THOTH: MID-TRAINING BRIDGES LLMS TO TIME SERIES UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have demonstrated remarkable success in general-purpose reasoning. However, they still struggle to understand and reason about time series data, which limits their effectiveness in decision-making scenarios that depend on temporal dynamics. In this paper, we propose **Thoth**, the first family of mid-trained LLMs with general-purpose time series understanding capabilities. As a pivotal intermediate stage, mid-training achieves task- and domain-agnostic alignment between time series and natural language, for which we construct *Book-of-Thoth*, a high-quality, time-series-centric mid-training corpus. *Book-of-Thoth* enables both time-series-to-text and text-to-time-series generation, equipping LLMs with a foundational grasp of temporal patterns. To better evaluate advanced reasoning capabilities, we further present KnoTS, a novel benchmark of knowledge-intensive time series understanding, designed for joint reasoning over temporal patterns and domain knowledge. Extensive experiments demonstrate that mid-training with *Book-of-Thoth* enables Thoth to significantly outperform its base model and advanced LLMs across a range of time series question answering benchmarks. Moreover, Thoth exhibits superior capabilities when fine-tuned under data scarcity, underscoring the effectiveness of mid-training for time series understanding.

1 INTRODUCTION

Large Language Models (LLMs) (Achiam et al., 2023; Zhao et al., 2023) have demonstrated remarkable capabilities in sequence modeling, complex reasoning (Wei et al., 2022), and mathematical problem-solving (Azerbayev et al., 2023), largely driven by pre-training on massive textual corpora. However, real-world data are not exclusively represented in language but are instead often recorded as time series. Such time series data underpin numerous practical applications, including finance (Tsay, 2005), healthcare (Kaushik et al., 2020), and transportation (Lippi et al., 2013). Bridging LLMs with time series data is therefore a critical step toward enabling LLM-based systems to reason over temporally grounded information and support more reliable decision-making in real-world scenarios.

Despite their promise, LLMs are fundamentally built upon language modeling and are not explicitly designed to model time series data, making it challenging for them to capture fine-grained temporal dependencies and the complex dynamics inherent in time series data. To address this gap, recent studies have predominantly relied on supervised fine-tuning (SFT), adapting LLMs to time series tasks using task-specific labeled datasets (Wang et al., 2025a; Kong et al., 2025; Ni et al., 2026). While effective within task-specific scenarios, such approaches often require substantial labeled data and exhibit limited generalization across diverse temporal patterns, tasks, and domains, thereby constraining their scalability and highlighting the limitations of post-training.

Recent research on large language models has witnessed a pronounced shift toward mid-training. Positioned between large-scale pre-training and task-specific post-training, mid-training serves as an intermediate stage for reconciling general-purpose competencies with domain-specific specialization. As illustrated in Figure 1, it acts as a bridge that expands the model's knowledge coverage by incorporating a large-scale, high-quality time series-centric corpus. By preserving a proportion of general-purpose data, this stage maintains foundational competencies of LLMs while amplifying targeted capabilities. Such intermediate adaptation better prepares the model for subsequent post-training, enabling a more stable and effective transition toward specialized task alignment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

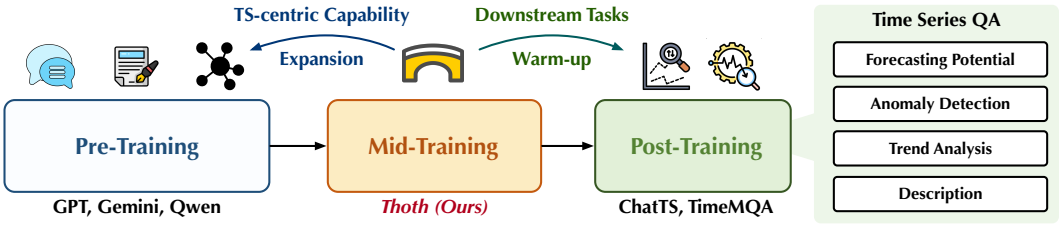


Figure 1: Pivotal role of mid-training in bridging LLMs to time series understanding. Mid-training expands the capabilities of pre-trained LLMs while providing a critical transition and warm-up for subsequent post-training on specific time series question answering tasks.

Despite its growing success in domain expansion, mid-training remains largely unexplored in time series analysis, primarily because it relies heavily on suitable large-scale training data. In contrast to task-specific fine-tuning, mid-training aims to endow models with general-purpose capabilities, which requires training data to be broad in scope, diverse in structure, and grounded in language. However, most real-world time series multimodal datasets are narrowly focused on individual tasks, such as forecasting or classification, and are curated to directly map time series to downstream task labels, making them ill-suited for endowing LLMs with general time series understanding.

