BEYOND FLAT TAXONOMIES: HIERARCHICAL CAPABILITY PROFILING FOR TIME-SERIES UNDERSTANDING AND REASONING IN LARGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series analysis is increasingly shifting toward large foundation models capable of multimodal perception and complex temporal reasoning. However, existing benchmarks largely rely on flat task taxonomies, making it difficult to systematically evaluate compositional capabilities and diagnose failure modes in temporal understanding. In this work, we propose a hierarchical capability taxonomy that decomposes time series analysis into interdependent dimensions spanning structural perception, feature extraction, temporal reasoning, sequence matching, and cross-modal understanding. Guided by this taxonomy, we construct a real-world multimodal time-series question answering (TSQA) benchmark comprising 1,724 QA pairs across three complementary subsets—**InWild**, **Match**, and **Align**. The dataset is generated through a multi-stage, consistency-verified pipeline integrating numerical signals, visual representations, domain context, and expert validation. We evaluate closed-source large language models (LLMs), open-source LLMs, and time-series-adapted foundation models (TS-LLMs), revealing that current TS-LLMs are largely dominated by backbone model capacity, with specialized time-series encoders providing only marginal gains under existing alignment paradigms, while multimodal inputs and explicit reasoning strategies substantially improve performance. These results highlight both the limitations of current alignment approaches and the importance of capability-oriented evaluation for advancing robust temporal intelligence in large models.[1].

Track: Research

1 INTRODUCTION

Time series data are fundamental to applications in finance, transportation, healthcare, and cloud systems (Zeng et al., 2023; Zhou et al., 2021; Liu et al., 2024a). Traditional research has focused on specialized tasks such as forecasting, classification, anomaly detection, and imputation (Nie et al., 2023; Zhang et al., 2020; 2024). Recently, large foundation models—particularly LLMs and multimodal models—have enabled new paradigms for time series analysis (OpenAI, 2023; Comanici et al., 2025; Anthropic, 2025a; Yang et al., 2024; Team et al., 2025b), including multimodal integration (Liu et al., 2023; Bai et al., 2025) and emerging tasks such as time series description (Zhang et al., 2023), text-assisted forecasting (Jin et al., 2024), and TSQA with reasoning (Wang et al., 2025a; Xie et al., 2024).

Advancing these capabilities requires principled evaluation frameworks. Existing datasets either augment time series with textual annotations (Liu et al., 2024a; Yu et al., 2024) or construct TSQA benchmarks (Wang et al., 2025b; Kong et al., 2025). However, most adopt flat task taxonomies (Wang et al., 2025a; Cai et al., 2024), treating low-level perception and high-level reasoning as independent tasks, which limits fine-grained capability diagnosis. Moreover, many benchmarks focus on narrow domains (Wang et al., 2025b; Dong et al., 2024), leaving cross-domain and out-of-distribution (OOD) generalization underexplored.

[1]Code and data are available at `https://anonymous.4open.science/r/TSQA-Bench-249A/`

To overcome these limitations, we introduce a hierarchical capability taxonomy that organizes time series tasks from structural perception to advanced temporal reasoning. Guided by this framework, we construct a real-world multimodal TSQA benchmark with 1,724 QA pairs across five domains from the LOTSA dataset (Woo et al., 2024b), covering structural awareness, feature analysis, temporal reasoning, sequence matching, and cross-modal understanding (Figure 3). We further propose a multi-stage, consistency-verified generation pipeline for high-quality TSQA construction, enabling fine-grained evaluation of LLMs' temporal understanding and reasoning abilities.

2 Related Work

Recent TS-LLMs explore diverse paradigms for integrating temporal data with LLMs. Some approaches transform time series into textual representations or discrete tokens (Kong et al., 2025; Wang et al., 2025a), while others convert signals into visual formats such as plots or charts for multimodal models (OpenAI, 2024; Zhuang et al., 2024; Zhang et al., 2023; Wang et al., 2023; Liu et al., 2023). These representations often incur high modality overhead with limited performance gains. Alternatively, alignment-based methods map time series into LLM embedding spaces via dedicated encoders and projectors (Chow et al., 2024; Xie et al., 2024), enabling description, QA, and reasoning over temporal data. On the data side, most recent benchmarks primarily target forecasting tasks (Hu et al., 2025; Liu et al., 2025; 2024a; Wang et al., 2024), while TSQA resources remain scarce. Existing TSQA datasets typically focus on narrow domains and adopt flat task taxonomies (Wang et al., 2025a; Kong et al., 2025; Wang et al., 2025b), often relying on fixed templates or single-turn generation (Wang et al., 2025a; Kong et al., 2025; Quinlan et al., 2025). Although iterative generation frameworks have been explored (Xie et al., 2024), they largely retain coarse-grained capability structures, limiting fine-grained evaluation of temporal understanding and reasoning.

3 Methodology

3.1 Multi-dimensional Task Classification Framework

To systematically evaluate temporal understanding and reasoning in large models, we propose a hierarchical, multi-dimensional task taxonomy. Existing TSQA benchmarks typically adopt flat classification schemes Cai et al. (2024), treating low-level perception and high-level reasoning as parallel tasks. Such designs ignore inherent capability dependencies in time series analysis, making it difficult to diagnose whether errors arise from deficient signal understanding or reasoning breakdowns. Inspired by hierarchical capability modeling in multimodal benchmarks Liu et al. (2024b) and compositional generalization theory Hupkes et al. (2020), we decompose time series analysis into five complementary dimensions: structural perception, feature analysis, temporal reasoning, sequence matching, and cross-modal understanding. Each dimension contains multiple subtasks (Appendix A), whose compositions form increasingly complex reasoning scenarios. This hierarchical formulation enables fine-grained capability profiling and systematic identification of bottlenecks in temporal reasoning.

3.2 Principles and Methodology for Dataset Construction

To ensure the benchmark's rigor and practical relevance, we construct three complementary evaluation subsets—**InWild**, **Match**, and **Align**—based on real-world LOTSA data spanning five domains: traffic, cloud operations, climate, economics, and healthcare. Together, these subsets cover the five core analytical dimensions introduced earlier. We adopt a systematic, reasoning-oriented construction framework leveraging a general-purpose LLM, Claude 3.7 Sonnet (Anthropic, 2025a), to generate high-quality, non-templated multimodal QA pairs with flexible formats.

While unconstrained LLM generation may introduce quality concerns, we mitigate this through a multi-stage, principled pipeline. **(1) Multimodal Context Integration.** The model is provided with comprehensive inputs, including raw numerical series, corresponding visualizations, detailed domain metadata (e.g., physical semantics), and pre-computed statistical features (e.g., seasonal strength), ensuring all generated content is grounded in underlying data. **(2) Reasoning-Driven Task Synthesis.** Guided by predefined analytical dimensions (e.g., feature analysis, temporal reasoning), the model converts contextualized inputs into structured Question–Answer–Explanation

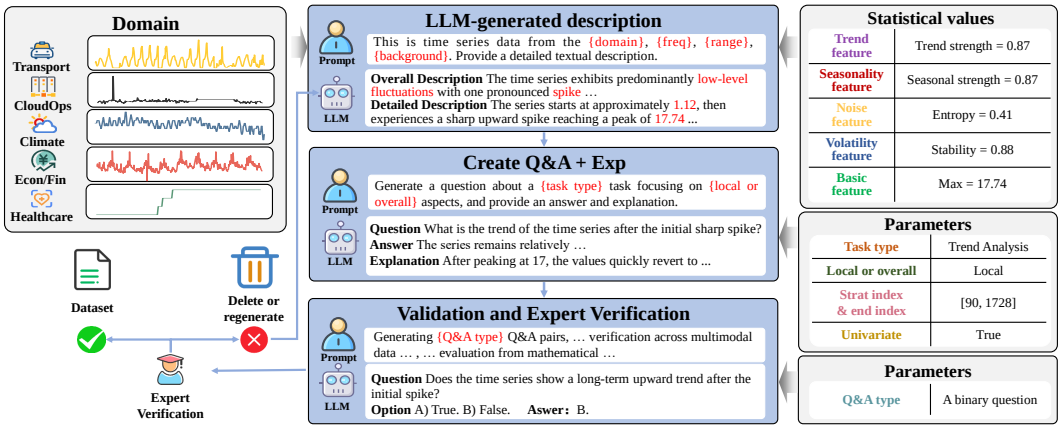


Figure 1: Dataset Construction Pipeline. The flowchart depicts the multi-stage, taxonomy-guided generation framework with human-in-the-loop validation, where real-world time series from multiple domains are enriched and systematically transformed into consistency-verified QA instances.

triples. In contrast to single-turn prompting approaches (e.g., TS-Instruct Quinlan et al. (2025)), this stage explicitly enforces multi-step reasoning by requiring explanations that justify conclusions. **(3) Multi-Dimensional Consistency Verification.** Each QA instance is rigorously validated for mathematical correctness, cross-modal consistency between numerical and visual representations, and coherence of the associated reasoning process.

