TimeMaster: Training Time-Series Multimodal LLMs to Reason via Reinforcement Learning

Junru Zhang[1,2*] **Lang Feng**[2*] **Xu Guo**[3†] **Yuhan Wu**[1] **Yabo Dong**[1†] **Duanqing Xu**[1]
[1]Zhejiang University [2]Nanyang Technological University [3]KTH Royal Institute of Technology
junruzhang@zju.edu.cn, lang005@e.ntu.edu.sg, xugu@kth.se,
{wuyuhan, dongyb, xdq}@zju.edu.cn

Abstract

Time-series reasoning remains a significant challenge for multimodal LLMs due to dynamic temporal patterns and semantic ambiguities, with existing models often lacking structured, human-aligned temporal understanding. In this work, we introduce TimeMaster, a novel reinforcement learning (RL)-based method that enables time-series MLLMs to perform structured, human-aligned reasoning over visualized temporal data. TimeMaster adopts a three-part output format (reasoning, classification, extension) and is optimized through a composite reward function within a two-stage pipeline (SFT followed by RL). Evaluated on TimerBed, TimeMaster achieves state-of-the-art performance, outperforming classical models by 8.3% and GPT-4o baselines by 7.3%, while also delivering human-aligned reasoning and actionable insights. This work offers a promising step towards equipping LLMs with robust temporal reasoning capabilities, paving the way for more interpretable and intelligent time-series analysis. Code is available at https://github.com/langfengQ/TimeMaster.

1 Introduction

Time series analysis is fundamental to data mining, enabling the modeling of temporal patterns and supporting decision-making across critical domains like healthcare [1, 2], industrial monitoring [3, 4], and environmental surveillance [5]. While deep learning has significantly advanced classical time-series tasks like forecasting [6, 7, 8, 9] and classification [7, 10, 11], these models primarily focus on numerical predictions. The recent surge in large language models (LLMs) [12, 13, 14, 15] presents a transformative opportunity for human-centric time-series analysis, promising capabilities beyond mere prediction to encompass genuine reasoning, explanation, and advice.

However, bridging the gap between LLMs and temporal data remains challenging. Standard text-based representations of time series often lead to inefficiencies and hallucination [16, 17]. While converting time series to visuals and leveraging multimodal LLMs (MLLMs) shows promise for pattern recognition, current methods primarily rely on prompt engineering, which often fails to elicit robust and coherent reasoning [18, 16, 19, 20, 21, 22]. A key limitation is the lack of pre-training on time-series visualizations in base MLLMs, and post-training approaches face hurdles like limited data diversity and high annotation costs. This hinders the development of MLLMs with deep, reliable, language-based *time-series reasoning* (TsR) capabilities [23], a crucial step for advanced applications.

In this work, we introduce TimeMaster, a novel reinforcement learning (RL) [24]-based framework that trains MLLMs for sophisticated multimodal time-series reasoning. TimeMaster directly addresses these limitations by learning through iterative RL, enabling the model to progressively acquire, refine, and generalize its reasoning capabilities. Our approach features a structured output format (reasoning, classification, extension) optimized by a composite reward function, balancing format

*Equal contribution.
†Corresponding authors.

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

adherence, accuracy, and the quality of generated insights. We employ a two-stage pipeline: initial SFT for foundational alignment and subsequent RL for targeted, self-improving reasoning. Our preliminary findings on the TimerBed benchmark [16] demonstrate TimeMaster's state-of-the-art performance, significantly outperforming classical models (by 8.3%) and GPT-4o baselines (by 7.3%), and exhibiting high-quality, context-aware reasoning and actionable extensions. This work pioneers a path towards equipping general-purpose LLMs with powerful temporal reasoning skills, opening new avenues for more intelligent and interpretable time-series analysis.

2 Related work

Language-free time-series models excel at learning temporal representations for forecasting and classification [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36], but they lack the linguistic capabilities essential for human-aligned reasoning. To address this, recent efforts integrate LLMs by transforming temporal data into modalities compatible with language models. One approach utilizes text sequences [37, 38, 39, 40, 21, 41, 42, 43, 44], while another focuses on visualizations [16, 19, 20, 22]. However, text-based approaches often struggle with prompt engineering for robust temporal reasoning [18] and face token inefficiencies [16, 17]. Visualization-based methods, while effective for pattern recognition, are fundamentally limited by the absence of explicit pre-training on time-series visualizations and high post-training data annotation costs. Concurrently, reinforcement learning (RL) [24] has emerged as a powerful paradigm, demonstrating efficacy in enhancing LLM reasoning and alignment [45, 46, 47, 48, 49, 50, 51]. This success suggests RL's significant potential to bridge the gap towards deep, reliable, and language-based temporal reasoning in LLMs, an aspiration our novel RL-based framework directly pursues to unlock advanced time-series reasoning capabilities.

3 Method

We introduce TimeMaster, an RL-enhanced framework that empowers general-purpose MLLMs with sophisticated, human-aligned time-series reasoning, enabling actionable insights beyond mere prediction through visualized inputs and a structured, reward-guided process.

3.1 Multimodal Time-Series Input and Structured Output

TimeMaster processes time series data $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^{T}$ (where $\mathbf{x}_t \in \mathbb{R}^D$) alongside textual context $\mathbf{q}$. Unlike traditional methods that focus on forecasting or classification, our approach transforms raw time series into visual representations (e.g., line plots) [16], enabling the MLLM's visual encoder to capture temporal patterns more effectively and efficiently [16, 37, 19].