Based on these motivations, we introduce **Thoth** [1] , the first model family that leverages mid-training as an intermediate stage to bridge LLMs and time series understanding. To support task- and domain-agnostic alignment between temporal data and natural language, we construct *Book-of-Thoth*, a large-scale, time-series-centric mid-training corpus. *Book-of-Thoth* comprises complementary time-series-to-text and text-to-time-series generation as core mid-training tasks, exposing models to diverse temporal dynamics and their linguistic descriptions. Mid-training on *Book-of-Thoth* enables Thoth to acquire transferable temporal representations while preserving the general reasoning capabilities of the base LLMs. To better evaluate the advanced reasoning capabilities of LLMs over both time series data and their inherent domain-specific knowledge, we introduce KnoTS, a novel benchmark of knowledge-intensive time series understanding. Experimentally, Thoth achieves substantial improvements on KnoTS and a wide range of time series question answering benchmarks, demonstrating strong few-shot and fine-tuned capabilities. These results indicate that Thoth serves as a critical intermediate step between general-purpose LLMs and advanced time series understanding. Our contributions are summarized as follows:

- We propose mid-training as a critical stage to endow LLMs with time series understanding.
- We automatically construct *Book-of-Thoth*, a large-scale, time-series-centric corpus for aligning temporal data with natural language. It features time-series-to-text and text-to-time-series generation as core tasks.
- We present Thoth, the first family of mid-trained LLMs with general-purpose time series understanding, which exhibits strong generalization across diverse time series question answering tasks.
- We introduce KnoTS, a novel benchmark for knowledge-intensive time series understanding.

2 Related Work

2.1 Mid-Training on LLMs

Mid-training has emerged as an important paradigm for LLMs. Positioned between large-scale pretraining and task-specific post-training, mid-training exposes models to high-quality specialized corpora, enabling domain expansion while preserving their general-purpose capabilities.

The efficacy of mid-training is increasingly recognized across a wide range of domains. In the medical and legal domains, models such as PMC-LLaMA (Wu et al., 2024) and SaulLM (Colombo et al., 2024) illustrate how large-scale ingestion of domain-specific literature bridges the gap between

[1]Thoth is the ancient Egyptian god of wisdom and the moon, traditionally depicted with the distinctive head of an ibis. He was credited with creating writing, reckoning time and seasons, and inventing the calendar. Many ancient Egyptian texts were attributed to him, collectively known as the Book of Thoth.

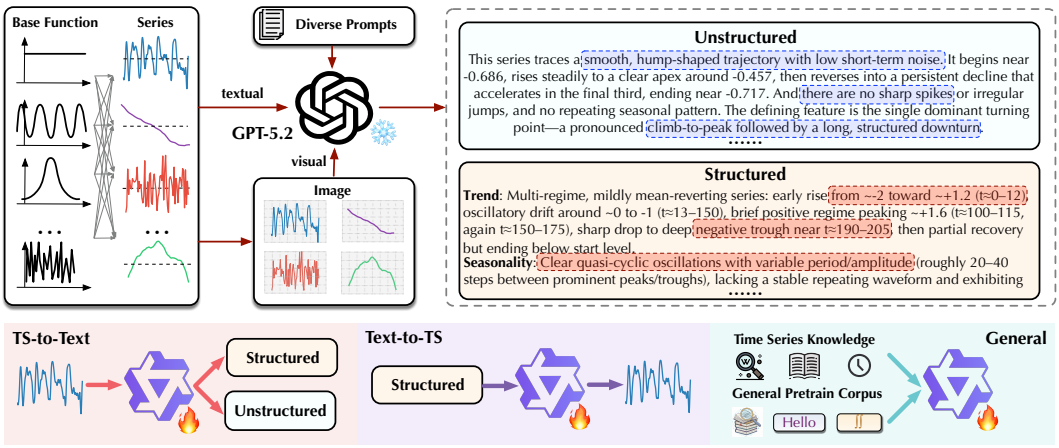


Figure 2: The automated pipeline starts by synthesizing time series through combinations of diverse base functions. For each generated series, both the raw values and their visualizations are provided to GPT-5.2 to produce a diverse range of time-series-to-text descriptions. These textual descriptions are then paired with the corresponding time series to construct complementary text-to-time-series generation data. In addition, a curated corpus of textual time series knowledge is incorporated to preserve the general-purpose capabilities of the pre-trained models during mid-training.

general linguistic logic and highly structured professional expertise. In code intelligence, models like Codex (Chen et al., 2021) and DeepSeek-Coder (Guo et al., 2024) leverage multi-stage training on heterogeneous programming corpora to enhance complex problem-solving abilities. Similarly, specialized datasets for the scientific sector, such as those used by GeoGalactica (Lin et al., 2023), improve technical reasoning in niche scientific fields.