3.3 Human-In-The-Loop Curation and Statistical Validation

To ensure high dataset quality, ten domain experts conducted systematic human-in-the-loop curation across all subsets. We further performed statistical validation to assess annotation reliability. Fleiss' Kappa (Fleiss, 1971) reached 0.73, indicating substantial inter-annotator agreement (Landis & Koch, 1977). Evaluation robustness was additionally examined through bootstrap analysis (Section C.1) and iterative subsampling (Section C.2). Finally, to mitigate shortcut learning and potential dataset artifacts (Geirhos et al., 2020), we analyzed the relationship between model performance and superficial factors such as sequence length and dimensionality (Section C.3).

3.4 Testing Models Against The Benchmark

Using the curated TSQA benchmark, we evaluate closed-source models, open-source models, and TS-LLMs, reporting *Accuracy* across all subsets. All experiments are repeated five times with temperature set to 1.0 and averaged. **Closed-source models** include Claude 3.7 Sonnet (Anthropic, 2025a), Claude Sonnet 4 (Anthropic, 2025b), Gemini 2.5 Flash/Pro (Comanici et al., 2025), GPT-5 Minimal/High (OpenAI, 2025b), GPT-4.1/4.1 mini (OpenAI, 2025a), and GPT-4o (OpenAI, 2024). **Open-source models** include DeepSeek V3 (DeepSeek-AI, 2024), Kimi K2 (Team et al., 2025a), and the Qwen2.5/Qwen3 series (Yang et al., 2024; 2025). **TS-LLMs** include ChatTS (Xie et al., 2024), ITFormer (Wang et al., 2025b), and ChatTime (Wang et al., 2025a).

4 Results

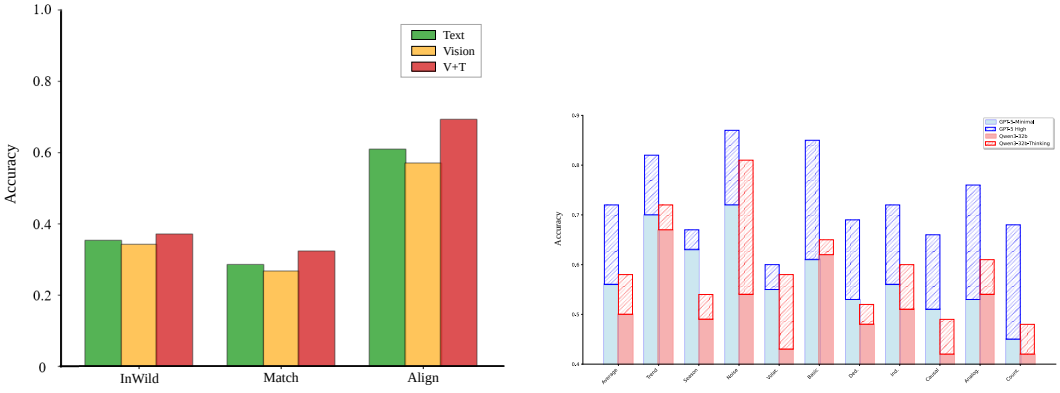
TS-LLMs Are Dominated by Backbone Capacity Rather Than Encoder Design. As shown in Table 1 (see Appendix B for detailed results), general-purpose large models consistently outperform TS-LLMs across all subsets, while TS-LLMs fail to achieve comparable gains despite specialized time-series encoders. For example, ITFormer performs substantially worse than its backbone Qwen2.5-7B. Ablation studies on ChatTS further reveal that variations in encoder architecture, scale, and positional encoding yield only marginal effects (Appendix D), whereas performance is primarily driven by the backbone model capacity. These results indicate that current time series–text alignment paradigms contribute limited benefit beyond the underlying large model.

Table 1: Performance comparison of typical LLMs across different categories on subsets. 'TS' = time series modality. Best results are **underlined and bolded**; second best are **bolded**.

Category	Model	Modality	Average	InWild	Match	Align
Closed-source	GPT-5-High	Text	**0.78**	**0.72**	**0.82**	**0.99**
	Claude-Sonnet-4	Text	0.75	0.71	0.71	0.98
	Gemini 2.5 Pro	Text	0.75	0.68	0.79	0.98
Open-source	DeepSeek-v3	Text	**0.67**	**0.61**	**0.65**	**0.95**
	Qwen2.5-14B	Text	0.59	0.53	0.57	0.88
	Qwen2.5-7B	Text	0.47	0.44	0.40	0.69
TS-LLMs	ChatTS	TS	**0.51**	**0.50**	**0.37**	**0.80**
	ITFormer	TS	0.30	0.33	0.24	0.29

Multimodal Fusion Enhances Temporal Understanding. As reported in Table 4 (Appendix B), Gemini 2.5 Pro achieves 68% accuracy with text-only inputs, 72% with vision-only inputs, and 76% when combining modalities. Qwen2.5-VL-7B exhibits similar trends (Fig. 2a). These findings indicate that multimodal signals complement incomplete temporal information, with performance gains influenced by fusion strategies and model training.

Structured Reasoning and Prompt Signals Outweigh Pure Scaling. As shown in Table 4, scaling Qwen2.5 from 7B to 14B yields noticeable improvements, while further expansion to 32B shows diminishing returns, consistent with scaling laws (Kaplan et al., 2020). In contrast, enabling CoT reasoning for Qwen3 and GPT-5 produces substantial gains on InWild subset (Fig. 2b). Moreover, prompt designs incorporating explicit statistical cues significantly enhance inference, highlighting prompt engineering as a practical avenue for strengthening temporal reasoning in large models.



(a) Qwen2.5-VL-7B on the our TSQA-Bench

(b) Thinking versus Non-thinking

Figure 2: (a) the evaluation results of Qwen2.5-VL-7B across subsets. (b) the accuracy gains of GPT-5 and Qwen3-32B on the **InWild**, comparing performance thinking and non-thinking mode.

5 Conclusion And Future Work

This paper presents a hierarchical capability-driven TSQA benchmark with 1,724 QA pairs for evaluating temporal understanding and reasoning in large models. Results show that TS-LLM performance is primarily driven by backbone capacity, with current alignment paradigms offering only marginal encoder gains, while multimodal inputs and explicit reasoning substantially improve inference. Nonetheless, models remain challenged by fine-grained localization and complex temporal reasoning. Future work will expand task coverage, extend temporal horizons, and explore improved time series–language alignment. We hope this benchmark advances capability-oriented evaluation for robust temporal understanding.

REFERENCES

Anthropic. Claude 3.7 sonnet and claude code. https://www.anthropic.com/news/claude-3-7-sonnet, February 2025a. Accessed: 2025-09-18.

Anthropic. Introducing claude 4: Claude opus 4 and claude sonnet 4. https://www.anthropic.com/news/claude-4, May 2025b. Accessed: 2025-09-18.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. Timeseriesexam: A time series understanding exam. *arXiv preprint arXiv:2410.14752*, 2024.

Winnie Chow, Lauren Gardiner, Haraldur T Hallgrímsson, Maxwell A Xu, and Shirley You Ren. Towards time series reasoning with llms. *arXiv preprint arXiv:2409.11376*, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Zihan Dong, Xinyu Fan, and Zhiyuan Peng. Fnspid: A comprehensive financial news dataset in time series, 2024. URL https://arxiv.org/abs/2402.06698.

Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Rakshitha Wathsadini Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wEc1mgAjU-.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, 2018.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing LLMs capabilities in time series. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=syC2764fPc.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

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. In *International Conference on Learning Representations (ICLR)*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. Time-mqa: Time series multi-task question answering with context enhancement. *arXiv preprint arXiv:2503.01875*, 2025.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pp. 159–174, 1977.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. TimeCMA: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. Time-mmd: A new multi-domain multimodal dataset for time series analysis, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

Tung Nguyen, Jason Jewik, Hritik Bansal, Prakhar Sharma, and Aditya Grover. Climatelearn: Benchmarking machine learning for weather and climate modeling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 75009–75025. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ed73c36e771881b232ef35fa3a1dec14-Paper-Datasets_and_Benchmarks.pdf.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

OpenAI. Gpt-4 research. https://openai.com/index/gpt-4-research/, March 2023. Accessed: 2025-09-18.

OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, May 2024. Accessed: 2025-09-18.

OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, April 2025a. Accessed: 2025-09-18.

OpenAI. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/, August 2025b. Accessed: 2025-09-18.

Paul Quinlan, Qingguo Li, and Xiaodan Zhu. Chat-ts: Enhancing multi-modal reasoning over time-series and natural language data. *arXiv preprint arXiv:2503.10883*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang,

Jinhong Wang, Shengjie Wang, Shuyi Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Xiaofei Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yangkun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Huabin Zheng, Shaojie Zheng, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2025a. URL https://arxiv.org/abs/2507.20534.

MiniCPM Team, Chaojun Xiao, Yuxuan Li, Xu Han, Yuzhuo Bai, Jie Cai, Haotian Chen, Wentong Chen, Xin Cong, Ganqu Cui, et al. Minicpm4: Ultra-efficient llms on end devices. *arXiv preprint arXiv:2506.07900*, 2025b.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025a.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. In *Neural Information Processing Systems*, 2024.

Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang Li, and Zhongyu Wei. Itformer: Bridging time series and natural language for multi-modal qa with large-scale multitask dataset. In *International Conference on Machine Learning (ICML)*, 2025b.

Ziao Wang, Yuhang Li, Junda Wu, Jaehyeon Soon, and Xiaofeng Zhang. Finvis-gpt: A multimodal large language model for financial chart analysis. *arXiv preprint arXiv:2308.01430*, 2023.

Gerald Woo, Chenghao Liu, Akshat Kumar, and Doyen Sahoo. Pushing the limits of pre-training for time series forecasting in the cloudops domain, 2024a. URL https://openreview.net/forum?id=ZkEsEFFUyo.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024b.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Han Yu, Peikun Guo, and Akane Sano. Ecg semantic integrator (esi): A foundation ecg model pretrained with llm-enhanced cardiological text. *arXiv preprint arXiv:2405.19366*, 2024.

Zhen Zeng, Rachneet Kaur, Suchetha Siddagangappa, Saba Rahimi, Tucker Balch, and Manuela Veloso. Financial time series forecasting using cnn and transformer. *arXiv preprint arXiv:2304.04912*, 2023.

Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6845–6852, 2020.

Yunkai Zhang, Yawen Zhang, Ming Zheng, Kezhen Chen, Chongyang Gao, Ruian Ge, Siyuan Teng, Amine Jelloul, Jinmeng Rao, Xiaoyuan Guo, et al. Insight miner: A time series analysis dataset for cross-domain alignment with natural language. In *NeurIPS 2023 AI for Science Workshop*, 2023.

Zhenwei Zhang, Ruiqi Wang, Ran Ding, and Yuantao Gu. Unravel anomalies: an end-to-end seasonal-trend decomposition approach for time series anomaly detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5415–5419, 2024. doi: 10.1109/ICASSP48485.2024.10446482.

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 *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pp. 11106–11115. AAAI Press, 2021.

Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, Ruiqi Wang, Jiawei Zhang, and Yuantao Gu. See it, think it, sorted: Large multimodal models are few-shot time series anomaly analyzers. *arXiv preprint arXiv:2411.02465*, 2024.

TABLE OF CONTENTS

A DATASET AND TAXONOMY DETAILS

A.1 DATASET DESCRIPTION AND SPECIALIZED DESIGN

Our benchmark includes three subsets: **(1) InWild,** assesses advanced time-series understanding and reasoning. By combining Structural Perception, Feature Analysis, and Temporal Reasoning, we generated 1,084 QA pairs spanning 140 distinct task types. **(2) Match,** evaluates time-series similarity matching and morphological correspondence. We designed four subtasks with progressive difficulty levels (see Table 2), totaling 400 QA pairs. The Dynamic Time Warping (DTW) algorithm quantifies morphological differences and guides both LLM generation and consistency verification. **(3) Align,** focuses on Cross-Modal Understanding. It contains 240 bidirectional QA pairs that test the conversion between numerical series and natural language.

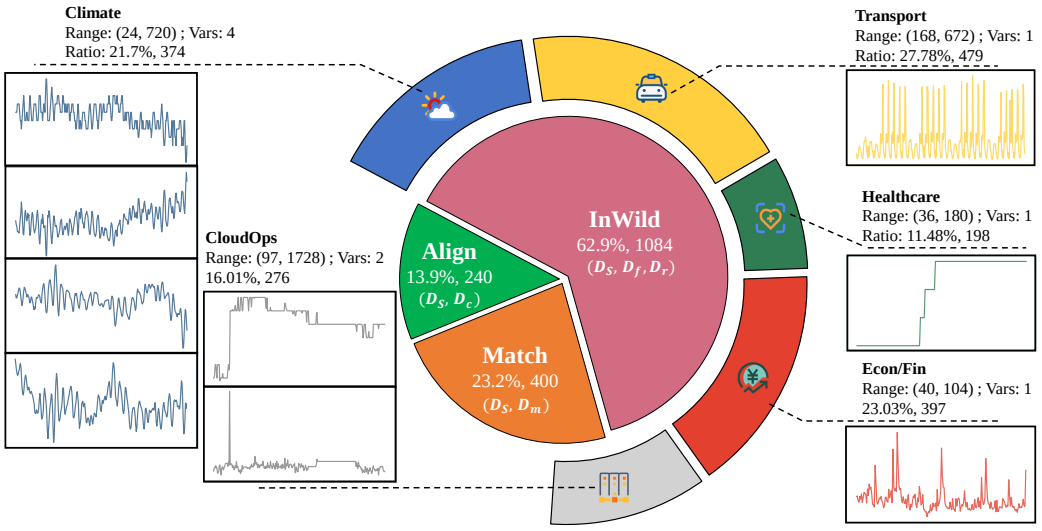


Figure 3: Overview of dataset distribution

A.2 REAL-WORLD DATASET SOURCES

Our benchmark is constructed from the LOTSA (Woo et al., 2024a; Godahewa et al., 2021; Nguyen et al., 2023; Woo et al., 2024b) dataset collection. To ensure broad domain coverage while maintaining representativeness and high quality, we selected five major domains: Transport, Cloud Operations, Climate, Economics, and Healthcare. Representative datasets from these domains include Traffic Hourly, Alibaba Cluster Trace 2018, ERA5 2018, M4 Weekly, Hospital, and COVID deaths, with statistical parameters summarized in Table 3. Their details are as follows.

Transport (Traffic Hourly) The dataset originates from the California Department of Transportation. It records hourly highway occupancy rates from multiple sensors in the San Francisco Bay Area over

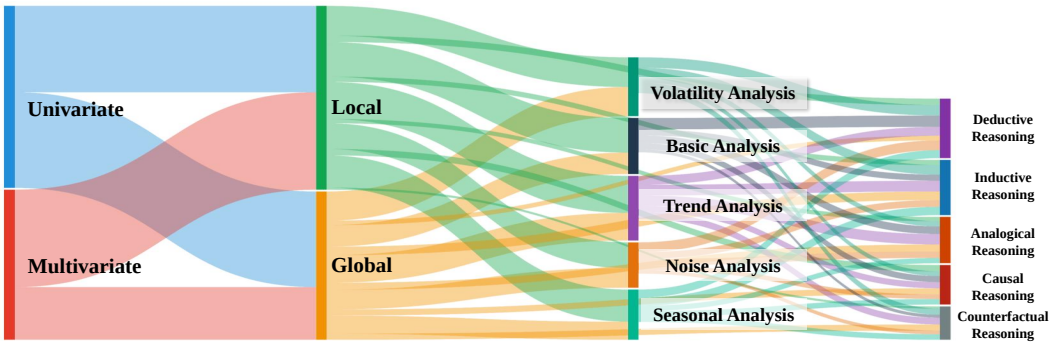


Figure 4: Sankey Diagram of Subtask Labels in the InWild Subset. This diagram illustrates the relationships and transitions between subtask labels in the InWild subset, highlighting their interdependencies in a clear and intuitive way.

a 48-month period (2015–2016). It contains 862 time series, each with 17,376 points in the range [0,1]. Because of the long time span and the strong seasonal patterns, we applied a sliding window with a maximum length of 672 points. Values were scaled by a factor of 100 and rounded to two decimal places.

