LLM-Integrated Bayesian State Space Models for Multimodal Time-Series Forecasting

Sungjun Cho, Changho Shin, Suenggwan Jo,
Xinya Yan, Shourjo Aditya Chaudhuri, Frederic Sala
Department of Computer Sciences
University of Wisconsin-Madison
{cho266, cshin23, sjo32, xyan89, sachaudhuri, fsala}@wisc.edu

Abstract

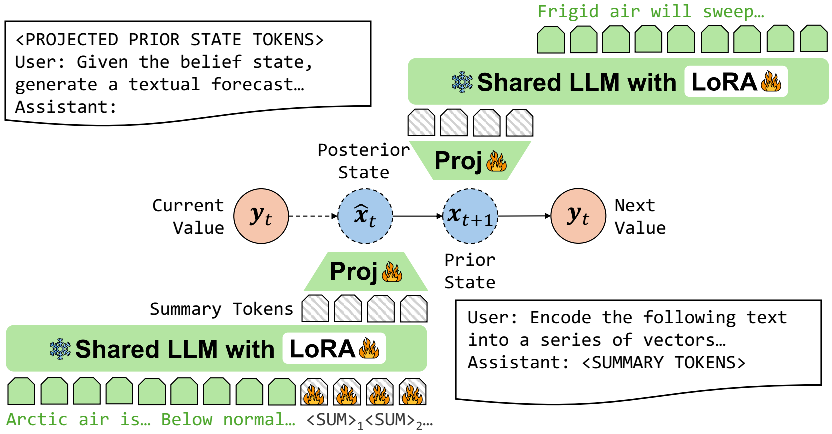
Forecasting in the real world often requires combining structured time-series data with unstructured textual information, yet most existing methods treat these modalities in isolation. We address this gap with the **LLM-integrated Bayesian State Space Model (LBS)**, a probabilistic framework for multimodal temporal forecasting. At a high level, LBS consists of two components: (1) a state space model (SSM) backbone captures the temporal dynamics of latent states from which both numerical and textual observations are generated, and (2) a pretrained large language model (LLM) is adapted to encode textual inputs for posterior state estimation and decode textual forecasts consistent with the latent trajectory. This design enables flexible lookback and forecast windows, principled uncertainty quantification, and improved temporal generalization thanks to the well-suited inductive bias of SSMs toward modeling dynamical systems. Experiments on the TimeText Corpus benchmark demonstrate that LBS improves the previous state-of-the-art by 13.20% while providing human-readable textual summaries. **Our work is the first to unify LLMs and SSMs for joint numerical and textual prediction, offering a novel foundation for multimodal temporal reasoning**.

1 Introduction

Time-series forecasting is a core machine learning task traditionally centered on predicting future numerical values from past data [27]. However, in many real-world domains, contextual information expressed in natural language—such as clinical notes, financial reports, or weather descriptions—plays a critical role in forecasting. This complementary modality can offer valuable signals that cannot be fully extracted from numeric data alone [25, 19]. Similarly, generating textual forecasts alongside numerical predictions can be particularly useful in high-stakes decision-making scenarios. These opportunities motivate the development of models that not only forecast from multimodal inputs, but also communicate their predictions through natural language, augmenting quantitative accuracy with qualitative explanations.

Probabilistic state space models (SSMs) offer compelling advantages for time-series forecasting: their inductive biases fit will for modeling temporal dynamics, quantify uncertainty in a principled manner, and support variable-length input/prediction horizons. While integrating probabilistic SSMs with pretrained large language models (LLMs) appear to be a natural direction to enable joint numeric and textual modalities, the direction presents two yet unexplored fundamental challenges: **(C1) Text-conditioned posterior state estimation:** How can we update the latent state of the SSM using a pretrained LLM and textual observations? **(C2) Latent state-conditioned text generation:** How can we adapt the LLM to generate accurate, temporally grounded textual forecasts conditioned on latent state trajectories?

NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT^{2}S).



```
<PROJECTED PRIOR STATE TOKENS>
User: Given the belief state,
generate a textual forecast…
Assistant:
```

```
User: Encode the following text
into a series of vectors…
Assistant: <SUMMARY TOKENS>
```

Figure 1: An illustration of LBS under a single-step forecasting scenario. **Bottom:** To enable Bayesian updates with text, the LLM is tuned to summarize the context into a set of summarization tokens, which are then used together with the target value to obtain the posterior state distribution. **Top:** Conditioned on the state forecast, the shared LLM is trained to generate its corresponding text. When using instruction-tuned LLMs, both steps are accompanied by prompt templates in order to preserve its capabilities.

In response, we propose the **LLM-integrated Bayesian State Space Model (LBS)**, a novel architecture that unifies a probabilistic SSM with a pretrained LLM for joint numeric and textual forecasting (see Figure 1). For **(C1)**, we adapt the LLM to summarize and compress textual inputs into a sequence of summary tokens, which are projected into the low-dimensional latent state space for deep Bayesian filtering. For **(C2)**, we leverage the LLM's in-context generation capabilities by conditioning it on latent state trajectories—treated as non-textual context akin to images or videos—enabling temporally coherent, state-grounded textual forecasts. Evaluated on the TIMETEXT CORPUS (TTC) spanning climate and clinical domains, LBS outperforms unimodal and multimodal baselines, ***improving numeric accuracy by 13.20%*** on average while producing coherent textual predictions.