The core of TimeMaster lies in its structured output format for Time-Series Reasoning (TsR), generating a three-part response:

$$\underbrace{\langle\texttt{think}\rangle \cdots \langle/\texttt{think}\rangle}_{\text{Reasoning}}\ \underbrace{\langle\texttt{class}\rangle \cdots \langle/\texttt{class}\rangle}_{\text{Classification}}\ \underbrace{\langle\texttt{extension}\rangle \cdots \langle/\texttt{extension}\rangle}_{\text{Extension (Optional)}}.$$

1) The `think` block articulates an open-ended reasoning process, detailing pattern recognition, trend analysis, and causal inference. **2**) The `class` block provides a discrete label for classification, enabling rigorous evaluation. **3**) The `extension` block offers optional follow-up insights, suggestions, diagnostics, or actionable advice, greatly enhancing practical usability. Figure 1 provides a visual overview of this process, illustrating how TimeMaster processes inputs and generates these structured outputs, guided by reward signals.

3.2 Reward Modeling for Temporal Reasoning

To effectively train TimeMaster, we designed a composite reward system that jointly optimizes structural correctness, classification accuracy, and extension quality. This system comprises: **1) Format Reward (r^{fmt}):** Enforces strict adherence to the XML-style output structure, penalizing missing tags. **2) Hard Reward (r^{hard}):** Evaluates classification accuracy: $r^{\text{hard}}\big(\hat{c}, c^{\star}\big) = \mathbb{I}[\hat{c} = c^{\star}]$, ensuring factual alignment. **3) Soft Reward (r^{soft}):** Assesses extension quality via an LLM-as-a-Judge [52] rating specificity, appropriateness, relevance, and depth, allowing for nuanced evaluation

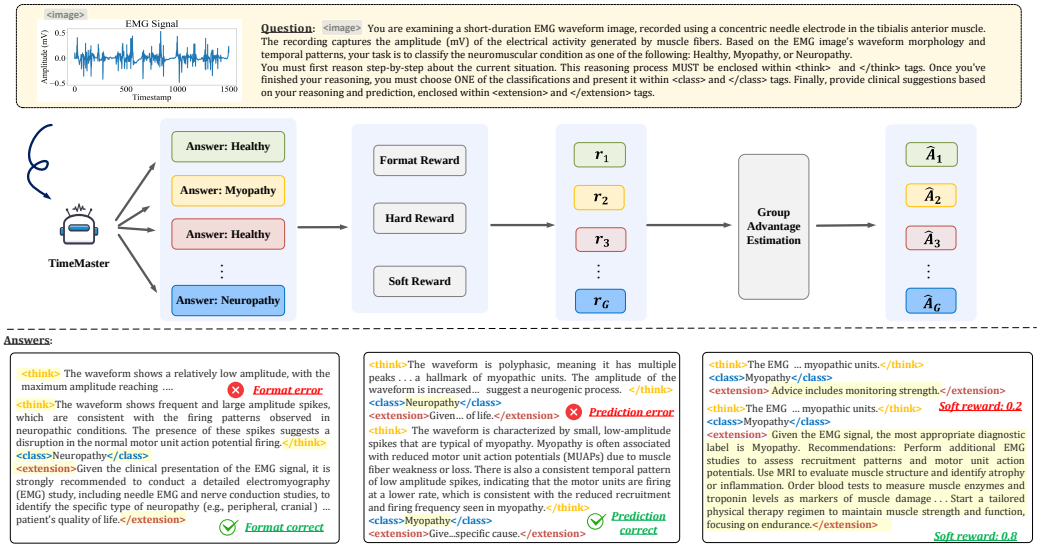

Figure 1: Overview of TimeMaster. The model is trained via RL with three reward signals. (Left) Format reward penalizes otherwise valid reasoning if required tags are missing. (Middle) Hard reward assigns zero if the Prediction is incorrect despite reasonable reasoning about myopathy features. (Right) Soft reward distinguishes between vague (e.g., "monitor strength") and high-quality (e.g., "recommend blood tests to measure muscle enzymes and troponin levels") clinical suggestions.

of actionable advice. These are combined into a unified composite reward:

$$r = \lambda^{\text{fmt}}\, r^{\text{fmt}} + \lambda^{\text{hard}}\, r^{\text{hard}} + \lambda^{\text{soft}}\, r^{\text{soft}}, \tag{1}$$

where λ values balance these objectives, guiding the model towards robust time-series analysis.

3.3 Optimization for TsR via RL

We employ a two-stage training pipeline to achieve expert-level temporal reasoning. **1) Stage 1: Supervised Fine-Tuning (SFT).** We initialize the model with foundational alignment via SFT using approximately 1,000 GPT-4o [18] generated examples, injecting domain knowledge and establishing the output format. **2) Stage 2: Reinforcement Learning (RL) with GRPO.** We then utilize token-level Group Relative Policy Optimization (GRPO) [50, 53] for RL training. This process enhances reasoning by maximizing a clipped surrogate objective with KL-divergence regularization. Normalized advantages are computed using group-wise statistics:

$$\hat{A}_i = \tfrac{r_i - \mu_r}{\sigma_r}, \quad \mu_r = \tfrac{1}{G}\textstyle\sum_{j=1}^{G} r_j, \quad \sigma_r = \sqrt{\tfrac{1}{G}\textstyle\sum_{j=1}^{G}(r_j - \mu_r)^2 + \varepsilon} \tag{2}$$

The objective function for parameter updates (θ) is:

$$\mathcal{L}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathbf{y}_i|}\sum_{k=1}^{|\mathbf{y}_i|}\min\big(\rho_{i,k}\hat{A}_i,\ \text{clip}(\rho_{i,k},\, 1\pm\epsilon)\,\hat{A}_i\big) - \beta\,\text{KL}\big[\pi_\theta \,\big\|\, \pi_{\text{ref}}\big] \tag{3}$$

This two-stage process enables TimeMaster to progressively refine its temporal reasoning capabilities, leading to more accurate and interpretable insights.