Cloud Operations (Alibaba Cluster Trace 2018) This dataset describes CPU and memory utilization in a cluster of about 4,000 machines over eight days (from January 2 to January 8, 2018), sampled at five-minute intervals. It consists of 58,409 pairs of time series. Theoretical sequence length is 1,728 points, although some sequences are shorter due to missing samples (100–1,728 points). Values are within [0,100]. Because sequence lengths are moderate, no windowing was applied. Instead, we randomly sampled sequences and retained two decimal places.

Climate (ERA5 2018) The dataset comes from the European Centre for Medium-Range Weather Forecasts. It provides hourly global reanalysis data for 2018 at 2.8125° resolution (64×128 grid points), covering 45 variables across seven pressure levels (50, 250, 500, 600, 700, 850, and 925 hPa). Each time series pair has 8,736 points. We selected relative humidity and temperature from the seven pressure levels, with values within [0,100]. To capture spatial diversity, we randomly sampled 50 locations worldwide and then applied sliding windows of length 720. All values were rounded to two decimal places.

Economics (M4 Weekly) The dataset is a subset of the M4 Competition (2018), which consists of 100,000 time series across different frequencies. The weekly subset includes 359 economic and business-related series, such as sales, demand, and index values. Sequence lengths range from 80 to 2,597 points. To preserve potential seasonalities and balance sequence lengths, we used a sliding window with a maximum length of 104 points, approximately two years in length. Shorter series were kept in full. Values were rounded to two decimal places.

Healthcare (Hospital & COVID deaths) The Hospital dataset records monthly patient counts related to medical products from January 2000 to December 2006. It contains 767 series of length 72. We applied sliding windows with common monthly cut lengths of 36, 60, and 72 points. All values were rounded to two decimal places. The COVID deaths dataset is sourced from the Johns Hopkins University repository. It contains cumulative daily death counts for countries and regions from January 22 to August 20, 2020. It consists of 266 daily series, each 182 points long. We applied a sliding window with a maximum length of 180 points. Values were rounded to two decimal places.

Table 2: Multi-dimensional Time Series Task Taxonomy.

Dimensions	Subtasks	Definition
Structural Awareness (D_s)	Non-Stationarity	Analyzes statistical properties of concatenated subsequences.
	Local-Global	Locates and analyzes specific sequence segments.
	Univariate-Multivariate	Processes and analyzes multiple time series data jointly.
Feature Analysis (D_f)	Trend Analysis	Identifies long-term directional patterns and trend strength.
	Seasonality Analysis	Captures seasonal patterns and seasonality strength.
	Noise Analysis	Distinguishes random fluctuations from signal components.
	Volatility Analysis	Quantifies temporal variability and instability.
	Basic Analysis	Computes fundamental statistics (mean, variance, range, etc.).
Temporal Reasoning (D_r)	Deductive Reasoning	Applies general rules to infer properties of specific intervals.
	Inductive Reasoning	Generalizes characteristics from observed sequences.
	Causal Reasoning	Identifies causal or lead-lag relationships between series.
	Analogical Reasoning	Infers similarity by comparing temporal patterns.
	Counterfactual Reasoning	Predicts outcomes under hypothetical changes.
Sequence Matching (D_m)	Isomorphic Matching	Finds the most similar sequence under equal-length constraints.
	Robust Matching	Robustly matches patterns under preprocessing transformations.
	Localization Matching	Locates target patterns within longer sequences.
	Reverse Matching	Recognizes similarity under temporal reversal.
Cross-Modal Understanding (D_c)	Time-series to Semantic	Converts time series patterns into textual descriptions.
	Semantic to Time-series	Maps textual descriptions to corresponding time series data.

Table 3: Statistical parameters of subsets in LOTSA.

Dataset	Domain	Frequency	#Time Series	#Obs.	#Vars
Traffic Hourly	Transport	H	862	14,978,112	1
Alibaba Cluster Trace 2018	CloudOps	5T	58,409	95,192,530	2
ERA5 2018	Climate	H	245,760	2,146,959,000	45
M4 Weekly	Economics	W	359	366,912	1
Hospital	Healthcare	M	767	55,224	1
COVID Deaths	Healthcare	D	266	48,412	1

B FULL RESULTS

Table 4: Performance of different models on the **InWild** subset. ^{cot} denotes *thinking* mode. [1] denotes models evaluated without any time series input. [2] denotes ChatTS without built-in statistical computation module. −VL = Vision-Language. 'TS' stands for time series modality, as time series-specific models introduce a TS encoder. '–' indicates that the model failed to respond correctly. In reasoning tasks, abbreviations are: Ded. (Deductive), Ind. (Inductive), Analog. (Analogical), and Count. (Counterfactual). Best and second-best results within each category are **underlined** and **bolded**, respectively.