2 Preliminaries

In this section, we provide background information on multimodal time-series forecasting, followed by a discussion on our assumed latent state space model and the objective function used to optimize its parameters. A comprehensive discussion of related work can be found in Appendix A.

Problem Setup. Given a temporal series of numerical values $\boldsymbol{y}_{1:t} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t] \in \mathbb{R}^{t \times M}$ and textual data $\boldsymbol{\mathcal{D}}_{1:t} = [\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_t]$ across t time steps, the objective of *multimodal time-series forecasting* is to predict the target values as well as corresponding text for the next H steps:

$$f_\Theta : (\boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t}) \mapsto (\boldsymbol{y}_{t+1:t+H}, \boldsymbol{\mathcal{D}}_{t+1:t+H})$$

Compared to a *unimodal* setup with no textual inputs or outputs, this multimodal setup captures richer predictive targets, modeling textual insights in addition to quantitative forecasts.

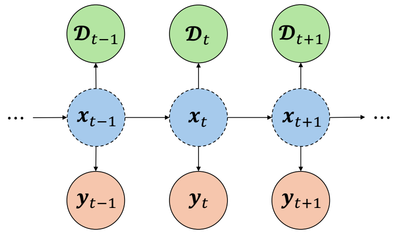


Figure 2: The latent model assumed in LBS. The temporal backbone SSM models the dynamics of states $\boldsymbol{x}_t$, from which multimodal data $\boldsymbol{y}_t$ and $\boldsymbol{\mathcal{D}}_t$ are generated.

Bayesian State Space Model. At each time step t, we assume that a shared unobservable latent state $\boldsymbol{x}_t \in \mathbb{R}^N$ encodes the system's internal condition, which evolves stochastically over time via a **state transition model** $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$. Then, the **numeric emission model** $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$ captures how the target numeric observations are generated from each latent state, and the **textual emission model** $p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)$ models the generation of textual descriptions from the same latent state. All illustration of this dynamical model can be found in Figure 2.

We parameterize the transition distribution $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ as a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t)$, where both the mean $\boldsymbol{\mu}_t$ and diagonal variance $\boldsymbol{\sigma}_t$ are produced by a recurrent neural network (e.g., GRU or LSTM) applied to the previous state $\boldsymbol{x}_{t-1}$. The numeric observation model $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)$ is also modeled as a Gaussian with fixed variance, the mean of which can be computed by passing $\boldsymbol{x}_t$ through a multi-layer perceptron (MLP). To leverage the capability of LLMs in modeling the likelihood of text given textual or non-textual contexts [24], we incorporate a pretrained LLM to model the conditional distribution $p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)$, architectural details for which are shared in the following section.

Training Objective. We train the model by maximizing the evidence lower bound (ELBO) on the joint likelihood of the observed numeric and textual data. The classical ELBO objective naturally extends to our multimodal setting as

$$\log p(\boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T}) \geq \textstyle\sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t \mid \boldsymbol{y}_t, \boldsymbol{\mathcal{D}}_t)} \Big[\underbrace{\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)}_{\text{value likelihood}} + \underbrace{\log p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)}_{\text{text likelihood}} \Big] - \underbrace{\mathrm{KL}(q(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t}) \mid\mid p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}))}_{\text{temporal regularization}}$$

where the variational posterior $q(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t})$ is parametrized via a deep Kalman filter [21, 9, 10], serving as a proxy for the computationally intractable $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t})$. The full derivation and further discussions can be found in Appendix B.

3 LBS: LLM-Integrated Bayesian State Space Model

To integrate LLMs into our latent dynamical model, we must address two technical challenges: **(C1) How can we design the LLM-based filter that estimates the a posteriori state conditioned on text (i.e., $q(\boldsymbol{x}_t \mid \boldsymbol{y}_t, \boldsymbol{\mathcal{D}}_t)$)? (C2) How can we model the likelihood of text conditioned on the latent state (i.e. $p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)$)?** In this section, we detail the architectural components that tackle these challenges, together forming our proposed framework LBS (Figure 1).

3.1 Text-conditioned Posterior State Estimation

Text Compression. To efficiently update our prior state estimates conditioned on text, we adapt a pretrained LLM to perform context compression [2, 6], generating encodings of textual observations into latent state summaries. The core idea is to introduce special tokens unique to the task of summarization and finetune the LLM to allocate critical information into summary tokens for effective posterior inference. More concretely, we first augment the vocabulary of the pretrained LLM with K special learnable tokens `<SUM>`$_k$ that facilitate the task of text compression. To encode a textual observation $\boldsymbol{\mathcal{D}}_t$, we append all K summary tokens after the input sequence $\boldsymbol{\mathcal{D}}_t$, and forward the augmented sequence through the pretrained LLM. The detailed prompts used during compression in our experiments can be found in Appendix C.

After processing the sequence through the LLM, we extract the final hidden states of the K summary tokens, then concatenate along the feature dimension to form a single summary vector of $\boldsymbol{\mathcal{D}}_t$. This vector is then projected through a MLP to obtain a low-dimensional representation $\boldsymbol{s}_t \in \mathbb{R}^N$ that matches the latent states $\boldsymbol{x}_t$ in dimension.