4 Experiment

Training Pipeline. We employ a two-stage training pipeline for TimeMaster. We first fine-tune our base MLLM (Qwen2.5-VL-3B-Instruct [62]) on approximately 1,000 task-specific samples per dataset, generated by GPT-4o [18], for initial instruction tuning and foundational alignment. We then proceed with RL fine-tuning. The training configuration is provided in Appendix D, where reward weights $(\lambda^{\text{fmt}}, \lambda^{\text{hard}}, \lambda^{\text{soft}}) = (0.1, 0.9, 1.0)$. We evaluate the base Qwen2.5-VL, its SFT-tuned variant, TimeMaster (RL), and the complete TimeMaster (SFT+RL) model.

Table 1: Accuracy (%) of different methods on TimerBed. "Simple Determ." denotes simple deterministic reasoning and "Complex Determ." denotes complex deterministic reasoning.

Modality	Type	Method	Reasoning	Simple Determ.		Complex Determ.		Probabilistic		Avg.
				RCW	TEE	ECG	EMG	CTU	HAR	
Numeric	Classical	MLP [54]	✗	$65.95_{\pm 2.14}$	$61.64_{\pm 0.15}$	$57.97_{\pm 0.54}$	$58.54_{\pm 0.78}$	$56.80_{\pm 1.77}$	$62.00_{\pm 0.25}$	$60.48_{\pm 1.11}$
		FCN [54]	✗	$65.80_{\pm 0.01}$	$50.68_{\pm 0.01}$	$61.52_{\pm 0.04}$	$\mathbf{100.00}_{\pm 0.05}$	$63.07_{\pm 0.01}$	$67.73_{\pm 0.02}$	$68.13_{\pm 0.02}$
		ResNet [54]	✗	$70.44_{\pm 0.04}$	$52.05_{\pm 0.01}$	$\mathbf{64.36}_{\pm 0.03}$	$\mathbf{100.00}_{\pm 0.01}$	$66.00_{\pm 0.04}$	$64.37_{\pm 0.05}$	$69.54_{\pm 0.02}$
		Transformer [55]	✗	$65.82_{\pm 0.85}$	$59.52_{\pm 1.27}$	$25.00_{\pm 2.15}$	$86.67_{\pm 1.78}$	$58.70_{\pm 1.24}$	$87.26_{\pm 0.78}$	$63.83_{\pm 1.43}$
		Autoformer [56]	✗	$62.59_{\pm 3.52}$	$26.19_{\pm 3.77}$	$23.95_{\pm 2.78}$	$46.67_{\pm 1.44}$	$67.20_{\pm 1.70}$	$75.04_{\pm 1.20}$	$50.27_{\pm 2.61}$
		Informer [57]	✗	$75.51_{\pm 2.75}$	$59.52_{\pm 1.47}$	$22.39_{\pm 3.41}$	$66.66_{\pm 1.20}$	$67.20_{\pm 0.65}$	$85.83_{\pm 1.17}$	$62.85_{\pm 2.02}$
		FEDformer [58]	✗	$76.59_{\pm 0.75}$	$42.86_{\pm 1.21}$	$26.40_{\pm 3.44}$	$73.33_{\pm 0.74}$	$51.60_{\pm 2.18}$	$\mathbf{89.88}_{\pm 0.44}$	$60.11_{\pm 1.80}$
		PatchTST [59]	✗	$82.11_{\pm 0.12}$	$57.14_{\pm 2.10}$	$24.82_{\pm 2.44}$	$60.00_{\pm 1.80}$	$64.00_{\pm 0.75}$	$79.60_{\pm 1.40}$	$61.28_{\pm 1.64}$
		iTransformer [60]	✗	$76.92_{\pm 0.54}$	$21.43_{\pm 1.20}$	$25.72_{\pm 0.87}$	$46.67_{\pm 1.46}$	$45.57_{\pm 1.22}$	$89.49_{\pm 1.71}$	$50.97_{\pm 1.23}$
		TimesNet [8]	✗	$80.23_{\pm 0.88}$	$61.90_{\pm 1.40}$	$26.20_{\pm 3.41}$	$73.33_{\pm 0.25}$	$64.00_{\pm 0.87}$	$88.65_{\pm 0.14}$	$65.72_{\pm 1.59}$
		DLinear [61]	✗	$56.96_{\pm 0.45}$	$47.63_{\pm 1.20}$	$24.67_{\pm 1.50}$	$46.67_{\pm 0.71}$	$52.40_{\pm 0.12}$	$47.79_{\pm 1.47}$	$46.02_{\pm 1.05}$
Base: GPT-4o										
Numeric+Text	Prompting	GPT-4o (Zero-shot) [16]	✓	$50.00_{\pm 0.00}$	$21.43_{\pm 6.50}$	$25.00_{\pm 8.25}$	$33.33_{\pm 5.25}$	$45.45_{\pm 9.09}$	$29.17_{\pm 8.25}$	$34.06_{\pm 6.22}$
Image+Text	Prompting	VL-Time (Zero-shot) [16]	✓	$70.02_{\pm 2.15}$	$24.88_{\pm 1.47}$	$26.33_{\pm 2.64}$	$33.33_{\pm 6.25}$	$50.71_{\pm 5.25}$	$37.50_{\pm 2.15}$	$40.46_{\pm 3.32}$
Numeric+Text	Prompting	GPT-4o (Few-shot) [16]	✓	$50.00_{\pm 0.00}$	$35.71_{\pm 1.21}$	$31.25_{\pm 2.50}$	$33.33_{\pm 6.25}$	$50.00_{\pm 2.25}$	$12.50_{\pm 0.05}$	$35.47_{\pm 2.04}$
Image+Text	Prompting	VL-Time (Few-shot) [16]	✓	$\mathbf{91.03}_{\pm 0.25}$	$64.29_{\pm 8.25}$	$43.75_{\pm 5.25}$	$91.67_{\pm 0.85}$	$63.64_{\pm 1.20}$	$66.67_{\pm 2.50}$	$70.18_{\pm 3.05}$
Base: Qwen2.5-7B-Instruct										
Numeric+Text	Training	Time-MQA [44]	✓	$36.84_{\pm 4.09}$	$10.48_{\pm 6.25}$	$25.00_{\pm 2.65}$	$18.94_{\pm 1.85}$	$38.40_{\pm 2.74}$	$16.83_{\pm 2.70}$	$24.42_{\pm 3.38}$
Base: Qwen2.5-VL-3B-Instruct										
Image+Text	Prompting	Qwen2.5-VL	✓	$47.66_{\pm 2.41}$	$13.70_{\pm 0.00}$	$20.00_{\pm 0.00}$	$17.03_{\pm 1.25}$	$46.40_{\pm 2.75}$	$16.49_{\pm 1.68}$	$26.88_{\pm 1.35}$
Image+Text	Training	Qwen2.5-VL (SFT)	✓	$49.29_{\pm 1.27}$	$19.18_{\pm 0.14}$	$21.92_{\pm 0.24}$	$34.15_{\pm 0.15}$	$50.00_{\pm 0.00}$	$21.95_{\pm 0.04}$	$32.75_{\pm 0.53}$
Image+Text	Training	**TimeMaster (RL)**	✓	$72.53_{\pm 0.75}$	$13.70_{\pm 0.00}$	$25.00_{\pm 0.00}$	$48.78_{\pm 1.20}$	$54.00_{\pm 2.50}$	$34.55_{\pm 1.80}$	$41.43_{\pm 1.38}$
Image+Text	Training	**TimeMaster (SFT+RL)**	✓	$75.56_{\pm 1.30}$	$\mathbf{68.49}_{\pm 2.09}$	$60.00_{\pm 0.77}$	$\mathbf{100.00}_{\pm 1.41}$	$\mathbf{84.40}_{\pm 0.40}$	$63.29_{\pm 0.74}$	$\mathbf{75.29}_{\pm 1.25}$