Category	Model Name	Modality	Average	Feature Analysis						Temporal Reasoning					
				Acc.	Trend	Season	Noise	Volat.	Basic	Acc.	Ded.	Ind.	Causal	Analog.	Count.
Open-source	DeepSeek-V3	Text	**0.61**	**0.67**	**0.73**	**0.57**	0.79	**0.55**	**0.75**	**0.59**	**0.60**	**0.63**	**0.52**	0.58	**0.57**
	Kimi-K2	Text	**0.63**	**0.69**	0.71	**0.63**	**0.85**	**0.58**	**0.71**	**0.61**	**0.62**	**0.65**	**0.56**	**0.64**	**0.55**
	Qwen3-32b^{cot}	Text	0.58	0.65	**0.72**	0.54	**0.81**	**0.58**	0.65	0.55	0.52	0.60	0.49	**0.61**	0.48
	Qwen3-32b	Text	0.50	0.56	0.67	0.49	0.54	0.43	0.62	0.48	0.48	0.51	0.42	0.54	0.42
	Qwen3-8b^{cot}	Text	0.50	0.57	0.54	0.51	0.74	0.54	0.55	0.47	0.41	0.52	0.43	0.49	0.48
	Qwen3-8b	Text	0.45	0.48	0.45	0.43	0.65	0.34	0.58	0.44	0.48	0.46	0.42	0.45	0.37
	Qwen2.5-32b	Text	0.53	0.62	0.66	0.53	0.77	0.52	0.62	0.49	0.47	0.53	0.51	0.50	0.44
	Qwen2.5-14b	Text	0.53	0.61	0.60	0.55	0.73	**0.58**	0.63	0.49	0.48	0.53	0.48	0.53	0.41
	Qwen2.5-7b	Text	0.44	0.45	0.61	0.48	0.35	0.35	0.42	0.44	0.41	0.47	0.43	0.45	0.44
	Qwen2.5-7b-VL	Text	0.37	0.37	0.41	0.32	0.44	0.28	0.38	0.37	0.38	0.35	0.40	0.35	0.36
	Qwen2.5-7b-VL	Vision	0.36	0.37	0.39	0.44	0.37	0.30	0.33	0.35	0.38	0.36	0.36	0.33	0.33
	Qwen2.5-7b-VL	V+T	0.39	0.41	0.48	0.41	0.45	0.30	0.40	0.38	0.39	0.37	0.43	0.36	0.35
	Qwen2.5-32b[1]	Text	0.34	0.34	0.40	0.40	0.24	0.33	0.31	0.34	0.35	0.35	0.38	0.31	0.30
Closed-source	Claude-3.7-Sonnet	Text	0.69	0.78	0.81	**0.72**	0.85	0.73	0.81	0.65	0.66	0.67	0.58	0.71	0.60
	Claude-3.7-Sonnet	Vision	0.69	0.75	0.82	0.67	0.82	0.70	0.75	0.66	0.64	0.72	0.62	0.71	0.60
	Claude-3.7-Sonnet	V+T	**0.73**	0.79	**0.84**	**0.71**	0.87	0.70	0.85	**0.71**	0.66	**0.76**	**0.70**	0.74	0.66
	Claude-Sonnet-4	Text	0.71	0.79	0.78	0.67	**0.91**	0.70	**0.89**	0.68	0.67	0.75	0.60	0.69	0.63
	Claude-Sonnet-4	Vision	0.67	0.74	0.84	0.59	0.87	0.63	0.79	0.64	0.59	0.75	0.55	0.66	0.60
	Claude-Sonnet-4	V+T	0.71	0.78	0.81	0.64	**0.90**	0.72	0.86	0.68	0.65	**0.76**	0.62	0.72	0.66
	Gemini-2.5-Pro	Text	0.68	0.73	0.76	0.69	0.83	0.66	0.72	0.66	0.65	**0.76**	0.56	0.68	0.63
	Gemini-2.5-Pro	Vision	0.72	**0.80**	**0.84**	0.63	0.89	**0.77**	0.85	0.69	**0.69**	**0.76**	0.60	0.71	0.64
	Gemini-2.5-Pro	V+T	**0.76**	**0.83**	**0.86**	**0.71**	0.87	**0.76**	**0.92**	**0.73**	**0.75**	**0.78**	0.62	**0.77**	**0.69**
	Gemini-2.5-Flash	Text	0.50	0.50	0.53	0.40	0.59	0.45	0.55	0.51	0.47	0.57	0.45	0.56	0.45
	GPT-5-High	Text	0.72	0.76	0.82	0.67	0.87	0.60	0.85	0.70	0.69	0.72	**0.66**	**0.76**	**0.68**
	GPT-5-Minimal	Text	0.56	0.64	0.70	0.63	0.72	0.55	0.61	0.52	0.53	0.56	0.51	0.53	0.45
	GPT-4.1	Text	0.58	0.64	0.63	0.57	0.76	0.58	0.68	0.56	0.58	0.61	0.52	0.52	0.55
	GPT-4.1	Vision	0.58	0.61	0.70	0.46	0.70	0.51	0.67	0.57	0.57	0.61	0.57	0.55	0.56
	GPT-4.1	V+T	0.63	0.67	0.73	0.55	0.85	0.60	0.68	0.62	0.62	0.69	0.59	0.59	0.56
	GPT-4.1-Mini	Text	0.59	0.66	0.69	0.49	0.88	0.55	0.73	0.56	0.56	0.60	0.54	0.56	0.49
	GPT-4.1-Mini	Vision	0.49	0.51	0.56	0.37	0.54	0.51	0.55	0.49	0.51	0.45	0.52	0.50	0.48
	GPT-4.1-Mini	V+T	0.58	0.64	0.68	0.54	0.75	0.53	0.73	0.56	0.59	0.61	0.55	0.54	0.44
	GPT-4o	Text	0.62	0.70	0.74	0.61	0.79	0.61	0.75	0.58	0.59	0.62	0.50	0.66	0.50
	GPT-4o	Vision	0.59	0.67	0.76	0.56	0.74	0.64	0.66	0.56	0.55	0.62	0.51	0.57	0.50
	GPT-4o	V+T	0.63	0.71	0.75	0.59	0.85	0.61	0.77	0.59	0.59	0.65	0.55	0.63	0.53
TS-LLMs	ChatTS	TS	0.50	0.55	0.50	0.61	0.53	0.54	0.58	0.48	0.51	0.50	0.49	0.44	0.45
	ChatTS[2]	TS	0.48	0.51	0.53	0.55	0.48	0.52	0.50	0.48	0.50	0.50	0.52	0.42	0.43
	ITFormer	TS	0.33	0.37	0.29	0.31	0.37	0.29	0.36	0.36	0.37	0.40	0.35	0.35	0.35
	ChatTime	TS	–	–	–	–	–	–	–	–	–	–	–	–	–
Human	Experts	–	0.67	0.71	0.59	0.67	0.75	0.75	0.80	0.66	0.66	0.72	0.63	0.67	0.58

Table 5: Performance of different models on the **Match** subset. ^{cot} denotes *thinking* mode. [1] denotes models evaluated without any time series input. [2] denotes ChatTS without built-in statistical computation module. −VL = Vision-Language. 'TS' stands for time series modality, as time series-specific models introduce a TS encoder. '–' indicates that the model failed to respond correctly. **Bold underlined** values indicate the best performance within each category for each metric, and **bold** values indicate the second-best performance.

Category	Model Name	Modality	Average	Isomorphic	Robust	Localization	Reverse
Open-source	DeepSeek-V3	Text	**0.65**	**0.90**	**0.79**	**0.57**	0.35
	Kimi-K2	Text	0.60	0.78	0.66	0.50	0.44
	Qwen3-32b^{cot}	Text	0.60	0.80	0.64	0.42	**0.52**
	Qwen3-32b	Text	0.50	0.72	0.58	0.36	0.35
	Qwen3-8b^{cot}	Text	0.50	0.62	0.56	0.36	**0.46**
	Qwen3-8b	Text	0.42	0.56	0.48	0.31	0.32
	Qwen2.5-32b	Text	**0.62**	**0.84**	**0.72**	**0.53**	0.41
	Qwen2.5-14b	Text	0.57	0.78	0.61	0.42	0.45
	Qwen2.5-7b	Text	0.40	0.45	0.49	0.35	0.29
	Qwen2.5-7b-VL	Text	0.30	0.31	0.36	0.27	0.28
	Qwen2.5-7b-VL	Vision	0.28	0.30	0.31	0.28	0.25
	Qwen2.5-7b-VL	V+T	0.34	0.40	0.41	0.27	0.30
	Qwen2.5-32b[1]	Text	0.25	0.25	0.25	0.25	0.25
Closed-source	Claude-3.7-Sonnet	Text	0.74	0.93	**0.81**	**0.67**	0.54
	Claude-Sonnet-4	Text	0.71	0.85	0.79	**0.61**	0.60
	Gemini-2.5-Pro	Text	**0.79**	**0.96**	0.80	0.59	**0.80**
	Gemini-2.5-Flash	Text	0.44	0.63	0.57	0.28	0.26
	GPT-5-High	Text	**0.82**	**0.98**	**0.81**	0.60	**0.86**
	GPT-5-Minimal	Text	0.57	0.84	0.67	0.53	0.26
	GPT-4.1	Text	0.67	0.89	**0.82**	0.55	0.40
	GPT-4.1-Mini	Text	0.63	0.90	0.78	0.44	0.40
	GPT-4o	Text	0.50	0.68	0.60	0.41	0.34
	GPT-4o	Vision	0.45	0.56	0.50	0.34	0.38
	GPT-4o	V+T	0.55	0.79	0.64	0.38	0.38
TS-LLMs	ChatTS	TS	0.37	0.47	0.40	0.24	0.36
	ChatTS[2]	TS	0.32	0.46	0.41	0.22	0.20
	ITFormer	TS	0.24	0.16	0.25	0.25	0.29
	ChatTime	TS	–	–	–	–	–

Table 6: Performance of the ChatTS model on the **Match** subset across different time series length ranges. *Total* corresponds to the range [13,504], and "–" indicates that no questions fall into the given length range for that task.

Length Range	Isomorphic	Robust	Localization	Reverse
Total	0.47	0.40	0.24	0.36
[64, 1024]	**0.53**	**0.57**	0.23	**0.44**
[256, 512]	–	–	**0.36**	–

Table 7: Performance of different models on the **Align** subset. ^{cot} denotes *thinking* mode. [1] denotes models evaluated without any time series input. [2] denotes ChatTS without built-in statistical computation module. `-VL` = Vision-Language. 'TS' stands for time series modality, as time series-specific models introduce a TS encoder. '–' indicates that the model failed to respond correctly. **Bold underlined** values indicate the best performance within each category for each metric, and **bold** values indicate the second-best performance.