Posterior Inference. Given the summary vector $\boldsymbol{s}_t$, we compute the mean and diagonal covariance of the variational posterior distribution $q(\boldsymbol{x}_t \mid \boldsymbol{y}_t, \boldsymbol{s}_t)$ (assumed to be Gaussian) via a neural Kalman filter parameterized by another MLP [21]. This MLP takes as input the summary vector $\boldsymbol{s}_t$, the corresponding numeric target $\boldsymbol{y}_t$, the prior latent state $\boldsymbol{x}_t$, and outputs the mean and log-variance of the posterior distribution. Note the entire process is end-to-end trainable, hence we finetune the LLM using LoRA [13] to effectively encode forecast-relevant information into the `<SUM>`$_k$ tokens without significantly altering the generative capabilities within its pretrained weights.

3.2 State-conditioned Text Generation

Given the posterior distribution, we use the reparameterization trick [20] to generate $\hat{\boldsymbol{x}}_t \sim q(\boldsymbol{x}_t \mid \boldsymbol{y}_t, \boldsymbol{\mathcal{D}}_t)$, generating Monte-Carlo samples in an end-to-end learnable manner. Then, we can model the posterior state-conditioned textual likelihood $p(\boldsymbol{\mathcal{D}}_t \mid \hat{\boldsymbol{x}}_t)$ by providing a projection of $\hat{\boldsymbol{x}}_t$ as a prefix to the LLM [22], similarly to vision-text instruction tuning frameworks [24]. Specifically, we project the sampled low-dimensional latent state vector $\hat{\boldsymbol{x}}_t$ into a sequence of tokens for the LLM using a linear layer. These projected tokens are then prepended to $\boldsymbol{\mathcal{D}}_t$, effectively allowing the LLM to condition its generation on the state dynamics captured by the temporal SSM backbone.

This design assumes that the temporal backbone is capable of encoding time-specific information such as event structure, trends, or contextual shifts within a compact latent space [1]. By projecting this information into the high-dimensional language space and augmenting it with instruction prompts, we provide the LLM with the necessary information to generate fluent and temporally consistent text.

Method	TTC-CLIMATE							TTC-MEDICAL					
	$H = 1$	2	3	4	5	6	7	$H = 1$	2	3	4	5	6
PatchTST [27]	4.912	5.305	6.021	6.576	6.980	7.170	7.360	5.735	6.757	7.350	7.687	7.996	8.470
NLinear [37]	4.981	6.129	6.501	6.710	6.834	6.916	6.962	5.195	5.279	5.275	5.406	5.562	5.875
NLinear-Text [37]	4.835	5.800	5.951	5.934	6.022	6.024	6.106	5.117	5.143	5.106	5.300	5.492	5.759
TSFLib [25]	6.351	6.446	6.143	6.360	6.096	6.379	**6.002**	6.767	7.066	7.427	7.050	7.165	7.210
TT2TT [19]	5.243	5.955	6.724	7.253	7.678	8.034	7.666	6.689	6.432	6.022	6.483	6.747	6.731
HybridMMF [19]	4.759	5.597	5.906	6.019	6.133	6.027	6.143	5.202	5.472	6.620	6.269	8.673	8.454
LBS (unimodal)	4.224	5.029	5.523	5.855	6.107	6.303	6.473	3.910	4.598	5.047	5.268	5.473	5.654
LBS (multimodal)	**4.117**	**4.908**	**5.341**	**5.627**	**5.833**	**5.998**	6.133	**3.583**	**4.268**	**4.721**	**5.043**	**5.296**	**5.487**

Table 1: Test RMSE results from TTC benchmark. Best results for each prediction horizon H are highlighted in **bold**.

4 Experimental Results

Datasets. We perform experiments on the TIMETEXT CORPUS (TTC [19]), a multimodal time-series forecasting benchmark that covers two distinct domains: TTC-CLIMATE consists of daily temperature measurements at Washington DC with textual weather descriptions. TTC-MEDICAL consists of daily heart rate measurements from hospitalized patients accompanied by nursing notes. Following the original work [19], we use a 8-1-1 train-validation-test split across time for both domains. Further details on the benchmark can be found in Appendix C.

Setup. We compare LBS against existing multimodal forecasters TSFLib [25], TextTime2TextTime (TT2TT [19]), NLinear-Text [37], and HybridMMF [19]. For TSFLib, we use Reformer as its time-series forecasting backbone, as it was the best-performing setup. We also compare against strong unimodal methods PatchTST [27] and NLinear [37]. Note that all baselines are specifically trained for each prediction horizon H, while for LBS which can generalize to arbitrary H, a single model is optimized via stateful training and then evaluated on each possible H. All multimodal models adopt LLaMA3.1-8B [7] as the base LLM, and LBS uses a single-layer GRU [3] with latent dimension 16 as the SSM backbone. Further details on training and model hyperparameters can be found in Appendix C.

Results. Table 1 presents forecasting results across varying prediction horizons. **For most prediction horizons considered, LBS achieves substantial gains over all baselines**, improving the state-of-the-art by 5.13% and 21.28% on TTC-CLIMATE and TTC-MEDICAL on average, respectively. Combined with the fact that a single LBS model is evaluated throughout all horizons, this result highlights the strength and generalizability of SSMs in capturing temporal dependencies, validating our choice of using a probabilistic SSM as our temporal backbone.