Evaluation Benchmark and Baselines. We evaluate TimeMaster on TimerBed [16] (Appendix B), a benchmark specifically designed for time-series reasoning that moves beyond simple accuracy by requiring models to *reason* and *explain* predictions. TimerBed features six real-world classification datasets categorized by reasoning complexity: simple deterministic (**RCW** for whale calls, **TEE** for electromagnetic events), complex deterministic (**ECG** for arrhythmias, **EMG** for muscle disorders), and probabilistic (**HAR** for physical activities, **CTU** for device usage).

Our comparisons include a comprehensive set of baselines (Appendix C), categorized as follows: **1) Classical Time-Series Models:** We evaluate 11 established models with diverse architectures (e.g., MLP [54], ResNet [54], Autoformer [56], TimesNet [8]). These models excel at capturing temporal patterns but lack explicit language reasoning capabilities. **2) LLM-based TsR Methods:** We assess GPT-4o with both numeric inputs and visualized time-series plots (VL-Time [16]), evaluated in zero-/few-shot settings. We also include Time-MQA [44], an LLM fine-tuned on 200k time-series question answering examples, to gauge its reasoning abilities.

5 Results & Discussion

Our evaluation on the TimerBed benchmark [16] showcases TimeMaster's superior performance in time-series reasoning. As detailed in Table 1, TimeMaster achieves state-of-the-art accuracy (75.29% average), significantly outperforming classical time-series models (by 8.3%) and prompting-based GPT-4o baselines (by 7.3%). This robust performance, achieved with a 3B-parameter model, highlights TimeMaster's remarkable parameter efficiency and the effectiveness of RL training in surpassing limitations of prompt engineering and supervised fine-tuning alone. Beyond classification, TimeMaster excels at generating human-aligned reasoning and actionable insights through its structured, integrated multi-scale signal patterns (e.g., waveform morphology, amplitude, and rhythm consistency) and context-aware extensions (Fig. 2), offering practical utility for decision-making. This demonstrates a viable path for empowering general-purpose MLLMs with enhanced temporal understanding. While our approach successfully transfers general LLM capabilities to the temporal domain, a potential limitation is the non-triviality of encoding complex multivariate data like HAR into visual representations. Future work will explore hybrid architectures and the incorporation of external knowledge to further improve performance and reasoning depth. Overall, TimeMaster marks a significant step toward structured, interpretable time-series reasoning in LLMs, opening new avenues for more intelligent and context-aware applications in critical domains.

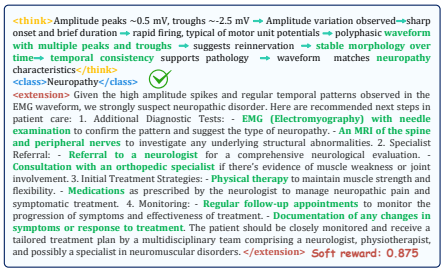

Figure 2: Output example: structured reasoning, classification, and clinical suggestion for the neuropathic EMG signal.

Acknowledgments

Xu Guo thanks the support from the Wallenberg-NTU Presidential Postdoctoral Fellowship.

References

[1] Junru Zhang, Cheng Peng, Zhidan Liu, Lang Feng, Yuhan Wu, Yabo Dong, and Duanqing Xu. Di2sdiff++: Activity style decomposition and diffusion-based fusion for cross-person generalization in activity recognition. *IEEE Transactions on Mobile Computing*, 2025.

[2] Junru Zhang, Lang Feng, Zhidan Liu, Yuhan Wu, Yang He, Yabo Dong, and Duanqing Xu. Diverse intra-and inter-domain activity style fusion for cross-person generalization in activity recognition. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4213–4222, 2024.

