domain Time Series Forecasting, February 2024. URL `http://arxiv.org/abs/2310.09751`. arXiv:2310.09751 [cs].

[13] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017.

[14] Na Luo and Tianzhen Hong. Ecobee donate your data 1,000 homes in 2017. 2022. doi: 10.25584/ecobee/1854924.

[15] Mike A Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*, 2024.

[16] Ozan Baris Mulayim, Pengrui Quan, Liying Han, Xiaomin Ouyang, Dezhi Hong, Mario Bergés, and Mani Srivastava. Are time series foundation models ready to revolutionize predictive building analytics? In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 169–173, 2024.

[17] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[18] Pengrui Quan, Xiaomin Ouyang, Jeya Vikranth Jeyakumar, Ziqi Wang, Yang Xing, and Mani Srivastava. Sensorbench: Benchmarking llms in coding-based sensor processing. *arXiv preprint arXiv:2410.10741*, 2024.

[19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[20] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting, February 2024. URL `http://arxiv.org/abs/2310.08278`. arXiv:2310.08278 [cs].

[21] N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[22] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

[23] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[24] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[25] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[26] Jiexia Ye, Weiqi Zhang, Ke Yi, Yongzi Yu, Ziyue Li, Jia Li, and Fugee Tsung. A Survey of Time Series Foundation Models: Generalizing Time Series Representation with Large Language Model, May 2024. URL `http://arxiv.org/abs/2405.02358`. arXiv:2405.02358 [cs].

[27] Michael Zhang, Khaled K Saab, Michael Poli, Tri Dao, Karan Goel, and Christopher Ré. Effectively modeling time series with simple discrete state spaces. *arXiv preprint arXiv:2303.09489*, 2023.

[28] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

A Appendix

A.1 Training dataset characteristics

Table. 3 summarizes the training datasets used in this paper and their corresponding domains.

A.2 Model characteristics

We summarize the size of models, size of training datasets, tokenization, objective function, and backbone architecture of models in Table. 4.

Training Dataset	Domain	Frequency	# Time Series	# Observations
london_smart_meters	Energy	30T	713	9,543,348
bdg-2_bear	Energy	H	91	1,482,312
bdg-2_fox	Energy	H	135	2,324,568
era5_1989	Climate	H	8192	71,565,312
era5_1990	Climate	H	8192	71,565,312
era5_1991	Climate	H	8192	71,565,312
cmip6_2005	Climate	6H	8192	59,801,600
cmip6_2010	Climate	6H	8192	59,801,600
weather	Climate	D	3010	12,717,250
oikolab_weather	Climate	H	8	800,456
uber_tlc_daily	Transport	D	262	47,087
SZ_TAXI	Transport	15T	156	464,256
PEMS03	Transport	5T	358	9,382,464
hospital	Healthcare	M	767	55,224
kaggle_web_traffic	Web	W	145063	16,537,182
tourism_yearly	Eco	A	419	14,665
tourism_monthly	Eco	M	366	59,658

Table 3: Trainign datasets and their characteristics

Dataset	SSM	TS-SSM	LA-SSM	TM	TS-TM	LA-TM	Moment-Large*	Chronos-Large*	Moirai-Large*
# param (M)	0.052	0.058	0.091	0.050	0.051	0.091	385	710	311
# observations (B)	0.098						1.13	84	27.65
Tokenization	Direct mapping TS to embeddings						Fixed-length patches	Multi-scale patches	Scaling & quantization
Loss	MSE						MSE	Cross-Entropy	Negative log likelihood
Backbone	SSM				GPT-2		T5 encoder	Transformer encoder	T5 encoder-decoder

Table 4: Model characteristics. The # observations denote the total number of variates in the training dataset.