When comparing LBS against a unimodal variant of LBS that does not use textual data, we find that the additional modality consistently leads to performance improvements, with 3.82% and 5.41% error reduction for TTC-CLIMATE and TTC-MEDICAL, respectively. This highlights the model's ability to leverage textual information for more accurate posterior inference, leading to sharper and more informed forecasts.

Extended results in Appendix D further show that LBS can generate temporally coherent textual forecasts. Interestingly, we also find that larger LLMs do not always improve forecasting accuracy, but benefits of textual information consistently grow with longer prediction horizons. This suggests that text provides complementary context that stabilizes long-term forecasts and mitigates compounding errors in autoregressive dynamics.

5 Concluding Remarks

We propose LBS, a novel architecture that integrates a Bayesian SSM with pretrained LLMs for multimodal time-series forecasting. By grounding both numeric and textual observations in a shared latent dynamical system, LBS enables coherent forecasting along with uncertainty estimation and flexible prediction horizons. Experiments on the TTC benchmark demonstrate that LBS outperforms existing baselines, with textual data providing greater gains at longer forecasting horizons. Our findings highlight the promise of probabilistic, LLM-integrated SSMs for robust and interpretable forecasting in real-world scenarios.

References

[1] Y. Cai, A. Choudhry, M. Goswami, and A. Dubrawski. Timeseriesexam: A time series understanding exam. *arXiv preprint arXiv:2410.14752*, 2024.

[2] A. Chevalier, A. Wettig, A. Ajith, and D. Chen. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[4] A. Doucet, N. De Freitas, N. J. Gordon, et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001.

[5] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in neural information processing systems*, 30, 2017.

[6] T. Ge, J. Hu, L. Wang, X. Wang, S.-Q. Chen, and F. Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.

[7] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023.

[9] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel. Backprop kf: Learning discriminative deterministic state estimators. *Advances in neural information processing systems*, 29, 2016.

[10] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[11] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[12] P. L. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble kalman filter technique. *Monthly weather review*, 126(3):796–811, 1998.

[13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[14] F. Jia, K. Wang, Y. Zheng, D. Cao, and Y. 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.

[15] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[17] A. Katrompas and V. Metsis. Enhancing lstm models with self-attention and stateful training. In *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1*, pages 217–235. Springer, 2022.

[18] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[19] K. Kim, H. Tsai, R. Sen, A. Das, Z. Zhou, A. Tanpure, M. Luo, and R. Yu. Multi-modal forecaster: Jointly predicting time series and textual data. *arXiv preprint arXiv:2411.06735*, 2024.

[20] D. P. Kingma, M. Welling, et al. Auto-encoding variational bayes, 2013.

[21] R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[22] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[23] Z. Li, X. Lin, Z. Liu, J. Zou, Z. Wu, L. Zheng, D. Fu, Y. Zhu, H. Hamann, H. Tong, et al. Language in the flow of time: Time-series-paired texts weaved into a unified temporal narrative. *arXiv preprint arXiv:2502.08942*, 2025.

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[25] H. Liu, S. Xu, Z. Zhao, L. Kong, H. Prabhakar Kamarthi, A. Sasanur, M. Sharma, J. Cui, Q. Wen, C. Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933, 2024.

[26] M. A. Merrill, M. Tan, V. Gupta, T. Hartvigsen, and T. Althoff. Language models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*, 2024.

[27] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[28] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

[29] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. Van Sloun, and Y. C. Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

[30] N. Roberts, N. Chatterji, S. Narang, M. Lewis, and D. Hupkes. Compute optimal scaling of skills: Knowledge vs reasoning. *arXiv preprint arXiv:2503.10061*, 2025.

[31] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[32] C. Wang, Q. Qi, J. Wang, H. Sun, Z. Zhuang, J. Wu, L. Zhang, and J. Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. *arXiv preprint arXiv:2412.11376*, 2024.

[33] X. Wang, M. Feng, J. Qiu, J. Gu, and J. Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024.

[34] Z. Xu, Y. Bian, J. Zhong, X. Wen, and Q. Xu. Beyond trend and periodicity: Guiding time series forecasting with textual cues. *arXiv preprint arXiv:2405.13522*, 2024.

[35] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[36] C.-H. Yen, H. R. Mendis, T.-W. Kuo, and P.-C. Hsiu. Stateful neural networks for intermittent systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11):4229–4240, 2022.

[37] A. Zeng, M. Chen, L. Zhang, and Q. Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

A Related Work

Multimodal Time-Series Forecasting. With the recent advancements in LLMs, several approaches have emerged to integrate language models with time-series forecasting. [14] introduces a textual data collection pipeline and a modified transformer architecture that uses pre-trained transformers. [26] examines whether LLMs can perform zero-shot time series forecasting with the aid of textual data, extending [8], and concludes that even frontier models still perform poorly. [34] proposes the Text-Guided Time Series Forecasting framework, which integrates news and descriptive textual data for time-series forecasting and introduces a new architecture that leverages a cross-attention layer for modality fusion. [25] presents the Time-MMD benchmark for evaluating text-time series multimodal models and demonstrates that incorporating additional textual data can improve time-series forecasting. [33] develops a reasoning agent for selecting and analyzing textual (news) data, streamlining the text processing pipeline for multimodal time-series forecasting. [19] develops the TimeText Corpus (TTC), a time-aligned text and time-series dataset for multimodal forecasting, along with a hybrid forecasting model (HybridMMF) that jointly predicts both text and time-series data using shared embeddings. [32] introduces ChatTime, a time-series foundation model that facilitates various zero-shot time-series tasks through continuous pretraining and instruction tuning on pretrained language models. [23] presents Texts as Time Series (TaTS), a multimodal time-series forecasting framework that incorporates concurrent textual data by converting it into auxiliary variables. This approach enables seamless integration of text-augmented time series into existing time-series models.