[3] Yucheng Wang, Yuecong Xu, Jianfei Yang, Zhenghua Chen, Min Wu, Xiaoli Li, and Lihua Xie. Sensor alignment for multivariate time-series unsupervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 10253–10261, 2023.

[4] Mohamed Ragab, Zhenghua Chen, Wenyu Zhang, Emadeldeen Eldele, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. Conditional contrastive domain generalization for fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

[5] Junru Zhang, Lang Feng, Yang He, Yuhan Wu, and Yabo Dong. Temporal convolutional explorer helps understand 1d-cnn's learning behavior in time series classification from frequency domain. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3351–3360, 2023.

[6] Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. Tactis: Transformer-attentional copulas for time series. In *International Conference on Machine Learning*, pages 5447–5493. PMLR, 2022.

[7] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. Tslanet: Rethinking transformers for time series representation learning. In *International Conference on Machine Learning*, pages 12409–12428. PMLR, 2024.

[8] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[9] Yuhan Wu, Xiyu Meng, Huajin Hu, Junru Zhang, Yabo Dong, and Dongming Lu. Affirm: Interactive mamba with adaptive fourier filters for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21599–21607, 2025.

[10] Junru Zhang, Lang Feng, Haowen Zhang, Yuhan Wu, and Yabo Dong. Adacket: Adaptive convolutional kernel transform for multivariate time series classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 189–204. Springer, 2023.

[11] Junru Zhang, Lang Feng, Xu Guo, Han Yu, Yabo Dong, and Duanqing Xu. Diffusion-guided diversity for single domain generalization in time series classification. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3764–3773, 2025.

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

[13] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[14] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[15] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[16] Haoxin Liu, Chenghao Liu, and B Aditya Prakash. A picture is worth a thousand numbers: Enabling llms reason about time series via visualization. *arXiv preprint arXiv:2411.06018*, 2024.

[17] Xiongxiao Xu, Yue Zhao, S Yu Philip, and Kai Shu. Beyond numbers: A survey of time series analysis in the era of multimodal llms. *Authorea Preprints*, 2025.

[18] OpenAI. Gpt-4o, 2024. Accessed: 2025-04-21.

[19] Xiongxiao Xu, Haoran Wang, Yueqing Liang, Philip S Yu, Yue Zhao, and Kai Shu. Can multimodal llms perform time series anomaly detection? *arXiv preprint arXiv:2502.17812*, 2025.

[20] 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.

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

[22] Zihao Zhou and Rose Yu. Can llms understand time series anomalies? *arXiv preprint arXiv:2410.05440*, 2024.

[23] Yaxuan Kong, Yiyuan Yang, Shiyu Wang, Chenghao Liu, Yuxuan Liang, Ming Jin, Stefan Zohren, Dan Pei, Yan Liu, and Qingsong Wen. Position: Empowering time series reasoning with multimodal llms. *arXiv preprint arXiv:2502.01477*, 2025.

[24] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[25] Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

[26] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[27] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

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

[29] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

[30] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

[31] Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. In *The Twelfth International Conference on Learning Representations*.

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

[33] Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18915–18923, 2025.

[34] Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. *arXiv preprint arXiv:2502.00816*, 2025.

[35] Haoran Zhang, Yong Liu, Yunzhong Qiu, Haixuan Liu, Zhongyi Pei, Jianmin Wang, and Mingsheng Long. Timesbert: A bert-style foundation model for time series understanding. *arXiv preprint arXiv:2502.21245*, 2025.

[36] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024.

[37] 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.

[38] Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. Explainable multi-modal time series prediction with llm-in-the-loop. *arXiv preprint arXiv:2503.01013*, 2025.

[39] Haoxin Liu, Zhiyuan Zhao, Shiduo Li, and B Aditya Prakash. Evaluating system 1 vs. 2 reasoning approaches for zero-shot time-series forecasting: A benchmark and insights. *arXiv preprint arXiv:2503.01895*, 2025.

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

[41] Mingtian Tan, Mike A Merrill, Zack Gottesman, Tim Althoff, David Evans, and Tom Hartvigsen. Inferring events from time series using language models. *arXiv preprint arXiv:2503.14190*, 2025.

[42] Zijia Liu, Peixuan Han, Haofei Yu, Haoru Li, and Jiaxuan You. Time-r1: Towards comprehensive temporal reasoning in llms. *arXiv preprint arXiv:2505.13508*, 2025.

[43] 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.

[44] 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.

[45] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[46] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

[47] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[48] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[49] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[50] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[51] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[52] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[53] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[54] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 35(2):401–449, 2021.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

[58] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.

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

[60] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[61] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

[62] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[63] Haoxin Liu, Harshavardhan Kamarthi, Zhiyuan Zhao, Shangqing Xu, Shiyu Wang, Qingsong Wen, Tom Hartvigsen, Fei Wang, and B Aditya Prakash. How can time series analysis benefit from multiple modalities? a survey and outlook. *arXiv preprint arXiv:2503.11835*, 2025.

A Pseudo Code

The training pipeline of TimeMaster is provided in Alg. 1.