Category	Model Name	Modality	Average	TS→Sem	Sem→TS
Open-source	DeepSeek-V3	Text	**0.95**	**0.95**	**0.94**
	Kimi-K2	Text	**0.95**	**0.94**	**0.96**
	Qwen2.5-32b	Text	0.93	0.92	**0.94**
	Qwen2.5-14b	Text	0.88	0.87	0.88
	Qwen2.5-7b	Text	0.69	0.68	0.71
	Qwen3-32b^{cot}	Text	0.89	0.92	0.86
	Qwen3-32b	Text	0.87	0.86	0.88
	Qwen3-8b^{cot}	Text	0.86	0.88	0.83
	Qwen3-8b	Text	0.79	0.74	0.84
	Qwen2.5-7b-VL	Text	0.64	0.68	0.59
	Qwen2.5-7b-VL	Vision	0.60	0.61	0.60
	Qwen2.5-7b-VL	V+T	0.73	0.78	0.67
	Qwen2.5-32b[1]	Text	0.27	0.29	0.26
Closed-source	Claude-3.7-Sonnet	Text	0.97	0.97	**0.98**
	Claude-Sonnet-4	Text	**0.98**	**0.98**	**0.99**
	Gemini-2.5-Pro	Text	**0.98**	0.97	**0.99**
	Gemini-2.5-Flash	Text	0.94	0.94	0.95
	GPT-5-High	Text	**0.99**	**0.99**	**0.99**
	GPT-5-Minimal	Text	0.97	0.97	**0.98**
	GPT-4o	Text	0.96	0.96	0.97
	GPT-4o-Mini	Text	0.86	0.82	0.90
TS-LLMs	ChatTS	TS	0.80	0.68	0.91
	ChatTS[2]	TS	0.45	0.49	0.42
	ITFormer	TS	0.29	0.32	0.26
	ChatTime	TS	–	–	–

C STATISTICAL ROBUSTNESS ANALYSIS

To assess the testing stability, robustness, and validity of our dataset, we conducted comprehensive statistical evaluations and bias analyses: **Bootstrap Confidence Interval**, **Iterative Subsampling Analysis**, and **Assessment of Dataset Artifacts**. These experiments evaluate the dataset's testing stability, the adequacy of its scale, and its resistance to spurious correlations, respectively.

C.1 BOOTSTRAP CONFIDENCE INTERVAL

To evaluate the reliability of model performance and quantify its uncertainty, we adopted the non-parametric bootstrap method. Specifically, the experimental setup is as follows: Given a test set of size D, we generated $N = 1000$ bootstrap samples, also of size D, through sampling with replacement.

We selected a few representative models (covering closed-source, open-source LLMs and TS-LLMs) for the experiments. We then calculated their respective accuracy scores to obtain an empirical distribution for this metric. Based on this distribution, we report the mean accuracy, standard deviation (std), and the 95% confidence interval (CI).

Table 8: Bootstrap confidence interval of different models on the dataset.

Models	Mean	Std	CI Low	CI High
Gemini-1.5-Pro (text)	0.7235	0.0094	0.7059	0.7422
GPT-4o (text)	0.6059	0.0103	0.5864	0.6265
DeepSeekV3 (text)	0.6366	0.0100	0.6165	0.6562
Qwen2.5-32b (text)	0.5789	0.0104	0.5580	0.5986

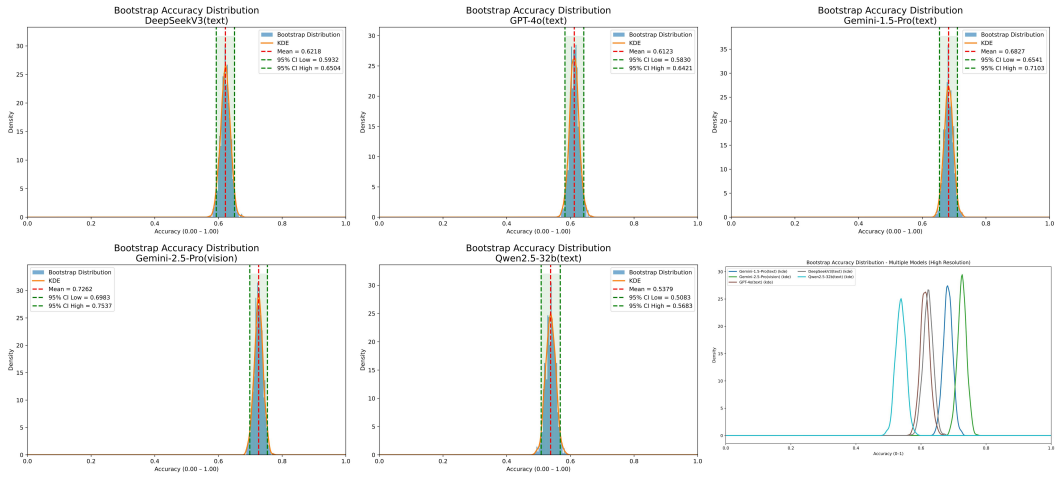


Figure 5: Visualization of the accuracy distribution and confidence intervals for different representative models on the InWild subset.

The experimental results (Table 8 and Figure 5) indicate that the performance evaluations across all models exhibit low statistical dispersion. Specifically, the bootstrapping experiment shows that the standard deviation (std) ranges only between $0.01 \sim 0.02$, and the width of the 95% CI remains within a narrow range (approximately $0.5 \sim 0.8$ percentage points).

These tight error bounds strongly confirm the statistical robustness of our dataset. It demonstrates that the benchmark is insensitive to data sampling variance and can provide stable and reproducible evaluation results for cross-model and cross-capability comparisons. Furthermore, to further suppress the stochastic noise from model generations, we also introduced a mechanism of multiple sampling and majority voting during the evaluation, thereby establishing a more robust performance baseline.

C.2 Iterative Subsampling Analysis

To investigate the relationship between evaluation stability and dataset scale, and to estimate the minimal sample size required to yield robust results, we conducted an Iterative Subsampling Analysis focusing on the representative model, `Gemini-2.5-Pro`, with text input.

The specific experimental setup is as follows: For a given dataset size D, we set the subsampling size S as a variable that progressively increases from an initial value up to D, with an increment step of $T = 20$. At each fixed size S, we perform $N = 50$ independent repetitions of sampling, and calculate the mean, standard deviation (std), and coefficient of variation (CV) of the model's performance.

We use the coefficient of variation ($CV = \sigma/\mu$) as the core metric to measure evaluation stability. The dataset size S is deemed to possess sufficient statistical stability when the CV curve, as S increases, shows a descending trend and falls below a pre-set convergence threshold of $\tau = 0.02$. [2]

Table 9: Comparison of the minimum required sample size for stable assessment versus the actual sample size in the subsampling analysis experiment, along with the model's mean, standard deviation, and coefficient of variation under the actual sample size across three subsets.

Dataset	Full Sample	Min Sample	Mean	Std	CV
Align	240	60	0.9813	0.0100	0.0041
Match	400	260	0.7795	0.0199	0.0047
InWild	1084	600	0.6811	0.0130	0.0011