While prior work demonstrates the potential of LLMs for time-series forecasting, ***none integrate them into state-space models—our key contribution.***This integration enhances forecasting performance and enables principled uncertainty quantification.

Bayesian State Space Models. Bayesian state estimation has a long-standing history in control theory and time-series analysis. The classical Kalman filter [16], provides an optimal recursive solution for state estimation in linear dynamical systems with Gaussian noise. To accommodate the nonlinearities common in real-world systems, the Extended Kalman Filter was developed by linearizing nonlinear functions via Taylor expansion around the current estimate. Later, the Unscented Kalman Filter was introduced to improve upon EKF by using deterministic sampling to better capture the mean and covariance of nonlinear transformations [15]. Other sampling-based methods, such as the Ensemble Kalman Filter [12] and Sequential Monte Carlo [4], have further advanced Bayesian filtering in nonlinear and non-Gaussian settings by representing posterior distributions through particle ensembles.

More recently, researchers have sought to combine the structure of state-space models with the flexibility of deep neural networks. For example, KVAE introduced variational approaches to learning latent dynamics in sequential data using neural parameterizations of the transition and emission functions [5]. [28] adapted state-space formulations for multivariate time-series forecasting in large-scale retail demand. In reinforcement learning and model-based control, works such as PlaNet [10], Dreamer [11], and KalmanNet [29] have shown that combining deep neural networks with probabilistic latent dynamics models can yield strong performance across pixel-based partially observable domains.

Despite these advances, existing work largely targets unimodal data like images or numerical signals. In contrast, ***our work is the first to combine pretrained LLMs with probabilistic state-space models for joint forecasting over numeric and textual inputs, extending Bayesian state estimation to the multimodal setting.***

B Implementation Details

B.1 Derivation of Training Objective

Our training objective can be derived using the autoregressive structure of the latent dynamical model as well as Jensen's inequality.

$$
\begin{aligned}
&\log p(\boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T}) \\
&= \log \int_{\boldsymbol{x}_{1:T}} p(\boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T}, \boldsymbol{x}_{1:T}) d\boldsymbol{x}_{1:T} \\
&= \log \int_{\boldsymbol{x}_{1:T}} \prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t) d\boldsymbol{x}_{1:T} \\
&= \log \mathbb{E}_{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T})} \left[\frac{\prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T})} \right] \\
&\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T})} \left[\log \frac{\prod_{t=1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T})} \right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\mathcal{D}}_{1:T})} \left[\sum_{t=1}^{t} \log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) + \log p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t) + \log \frac{p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}{q(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t})} \right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t \mid \boldsymbol{y}_t, \boldsymbol{\mathcal{D}}_t)} \Big[\underbrace{\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)}_{\text{value likelihood}} + \underbrace{\log p(\boldsymbol{\mathcal{D}}_t \mid \boldsymbol{x}_t)}_{\text{text likelihood}} \Big] - \underbrace{\mathrm{KL}(q(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t}) \mid\mid p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}))}_{\text{temporal regularization}}
\end{aligned}
$$

Replacing the computationally intractable posterior distribution $p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t})$, we introduce a variational posterior $q(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\mathcal{D}}_{1:t})$ over the latent states, parameterized via a deep Kalman filter [21, 9, 10]. Intuitively, the training objective effectively balances three essential aspects of Bayesian state estimation. First, the expected likelihood terms ensure fidelity to the observed data by encouraging the latent states to retain enough information to accurately reconstruct both the numeric values and textual descriptions. Second, the KL regularizer imposes temporal coherence by penalizing latent trajectories that deviate too strongly from the prior dynamics controlled by the SSM. Lastly, the variational expectation allows the model to predict under uncertainty in the latent trajectory, inducing more robust and generalizable forecasts.

B.2 Architectural Details

Shared or Separate LLMs? While it is possible to use two separate LLMs for encoding and decoding, for LBS we assume the same LLM weights are shared between the two steps: the LLM that encodes text into compressed embeddings for posterior estimation also serves as the decoder for text generation. This weight sharing not only reduces the computational burden, but also encourages the LLM to encode forecasting-relevant information in a way that it can later reuse for generation, learning prediction and inference in a self-consistent manner.

Stateful single-step training. Ideally, training LBS on long sequences would allow the model to better capture long-range dependencies. However, each timestep t requires passing $\boldsymbol{\mathcal{D}}_t$ through the LLM twice—once for encoding and once for decoding—making naïve long-horizon training computationally expensive. To address this, we adopt *stateful* training [36, 17], where the model is trained on single-step batches under its temporal ordering, with hidden states sampled from the posterior distribution is passed onto the next training iteration. The detailed algorithm and illustration of a single training step can be found in Algorithm 1 and Figure 3.