Algorithm 1 Training pipeline of TimeMaster

Require: Initial time-series MLLM π_θ, judge J, dataset $\mathcal{D}$, group size G, PPO clip ϵ, KL weight β
Supervised fine-tune π_θ on cold-start data with structured outputs
for each RL iteration **do**
  Update the reference model: $\pi_{\text{ref}} \leftarrow \pi_\theta$
  for Step $= 1, 2, \ldots$ **do**
    Sample a mini-batch $\mathcal{B}$ from $\mathcal{D}$
    Update the old model: $\pi_{\text{old}} \leftarrow \pi_\theta$
    Sample G outputs $\{\mathbf{y}_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{X}, \mathbf{q})$ for each time-series instance $(\mathbf{X}, \mathbf{q}) \in \mathcal{B}$
    for each sampled $\mathbf{y}_i$ **do**
      Parse tags: `<think>`$_i$, `<class>`$_i = \hat{c}_i$, `<extension>`$_i = e_i$
      Compute format reward: $r_i^{\text{fmt}} = \mathbb{I}[\text{tags well-formed and non-empty}]$
      Compute hard reward: $r_i^{\text{hard}} = \mathbb{I}[\hat{c}_i = c^\star]$
      Compute soft reward: $r_i^{\text{soft}} = r_i^{\text{hard}} \cdot \text{mean}\{\phi(e_i, c^\star)\}$
      Compute composite reward: $r_i = \lambda^{\text{fmt}} r_i^{\text{fmt}} + \lambda^{\text{hard}} r_i^{\text{hard}} + \lambda^{\text{soft}} r_i^{\text{soft}}$
    end for
    Compute $\{\hat{A}_i\}_{i=1}^{G}$ for each group via Eq. (2)
    Update π_θ by maximizing $\mathcal{L}(\theta)$ via Eq. (3)
  end for
end for
return π_θ

B Dataset Statistics

In this section, we provide additional details on the real-world time-series classification datasets used in our experiments, drawn from **TimerBed** [16]. These datasets span a wide range of domains, signal characteristics, and reasoning complexities. Their key statistics are summarized in Table 2, while detailed descriptions are provided below.

- **Right Whale Call Detection (RCW)**[3] involves identifying North Atlantic right whale vocalizations from underwater acoustic recordings. A distinctive short, rising "whoop" sound serves as a necessary and sufficient indicator of whale presence, enabling direct mapping between signal features and class labels.
- **Transient Electromagnetic Events (TEE)**[4] contains satellite-collected power density signals from the FORTE satellite, used to classify various types of lightning-related electromagnetic discharges. Each class corresponds to a well-defined physical signature in the waveform, making the task pattern-centric and signal-driven.
- **Electrocardiogram (ECG)**[5] includes single-lead ECG recordings used to diagnose cardiac arrhythmias. Accurate classification requires a holistic interpretation of multiple waveform components—such as P-wave absence, irregular R-R intervals, and atrial fibrillation indicators.
- **Electromyogram (EMG)**[6] comprises EMG signals used to distinguish healthy subjects from patients with neuropathic or myopathic disorders. The task involves reasoning over diverse waveform features including long-duration, high-amplitude motor unit potentials and polyphasic activity.

[3] `https://www.kaggle.com/competitions/whale-detection-challenge/data`
[4] `https://www.timeseriesclassification.com/description.php?Dataset=Lightning7`
[5] `https://physionet.org/content/challenge-2017/1.0.0/`
[6] `https://physionet.org/content/emgdb/1.0.0/`

- **Human Activity Recognition (HAR)**[7] uses tri-axial accelerometer data from smartphones to classify six daily physical activities such as walking, standing, and lying. Labels are automatically generated and reflect latent user-dependent patterns, adding complexity to the modeling process.
- **Computer Type Usage (CTU)**[8] aims to differentiate between desktop and laptop usage based on 24-hour electricity consumption traces. Although labels are programmatically derived, classification relies on subtle temporal usage signatures influenced by individual behavior.

Table 2: Summary of datasets in TimerBed, including domain, number of variables, series length, number of classes, sample size, and reasoning type.

Dataset	Domain	# Variables	Length	# Classes	# Samples	Reasoning Type
RCW	Bioacoustics	1	4000	2	30,000	Simple Deterministic
TEE	Geophysics	1	319	7	143	Simple Deterministic
ECG	Healthcare	1	1500	4	43,673	Complex Deterministic
EMG	Healthcare	1	1500	3	205	Complex Deterministic
HAR	Sports Monitoring	3	128	6	10,299	Probabilistic
CTU	Energy/Usage	1	720	2	500	Probabilistic

As shown in Table 2, each dataset in TimerBed is associated with one of three reasoning types, reflecting varying levels of complexity and semantic abstraction:

- **Simple deterministic reasoning**: Tasks where a single salient feature is sufficient to determine the label. The decision boundary is often explicit and rule-based, allowing for direct mapping from input to output. For example, the presence of a distinct acoustic pattern in whale calls or a spike in satellite signal indicates class membership.
- **Complex deterministic reasoning**: Tasks that require the integration of multiple temporal patterns or signal components to make a decision. These problems demand holistic reasoning over structured signal relationships, such as diagnosing arrhythmias by jointly considering P-wave morphology, heart rate regularity, and waveform intervals.
- **Probabilistic reasoning**: Tasks characterized by user-specific or hidden variables, where labels are automatically derived and may not be directly observable in the input. As a result, the model must learn to infer outcomes under ambiguity and latent context, such as predicting user activity or device type based on behavior-driven time series.

C Baselines

We follow the recent time-series reasoning benchmark [16] and the survey [63] for the selection of the following baselines. Except for Time-MQA, other results in Table 1 are adopted from [16].

- **Fully-connected and CNN-based Models** (MLP, FCN, ResNet [54]): We adopt MLP with ReLU and dropout, FCN with Conv-BN-ReLU and pooling, and ResNet with residual connections, three classical architectures widely used in time series classification.
- **Transformer-based Models** (Transformer [55], Autoformer [56], Informer [57], FEDformer [58], PatchTST [59], iTransformer [60]): Capture long-range dependencies in time-domain sequences using self-attention mechanisms. Serve as strong baselines for time-series modeling.
- **CNN-based Models** (TimesNet [8]): Leverage convolutional operations to extract temporal features across different time scales.
- **MLP-based Models** (DLinear [61]): Employ lightweight feedforward layers for efficient modeling of local patterns in time-series data.
- **GPT-4o (Numeric, Zero-shot)** [16]: Receives tokenized numerical time-series data as input without any demonstrations. Serves as a unimodal, language-only baseline for evaluating zero-shot generalization.