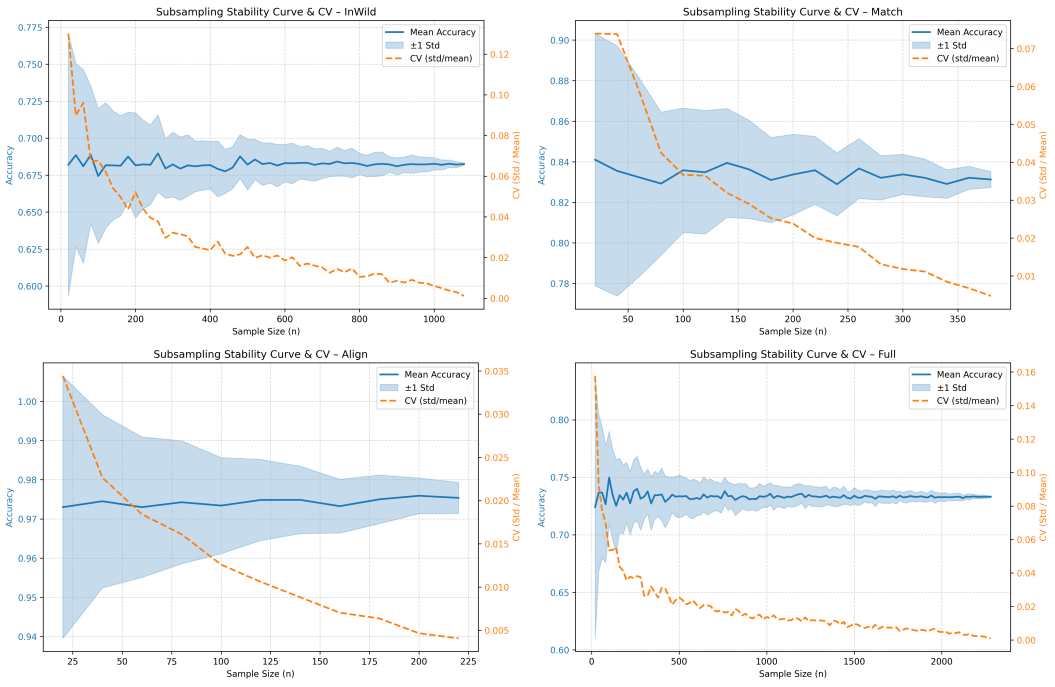


Figure 6: Trends of model accuracy metrics (mean, standard deviation, and coefficient of variation) with varying subsampling size on three subsets and the entire dataset.

[2] We empirically set the convergence threshold to $\tau = 0.02$, which requires the standard deviation of the evaluation score to be controlled within 2% of the mean. For a typical model accuracy range ($50\% \sim 80\%$), this means the measurement error is limited to an absolute range of approximately $1\% \sim 1.6\%$. This strict stability constraint is crucial for suppressing "ranking flips" caused by sampling variance, ensuring the benchmark can reliably distinguish between models with slight performance differences.

The experimental results (Table 9 and Figure 6) demonstrate that the current data scale of our dataset provides an ample safety margin for evaluation stability. Specifically, the actual sample size of each subset significantly exceeds the minimum number of samples required to reach the convergence threshold (approximately $1.42 \sim 4$ times the required minimum), and the lowest CV has dropped to the 10^{-3} magnitude. This outcome confirms that we have not only ensured high confidence in the evaluation results at the current scale but have also achieved a good balance between statistical robustness and evaluation efficiency (computational and time costs).

C.3 Assessment of Dataset Artifacts and Shortcut Learning

A substantial body of research warns against benchmark performance driven by spurious correlations or explicit features rather than genuine reasoning (Geirhos et al., 2020; Gururangan et al., 2018). To ensure our dataset evaluates robust time-series reasoning rather than relying on dataset artifacts, we analyzed the dependency of model performance on explicit surface-level attributes.

Specifically, we examined the correlation between TSQA accuracy and three explicit factors: sequence length (L), variable count (V), and question text length (T) on the InWild subset. We evaluated three representative models: GPT-4o, Qwen2.5-32B, and ChatTS(Xie et al., 2024). We introduce three metrics to quantify these dependencies: **(a) Correlation (r_L, r_T).** The Pearson correlation coefficient between accuracy and the logarithm of sequence length (r_L) or question text length (r_T). A value close to 0 indicates no linear dependency. **(b) Length Sensitivity (Δ_{long}).** The difference in mean accuracy between the samples in the longest quartile ($\geq$ 75th percentile) and the shortest quartile ($\leq$ 25th percentile). **(c) Dimensionality Gap (Δ_{dim}).** The difference in mean accuracy between multivariate and univariate samples.

As presented in Table 10, the results reveal minimal dependence on these artifacts. The correlation with sequence length is negligible across all models ($|r_L| < 0.08$), and the accuracy gap between extreme lengths (Δ_{long}) remains within a narrow range (approx. $0.05 \sim 0.08$), showing no consistent bias towards short or long sequences. Similarly, the performance gap between univariate and multivariate series is marginal ($|\Delta_{\text{dim}}| < 0.04$), and question length shows only a weak effect ($|r_T| \approx 0.15$). These findings confirm that our dataset is not trivially predictable by simple metadata features.

Table 10: Analysis of potential dataset artifacts and shortcut learning on the InWild subset. Small absolute values for all metrics indicate that model performance is not dominated by simple features like length or dimensionality.

Model Name	r_L	Δ_{long}	Δ_{dim}	r_T
GPT-4o	-0.0693	0.0824	-0.0101	-0.1451
Qwen2.5-32B	0.0547	-0.0525	0.0353	-0.0264
ChatTS	0.0727	-0.0502	-0.0163	-0.0862

C.4 Conclusion

We conducted three systematic analyses—Bootstrap Confidence Interval, Iterative Subsampling Analysis, and Assessment of Dataset Artifacts—which collectively demonstrate that our dataset possesses high statistical robustness, an efficient scale, and validity against shortcut learning. The results confirm that model performance on our dataset is driven by genuine time-series understanding rather than explicit surface-level features (e.g., sequence length or dimensionality), ensuring that the evaluation results are stable, reliable, and trustworthy.

D Ablation Study

Across all tasks in our benchmark, *ChatTS* demonstrates the best performance within the open-source TS-LLM category, showcasing robust time series analysis and reasoning capabilities. To further investigate the key factors that influence TS-LLM performance and provide insights for future research, we modified the official ChatTS training pipeline [3], adopting their released training data and recommended training strategy. We conducted controlled ablations on the **encoder architecture and size, positional encoding strategies, LLM backbone size, and prompt prefix design**.

D.1 Experimental Setup

In its original implementation, ChatTS employs Qwen2.5-14B-Instruct as the backbone LLM, with a 5-layer MLP serving as the time series encoder. During training, textual embeddings are aligned with time series embeddings to equip the model with time series reasoning capabilities. To examine the role of the encoder, we replaced the MLP with alternative architectures, including CNN and Transformer encoders with variable depth. We further tested the effect of introducing learnable positional embeddings or index-based positional features into the time series input.

Due to computational constraints, our experiments use Qwen2.5-3B-Instruct as the backbone, and we also report its text-only baseline performance on our benchmark. For comparison, we include performance of Qwen2.5-14B-Instruct, allowing us to isolate the effect of LLM backbone size. Finally, since ChatTS incorporates a prompt prefix that contains statistical information (e.g., offset, scale factor, length, max/min values, left/right boundary values), we tested models trained with and without this prefix to measure its contribution.

All models were evaluated on **InWild**, **Match**, and **Align**. While we closely followed the ChatTS training methodology, inevitable differences arise due to random training data mixing, limited compute budgets, and variations in model size and hyperparameters. Nonetheless, the relative comparisons across ablations yield consistent and reliable conclusions.

D.2 Evaluation Results

We categorize the factors related to the time series encoder into three dimensions: **(i) encoder architecture**, **(ii) encoder size**, and **(iii) positional encoding**. For the architecture study, we compared a 5-layer MLP (17.1M parameters), a CNN (50.4M), and a Transformer (6.3M) as the TS Encoders of our TS-LLMs. As shown in Table 11, the results indicate that model performance is largely insensitive to encoder architecture, with only marginal differences across tasks. Relative to the Qwen2.5-3B baseline, trained models exhibit no significant improvements on **InWild** and **Match**, but achieve clear gains on Sem→TS while degrading on TS→Sem. This suggests that the encoder introduces a directional bias in learning, which may be related to the distributional characteristics of the training data.

Table 11: Performance of TS-LLMs with different time series encoder architectures. We compare a 5-layer MLP, CNN, and Transformer as encoders, with their parameter sizes indicated in parentheses. The baseline is Qwen2.5-3B-Instruct, which treats the time series as plain text input.