Algorithm 1: Stateful training step of LBS

```
Input  : Previous state and hidden $(\hat{\boldsymbol{x}}_{t-1}, \boldsymbol{h}_{t-1})$
         Current text and value $(\boldsymbol{\mathcal{D}}_t, \boldsymbol{y}_t)$
Output : Current state and hidden $(\hat{\boldsymbol{x}}_t, \boldsymbol{h}_t)$
Get prior $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t), \boldsymbol{h}_t = \text{SSM}(\hat{\boldsymbol{x}}_{t-1}, \boldsymbol{h}_{t-1})$
Summarize text $\boldsymbol{s}_t = \text{LLM}_{\text{encoder}}(\boldsymbol{\mathcal{D}}_t)$
Get posterior $\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\sigma}}_t) = \text{MLP}_{\text{post}}(\boldsymbol{h}_t, \boldsymbol{y}_t, \boldsymbol{s}_t)$
Sample $\hat{\boldsymbol{x}}_t \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\sigma}}_t)$ via reparameterization
$\mathcal{L}_{\text{val}} = \|\boldsymbol{y}_t - \text{MLP}_{\text{val}}(\hat{\boldsymbol{x}}_t)\|^2$
$\mathcal{L}_{\text{text}} = \text{LLM}_{\text{decoder}}(\hat{\boldsymbol{x}}_t, \boldsymbol{\mathcal{D}}_t)$
$\mathcal{L}_{\text{KL}} = \text{KL}(\mathcal{N}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\sigma}}_t) \mid\mid \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t))$
Update parameters via $\mathcal{L} = \mathcal{L}_{\text{val}} + \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{KL}}$
return $(\hat{\boldsymbol{x}}_t, \boldsymbol{h}_t)$
```

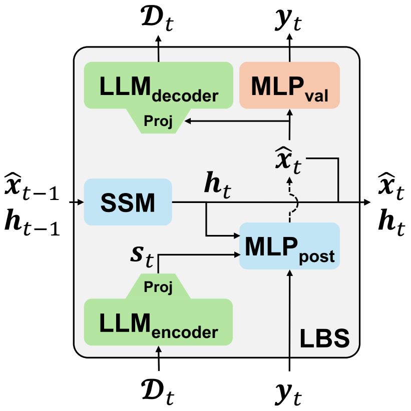


Figure 3: Illustration of a single forward pass through LBS.

C Details on Experimental Setup

Datasets. TTC-CLIMATE is consisted of daily temperature measurements at Washington DC, accompanied by textual weather descriptions spanning from January 1st, 2014 to December 1st, 2023. TTC-MEDICAL stores daily heart rate measurements from 73 patients accompanied by nursing notes writing observations and treatment plans. Each patient data spans an average length of 104 days. Following previous work [19], we train the model on the first 80% of all timestamps, validate on the next 10%, then test on the last 10%.

LLM Prompts. For both TTC-CLIMATE and TTC-MEDICAL experiments, we use the following prompts for text-conditioned posterior estimation and state-conditioned text generation, respectively.

Prompt for text-conditioned posterior estimation

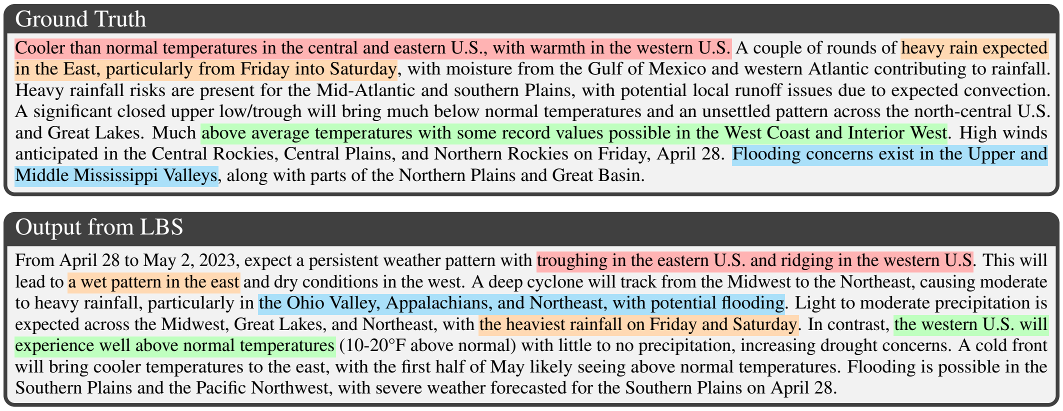
User: Encode the information into a sequence of vectors. `<INSERT TEXT>`
Assistant: `<INSERT SUMMARY TOKENS>`

Prompt for state-conditioned text generation

User: `<INSERT STATE>` Given this belief state, generate a textual forecast.
Date: `<INSERT FORECAST DATE AS YYYY-MM-DD>`
Assistant:

Models. For the SSM backbone, we use a single layer GRU with state and hidden dimensions both equal to 16. For the LLM, we use LLaMA3.1-8B [7] as the default model, and adapt the MLP weights is all layers using LoRA with rank and alpha parameters equal to 8 and 16, respectively. For text compression, we augment and use a set of 8 summary tokens. Similarly for textual forecasting, we project the states into 8 prefix tokens, which are prepended for in-context generation.

Optimization. For all experiments, we use the AdamW optimizer with a learning rate that follows a cosine annealing schedule, starting from 5e-4 and reduced towards 5e-5 during training. We run a maximum of 20 training epochs, and if the model does not improve its validation loss for 5 consecutive epochs, we stop early to prevent further overfitting. Following previous work [10], we use a free nats parmaeter set to 2.5, which effectively clamps the KL loss and thus allows the model to learn meaningful latents at the beginning of training. This free nats parameters is linearly annealed towards zero during training.