[7] `https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones`

[8] `https://www.timeseriesclassification.com/description.php?Dataset=Computers`

- **GPT-4o (Numeric, Few-shot)** [16]: Extends the numeric input with a few in-context examples[9] per class. Evaluates the model's few-shot reasoning capability using raw numerical sequences in standard prompt format.
- **VL-Time (Zero-shot)** [16]: Provides multimodal LLMs with visualized time-series plots and natural language prompts. Assesses general reasoning ability without demonstrations.
- **VL-Time (Few-shot)** [16]: Adds a few in-context examples[10] to the visual-language input, enabling pattern generalization with minimal supervision.
- **Qwen2.5-7B-Instruct (Time-MQA)** [44]: A fine-tuned Qwen2.5-7B model[11] on the TSQA dataset ($\sim$200k pairs) for time-series question answering. It enables multi-task reasoning and open-ended question answering via natural language prompts.

D Implementation Details of TimeMaster

D.1 Plotting Time Series as Images

Visualizing time series as images offers an intuitive and cost-efficient approach to understanding temporal patterns, and has been widely adopted in recent studies [17, 63]. Following the methodology of VL-Time [16], we transform time-series data into RGB line plots in the time domain to serve as inputs for the vision-language model, ensuring a fair comparison. Each channel is rendered in a distinct color and aligned along a shared timestamp axis. The **x**-axis represents the *Timestamp*, while the **y**-axis denotes the corresponding signal *Value*. Legends are incorporated to distinguish between channels (e.g., `body_acc_x`, `body_acc_y`, `body_acc_z` in HAR datasets).

For each dataset, the signals are rendered into images with resolutions adapted to their sequence lengths and signal characteristics: *ECG* samples are plotted at 980×230 pixels, *CTU* at 562×230, *TEE*, *RCW*, and *EMG* at 789×239, and *HAR* at 389×233. All plots are saved in `PNG` format with minimal padding and a tight layout to ensure visual clarity. The time-domain signals are plotted using raw (non-normalized) values to faithfully preserve their original temporal dynamics.

D.2 Training Setup

We initialize our backbone with the publicly available Qwen2.5-VL-3B-Instruct checkpoint [62][12]. Our overall training pipeline comprises two stages: warm-up through supervised fine-tuning (SFT) and reinforcement learning with GRPO.

Warm-up via Supervised Fine-tuning. We first sample $\sim$ 1,000 time series–text paired instances per dataset using `GPT-4o` (temperature = 1.0) via the OpenAI API, where the model is prompted to reason over each time series and generate a corresponding answer. These examples are used to warm-start the SFT model, which is adapted from a publicly available LLM training repository[13]. The corresponding training hyperparameters are summarized in Table 3.

Reinforcement Learning with GRPO. After warm-up, we train the model using the GRPO algorithm, adapted from a public RL training library[14]. The complete GRPO configuration is provided in Table 4. Rewards are computed using Eq. 1, with coefficients $(\lambda^{\text{fmt}}, \lambda^{\text{hard}}, \lambda^{\text{soft}}) = (0.1, 0.9, 0)$. In case studies assessing extrapolation, we set $\lambda^{\text{soft}} = 1.0$.

D.3 System Configuration

All experiments were conducted on a computing setup equipped with 4 NVIDIA A100-SXM4 GPUs (80GB each) for the RCW, HAR, and ECG datasets, and 4 NVIDIA RTX A6000 GPUs (48GB each) for the TEE, EMG, and CTU datasets.

[9] Few-shot refers to fewer than six examples per class, following [16]

[10] Few-shot refers to fewer than six examples per class, following [16]

[11] `https://huggingface.co/Time-MQA`

[12] `https://huggingface.co/Qwen/Qwen2.5-3B-Instruct`

[13] `https://github.com/2U1/Qwen2-VL-Finetune`

[14] `https://github.com/volcengine/verl`

Table 3: Training configuration for supervised fine-tuning (SFT) using Qwen2.5-VL-3B-Instruct.

Parameter	**Value**
Model	Qwen/Qwen2.5-VL-3B-Instruct
Training mode	Full fine-tuning (LLM + Vision + Merger)
Use Liger	True
Batch size per device	4
Number of devices	4
Global batch size	128
Gradient accumulation steps	8
Epochs	2
Learning rate (LLM)	1e-5
Learning rate (Merger)	1e-5
Learning rate (Vision)	2e-6
Weight decay	0.1
Warmup ratio	0.03
LR scheduler	Cosine
Precision	bf16
Freeze vision tower	False
Freeze LLM	False
Tune merger	True

Table 4: GRPO-related hyperparameters used in TimeMaster across different time-series tasks.

Parameter	**Value**
π_θ^{init}	Qwen2.5-VL-3B
$L_{\max}$ (max sequence length)	2048
G (group size)	5
β (KL divergence coefficient)	0.001
ϵ (PPO clip threshold)	0.2
$(\lambda^{\text{fmt}}, \lambda^{\text{hard}}, \lambda^{\text{soft}})$	(0.1, 0.9, 0) or (0.1, 0.9, 1)
Batch size	16 (TEE, EMG, CTU); 32 (RCW, ECG, HAR)
Learning rate (RL)	1×10^{-6}
RL training epochs	40 (RCW, ECG, HAR); 100 (EMG, CTU); 300 (TEE)