Dataset	Baseline	MLP(17.1M)	CNN(50.4M)	Transformer(6.3M)
InWild	38.75	38.25	37.36	38.84
Match	27.45	30.67	28.42	30.00
Sem→TS	49.17	59.44	60.83	60.56
TS→Sem	64.17	45.56	44.44	47.50

To further examine scaling effects within a fixed architecture, we tested MLP encoders of varying depths (1, 3, 5, and 7 layers), as reported in Table 12. For reference, we also include the Qwen2.5-3B baseline and the original ChatTS(14B) checkpoint released on Hugging Face.[4] Results show that increasing the number of MLP layers does not yield a monotonic improvement, indicating

[3] https://github.com/xiezhe-24/ChatTS-Training

[4] https://huggingface.co/bytedance-research/ChatTS-14B

that simply enlarging the encoder does not directly translate into better performance. In contrast, comparing models with 3B and 14B backbones reveals consistent improvements of 10%–30% across tasks for the larger ones. This highlights the dominant role of the backbone's intrinsic reasoning capacity in determining TS-LLM performance. Besides, a comparison with the unaligned Qwen2.5 backbones yields results consistent with the above trend: while InWild and Match remain largely unchanged(except ChatTS in Match), aligned models achieve clear improvements on Sem→TS but show noticeable degradation on TS→Sem, thereby further validating our observation.

Table 12: Ablation study on the number of layers in time series encoders. We evaluate different layer counts for the MLP encoder, comparing performance with the baseline models Qwen2.5-3B-Instruct and Qwen2.5-14B-Instruct, and the ChatTS model. For the ChatTS model, we use the weights released by the original authors on Hugging Face, with Qwen2.5-14B-Instruct as the backbone and a 5-layer MLP as the time series encoder.

Dataset	Qwen2.5-3B	1 Layer(0.3M)	3 Layers(8.7M)	5 Layers(17.1M)	7 Layers(25.5M)	Qwen2.5-14B	ChatTS
InWild	38.75	38.90	38.44	38.25	37.45	52.84	50.28
Match	27.45	30.67	30.42	30.67	33.00	56.55	36.80
Sem→TS	49.17	60.83	51.67	59.44	61.39	86.83	91.17
TS→Sem	64.17	45.28	44.44	45.56	45.28	88.50	68.33

We also evaluated the effect of positional encoding strategies, following the three configurations in the official training code: no positional encoding, learnable embeddings appended to the input series, and normalized index values concatenated with the input series. Using a 3-layer MLP encoder, the results are reported in Table 13. These findings suggest that positional encoding design also has only a limited impact on performance compared with the backbone scale.

Table 13: Performance of models with different positional encoding strategies. `no_emb` denotes no positional encoding, `pos_emb` denotes learnable embeddings, and `pos_idx` denotes normalized index values used as positional encoding.

Dataset	no_emb	pos_emb	pos_idx
InWild	39.11	38.44	39.42
Match	31.00	30.42	28.67
Sem→TS	58.83	51.67	52.22
TS→Sem	42.22	44.44	45.00

Finally, we investigated the role of the prompt prefix introduced in ChatTS, which encodes statistical descriptors of the time series (e.g., offset, scale factor, length, min/max, and boundary values). We compared models trained with and without the prefix using a 5-layer MLP encoder, and additionally tested a model trained without the prefix but provided with the prefix at inference time. Results (Table 14) demonstrate that the statistical prompt prefix has a significant impact on model performance, especially in Align tasks, likely because it provides auxiliary information that enhances both interpretability and reasoning efficiency.

Table 14: Performance of models trained with and without the prompt prefix. **ON** indicates that the prefix is used during both training and testing; **OFF** indicates that it is used in neither; and **OFF*** denotes models trained without the prefix but evaluated with it.

Dataset	ON	OFF	OFF*
InWild	38.25	36.53	37.01
Match	30.67	25.50	33.41
Sem→TS	59.44	24.17	59.72
TS→Sem	45.56	21.39	45.83

D.3 Conclusion on Ablations

Our ablation findings can be summarized as follows:

- **Limited Encoder Contribution.** Under the current alignment paradigm, encoder architecture, scale, and positional encoding have only marginal effects. More effective paradigms for time series–text alignment remain an open challenge.
- **Backbone Dominance.** The LLM backbone size is the primary determinant of performance; scaling the backbone directly boosts temporal reasoning ability.
- **Prompt Engineering Effectiveness.** Incorporating statistical information into prompts substantially enhances model inference, suggesting that prompt engineering is a promising direction for strengthening TS-LLM reasoning. Future work should explore alternative prompt formats and auxiliary signals.

E STANDARDIZED TS INPUT FORMAT

We use the following prompts to standardize multiple-choice QA and numerical QA evaluation. The system prompt mandates the answer format and ambiguity policy; the user prompt injects per-item content. The model's output is scored by extracting the letter inside the `<final_answer>` tag.

E.1 SYSTEM PROMPT FOR MULTIPLE-CHOICE QA

```
You are an expert AI assistant specialized in answering multiple-choice
    questions with high accuracy and consistency. Your task is to analyze
     questions carefully and provide clear, well-reasoned answers.

IMPORTANT INSTRUCTIONS:
1. You must select your answer from the given options only
2. Your final answer must be a single letter (A, B, C, D, etc.)
3. If the question seems ambiguous, choose the most reasonable
    interpretation
4. Do not make up information not provided in the question

RESPONSE FORMAT:
You must structure your final answer exactly as follows:

<final_answer>
[State your chosen option as a single letter: A, B, C, or D]
</final_answer>

Remember: Your final answer should contain ONLY the letter of your chosen
     option, nothing else.
```

Listing 1: System Prompt for multiple-choice QA

E.2 SYSTEM PROMPT FOR NUMERICAL QA

```
You are an expert AI assistant specialized in answering numerical
    questions with high accuracy and consistency. Your task is to analyze
     questions carefully and provide precise numerical answers.

IMPORTANT INSTRUCTIONS:
1. You must provide a numerical answer (integer, decimal, or scientific
    notation)
2. Your final answer must be a single number only
3. If the question seems ambiguous, choose the most reasonable
    interpretation
4. Do not include units unless specifically requested in the question

RESPONSE FORMAT:
You must structure your final answer exactly as follows:

<final_answer>
[State your numerical answer as a single number only]
</final_answer>

Remember: Your final answer should contain ONLY the numerical value,
    nothing else.
```

Listing 2: System Prompt for numerical QA

E.2.1 User Prompt

```
Please answer the following multiple-choice/numerical question based on
    the given information:

Question: {question}

{given_values_str}

Available Options: {option}

Please analyze this question carefully, consider the given value and all
    available options, then provide your answer following the exact
    format specified in the system instructions.
```

Listing 3: User Prompt used for Evaluation

E.3 Time Series to Image Conversion

We follow the plotting style of Zhuang et al. Zhuang et al. (2024) and adapt it for multi-channel time series. Specifically, we preserve the single-channel resolution (1500×320 at 100 dpi) and scale the figure height linearly with the number of channels by stacking channel-wise subplots with a fixed per-channel height (320 px at 100 dpi). This keeps a consistent time axis across channels while maintaining comparable vertical resolution per channel.

Listing 4: Python code for converting time series data into images

```python
def plot_time_series_as_image(value_list):
    if len(value_list) > 8:  # single-channel time series
        num_channels = 1
    else:  # multi-channel time series
        num_channels = len(value_list)

    # Figure parameters: base width and per-channel height
    # Single channel: 1500x320; increase height by 320 for each
        additional channel
    width_inches = 15.0  # 1500 pixels / 100 dpi = 15 inches
    height_per_channel = 3.2  # 320 pixels / 100 dpi = 3.2 inches
    total_height = height_per_channel * num_channels

    plt.figure(figsize=(width_inches, total_height), dpi=100)

    if num_channels == 1:  # single-channel time series
        plt.plot(range(len(value_list)), value_list, 'b-', linewidth=1.5)
        plt.title('Time Series', fontsize=12)
        plt.xlabel('Time Index', fontsize=10)
        plt.ylabel('Value', fontsize=10)
        plt.grid(True, alpha=0.3)
        plt.xlim(0, len(value_list) - 1)
    else:  # multi-channel time series
        for i, channel_data in enumerate(value_list):
            plt.subplot(num_channels, 1, i + 1)
            plt.plot(range(len(channel_data)), channel_data, 'b-',
                linewidth=1.5)
            plt.title(f'Time Series {i+1}', fontsize=10)
            plt.xlabel('Time Index', fontsize=8)
            plt.ylabel('Value', fontsize=8)
            plt.grid(True, alpha=0.3)
            plt.xlim(0, len(channel_data) - 1)
    plt.tight_layout()
```