Ground Truth

Cooler than normal temperatures in the central and eastern U.S., with warmth in the western U.S. A couple of rounds of heavy rain expected in the East, particularly from Friday into Saturday, with moisture from the Gulf of Mexico and western Atlantic contributing to rainfall. Heavy rainfall risks are present for the Mid-Atlantic and southern Plains, with potential local runoff issues due to expected convection. A significant closed upper low/trough will bring much below normal temperatures and an unsettled pattern across the north-central U.S. and Great Lakes. Much above average temperatures with some record values possible in the West Coast and Interior West. High winds anticipated in the Central Rockies, Central Plains, and Northern Rockies on Friday, April 28. Flooding concerns exist in the Upper and Middle Mississippi Valleys, along with parts of the Northern Plains and Great Basin.

Output from LBS

From April 28 to May 2, 2023, expect a persistent weather pattern with troughing in the eastern U.S. and ridging in the western U.S. This will lead to a wet pattern in the east and dry conditions in the west. A deep cyclone will track from the Midwest to the Northeast, causing moderate to heavy rainfall, particularly in the Ohio Valley, Appalachians, and Northeast, with potential flooding. Light to moderate precipitation is expected across the Midwest, Great Lakes, and Northeast, with the heaviest rainfall on Friday and Saturday. In contrast, the western U.S. will experience well above normal temperatures (10-20°F above normal) with little to no precipitation, increasing drought concerns. A cold front will bring cooler temperatures to the east, with the first half of May likely seeing above normal temperatures. Flooding is possible in the Southern Plains and the Pacific Northwest, with severe weather forecasted for the Southern Plains on April 28.

Figure 4: Example text comparison generated by LBS vs. ground truth text from TTC-CLIMATE. LBS is able to textually forecast key characteristics by contextualizing the LLM on the latent states.

Our training process uses the AdamW optimizer in combination with a cosine decay schedule that initiates at a learning rate of 5e-4 and anneals gradually to 5e-5. Each model is trained for up to 20 epochs, with early termination triggered if validation performance fails to improve over five successive epochs. Inspired by strategies in prior latent sequence modeling [10], we introduce a "free nats" threshold of 2.5 to restrict the KL penalty early in training. This constraint encourages the model to utilize its latent capacity more effectively at initialization and is gradually reduced to zero as optimization proceeds.

Loss weighting. Despite using a small LoRA rank, the number of trainable parameters in the LLM still far exceeds those in the SSM. Consequently, we find that uniform weighting of the loss components in our objective function tends to bias optimization toward the text likelihood term, often overfitting to language modeling while underfitting on structured numerical predictions. Although dynamic or adaptive weighting schemes (e.g., uncertainty-based or gradient norm balancing) could be employed [18], we find that a simple weighting scheme with $\alpha_{\text{val}} = \alpha_{\text{KL}} = 1.0$ and $\alpha_{\text{text}} = 0.1$ provide a good trade-off between tasks without requiring additional tuning.

D Additional Experimental Results

D.1 Text Generation

Beyond forecasting numeric values, **LBS is also capable of generating temporally coherent textual forecasts**. Figure 4 shows a sample forecast generated by LBS on the TTC-CLIMATE dataset. While only provided with the prior state embedding and a simple prompt, LBS can produce context-aware descriptions that align well with the ground-truth dynamics without direct access to previous text. This result highlights the latent states' ability to encode rich semantic structure that can further rationalize model forecasts, and also demonstrates its utility in applications where human-readable justifications are essential alongside quantitative predictions.

D.2 Uncertainty in Forecasts

Setup. In order to observe how LBS allocates uncertainty across forecasting, we compute and report the variance in predictions across 10 states sampled from the prior distribution at each step, during test time on TTC-Climate. We compare results from LBS against those from deterministic HybridMMF.

Results. Figure 5 shows that in contrast to deterministic baselines such as HybridMMF, **LBS provides meaningful uncertainty intervals in addition to accurately capturing the overall trend**. We observe that the predicted variance increases in regions where the ground-truth data shows higher fluctuation (e.g., the early winter), while periods with lower fluctuation leads to lower predicted

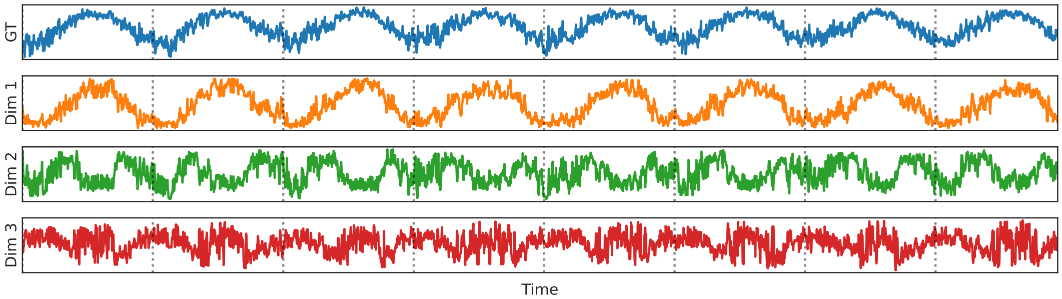


Figure 6: Visualization of ground-truth signals and three t-SNE components of the state trajectory during training on TTC-Climate. The dashed lines indicate yearly intervals. LBS learns states that exhibit the same seasonal patterns as the target values, promoting model transparency.

variance (e.g., the summer). This property makes LBS particularly suitable for real-world forecasting tasks that require assessing confidence in predictions for risk-aware decision making.