E More Experimental Results

E.1 Impact of Input Modality

To explore the effect of input modality, we compare TimeMaster using visual inputs (line plots) with a numeric-input variant that processes raw tokenized values in Fig. 3. Both models share the same Qwen2.5-3B architecture and are trained under an identical two-stage pipeline (SFT + RL), differing only in input format. (**1**) *Visual inputs enable more robust reasoning*: As shown, visual inputs consistently yield higher accuracy than numeric inputs. This is because numeric inputs impose a heavy symbolic burden on the model, often leading to hallucinations and fragmented reasoning that undermine performance. In contrast, visual representations capture global temporal structures, such as trends, peaks, and rhythmic patterns, that closely mirror the diagnostic strategies employed by human experts in fields such as ECG and EMG analysis. (**2**) *Visual inputs offer superior token efficiency*: Numeric sequences scale linearly with length and often consume 5× more tokens than visual representations. For example, a 4k-point sequence can yield over 40k tokens, more than 88×

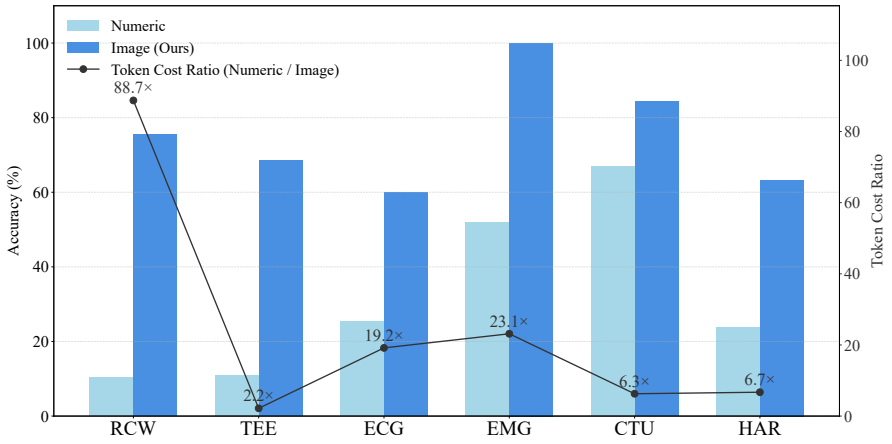

Figure 3: Accuracy and Token Cost: Image vs Numeric Inputs. Accuracy and relative token cost of TimeMaster (SFT+RL) using image-based and numeric inputs across six tasks.

its visual counterpart, whereas our visual inputs maintain fixed size regardless of sequence length, greatly improving scalability. **(3)** *Training with numeric inputs leads to instability*: models trained on raw values frequently exhibit semantic drift, hallucinated reasoning, and eventual collapse of output structure. Due to persistent degradation, we halted training early and report the best validation result prior to termination here.

As shown in Figure 4, we frequently observe training collapse with numeric inputs due to hallucination. For instance, on the TEE dataset, the model experienced a sudden degradation after only 45 RL steps, classification accuracy dropped to 0, and all subsequent outputs collapsed into repetitive or nonsensical tokens (e.g., a sequence of '!!!'), as illustrated in Figure 5. This form of semantic collapse not only disrupts reasoning chains but also renders the model unusable beyond that point.

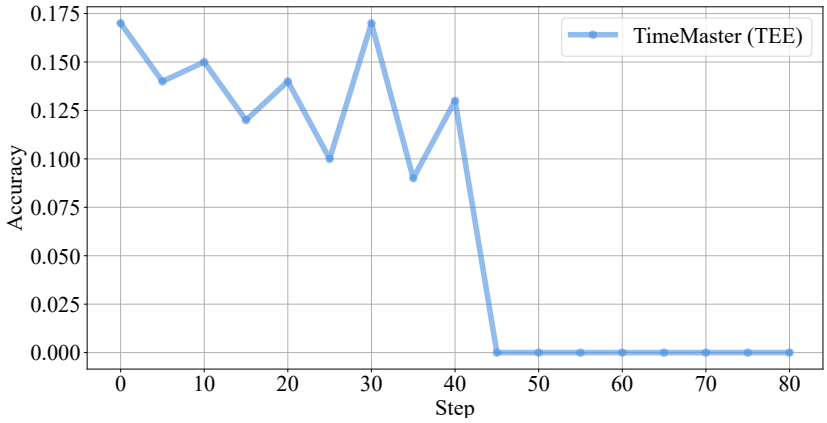

Figure 4: Training Collapse on TEE (Numeric Input).

E.2 Case Studies of Temporal Reasoning

Figure 6 shows TimeMaster (RL+SFT)'s complete reasoning compared to baselines. This highlights Qwen2.5-VL (SFT)'s shallow reasoning, often misclassifying complex cases (e.g., "other cardiac rhythms" as "atrial fibrillation," neuropathic EMG as "healthy") due to reliance on superficial cues. In contrast, TimeMaster (RL) demonstrates contextual awareness by integrating uncertainty and noise (e.g., "f waves... doesn't match atrial fibrillation"). The full TimeMaster (SFT+RL) achieves human-like interpretations (e.g., "polyphasic morphology... consistent with neuropathy") by leveraging multiple features. This illustrates RL's power, amplified by SFT, in refining reasoning and overcoming supervised limitations for robust temporal understanding.

Failure Case on TEE Input Prompt and Model Output (Step 45)

Model Input:
You are analyzing a time series signal derived from lightning-related electromagnetic events, recorded by the FORTE satellite. Your task is to classify the signal into one of the following 7 event types: CG Positive, IR Negative, Subsequent Return Stroke, Impulsive, Impulsive Pair, Gradual Intra-Cloud, or Off-record.
You are required to identify and report the approximate value ranges (minimum and maximum) of the signals over the time period. Please choose the best matching label for the full signal.
Here are the data points: `-0.610, -0.610, -0.610, ..., -0.399, -0.495, -0.597, -0.610, -0.610`
You MUST first perform step-by-step reasoning for your prediction. This reasoning MUST be enclosed within `<think>` and `</think>` tags. After completing your reasoning, you must select ONE most appropriate label and enclose it within `<class>` and `</class>` tags.

Model Output:
!!!