D.3 Analysis on Latent State Trajectory

Setup. To evaluate the qualitative dynamics of states learned by LBS, we extract the posterior latent state trajectory learned by LBS on the training set of TTC-Climate. For visualization, we apply t-SNE [31] to the full trajectory and plot the top three components with the highest variance.

Results. As shown in Figure 6, **the latent states in LBS exhibit strong seasonal periodicity that is closely aligned with the ground-truth signal**. This alignment promotes transparency: the learned states are not black-box embeddings but instead encode temporally coherent structure and semantics. Such feature supports straightforward validation of the learned dynamics and enables effective diagnosis of potential errors, especially useful in high-stakes scenarios such as finance or healthcare.

D.4 Effect of LLM Scaling

Setup. As larger LLMs are known to more effectively compress information into compact summaries [2, 6], we verify whether increasing the LLM size also improves forecasting performance by evaluating LBS using a range of backbone LLMs with varying parameter sizes. While larger LLMs are known to exhibit stronger reasoning and generation capabilities, it remains unclear whether these benefits translate to the setting of text-conditioned time-series forecasting. We fix our evaluation domain to TTC-Climate and train LBS while switching the LLM within variants of LLaMA3 (1B, 3B, 8B) [7] and Qwen2.5 (0.5B, 1.5B, 3B, 7B) [35].

Results. Surprisingly, Figure 7 shows that **scaling the LLM does not necessarily lead to better forecasting performance**: for instance, Qwen2.5-7B is consistently outperformed by its 1.5B variant. There are several plausible explanations. First, the relatively low capacity of the SSM may introduce a representational bottleneck, preventing LBS from fully leveraging the richer representations offered by larger LLMs. Second, the task of compressing text into a single significantly lower dimensional vector followed by textual forecasting may not benefit from scaling as with more conventional language tasks such as ques-

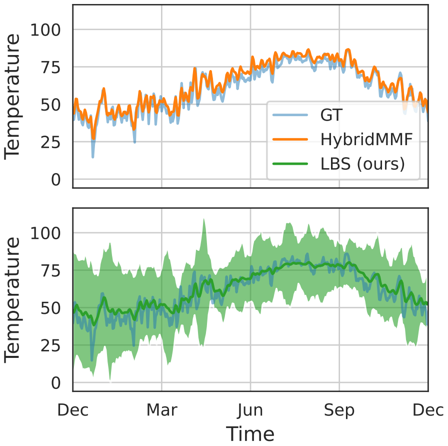


Figure 5: Single-step predictions ($H = 1$) of HybridMMF (top) and LBS (bottom) on the TTC-Climate test set. The shaded region indicates the variance of each prediction of LBS, with true values shown in light blue. Forecasts in the initial winter exhibits relatively larger variance than in the summer, as expected from the high variance in actual data.

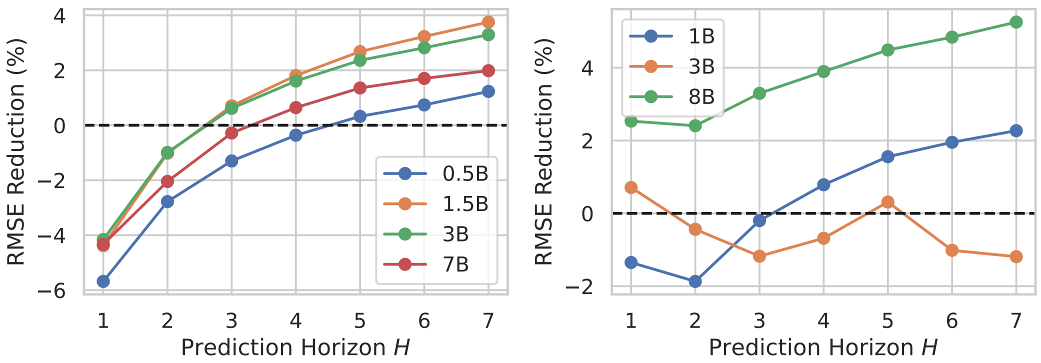


Figure 7: Test RMSE reductions of LBS relative to its unimodal counterpart on TTC-CLIMATE with varying LLMs from the Qwen2.5 (left) and LLaMA3 (right) series. The dashed line indicates the baseline from unimodal LBS. A larger LLM does not consistently lead to better performance, but the gain from textual inputs tends to increase with increasing prediction horizon.

tion answering or code generation [30]. Finally, larger LLMs may tend to memorize training patterns rather than learn generalizable forecasting strategies, diminishing the role of the dynamical model.

Nonetheless, we observe an encouraging overall trend: **the performance gain from incorporating textual information tends to grow with longer prediction horizons**. This suggests that textual information offers complementary context that helps stabilize forecasts over time, making them more robust to compounding noise in autoregressive dynamics.

In summary, our findings highlight potential directions to better integrate LLMs for multimodal time-series forecasting: better posterior estimation strategies or capacity-aligned training of SSMs could allow larger LLMs to be used more effectively.