Nevertheless, intensive training within these knowledge silos can sometimes lead to "vertical forgetting" (Tu et al., 2025), where models may lose general capabilities. Prior work has explored regularization-based and rehearsal-based approaches to alleviate this issue.

2.2 Time Series Language Models

The intersection of time series modeling and LLMs has recently attracted increasing attention, particularly in the context of time series question answering (TSQA), driving the development of diverse datasets and specialized modeling techniques. Initially, (Merrill et al., 2024) show, via human-expert evaluations, that even highly capable LLMs exhibit limited time series understanding. ChatTS (Xie et al., 2024) establishes a multimodal framework by treating time series as a native modality and leveraging synthetic data for alignment. ChatTime (Wang et al., 2025a) innovatively models temporal data as a "foreign language", utilizing multi-stage training to enable zero-shot time series forecasting and bimodal interactions. To expand task diversity, Time-MQA (Kong et al., 2025) introduces the TSQA dataset, which unifies multiple time series tasks into a comprehensive multi-task question-answering framework. ITFormer (Wang et al., 2025b) further advances efficient integration by bridging temporal encoders with frozen LLMs through a lightweight instruction-tuning paradigm. More recently, TIMEOMNI-1 (Guan et al., 2025) introduces TSR-SUITE to incorporate Chain-of-Thought (CoT) reasoning, shifting the focus from surface-level alignment to complex causal understanding and decision-making.

Despite the progress, most existing approaches rely heavily on instruction tuning on task-specific datasets. Recent advances in general-purpose LLMs suggest that strong domain performance often benefits from mid-training between large-scale pretraining and task-level adaptation (Tu et al., 2025). However, such an intermediate stage for time series remains underexplored, which motivates the need for a mid-training phase tailored to time series understanding.

3 Problem Formulation

Time series question answering is a prevalent paradigm for integrating LLMs with time series data. Given time series $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\} \in \mathbb{R}^{T \times C}$, with T time steps and C variates, and a natural

language query $\mathcal{Q}$, the goal is to generate a text-based response $\mathcal{A}$ that addresses $\mathcal{Q}$ by grounding the response in the evidence contained in $\mathbf{x}$. Formally, the task can be formulated as follows:

$$f_\theta(\mathcal{Q}, \mathbf{X}) \rightarrow \mathcal{A}, \tag{1}$$

where f_θ denotes the model that models the query $\mathcal{Q}$ and the time series $\mathbf{x}$ to produce the corresponding answer $\mathcal{A}$. Depending on the query, the response may consist of direct descriptions derived from observed values, or higher-level, context-aware reasoning that incorporates the semantic background provided in $\mathcal{Q}$ and justifies conclusions using evidence from $\mathbf{X}$.

4 Approach

4.1 Book of Thoth: Mid-Training Corpus

We present *Book-of-Thoth*, a comprehensive dataset designed for mid-training LLMs that integrates time series data with natural language descriptions. Compared to existing datasets, which are typically designed for narrowly defined objectives such as forecasting or pattern classification, *Book-of-Thoth* is constructed to capture the bidirectional relationships between temporal patterns and textual representations, offering a robust foundation for time series understanding, as illustrated in Figure 2.

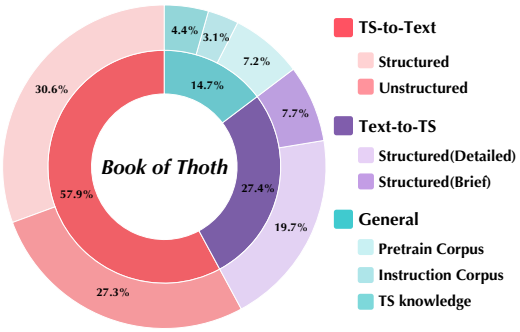


Figure 3: Components of *Book of Thoth* with 26.6M tokens.

Specifically, *Book-of-Thoth* contains two key interrelated components: (1) time series-to-text, in which each time series is annotated with a natural language caption highlighting salient temporal features, and (2) text-to-time series, in which textual descriptions are aligned with corresponding time series. To ensure scalability and consistency, an automated pipeline is developed to generate both types of high-quality time series–text pairs.

Time Series-to-Text The pipeline begins with time series generation. We employ KernelSynth (Ansari et al., 2024), a Gaussian-process–based method, to synthesize time series with rich and diverse temporal patterns. This *domain-agnostic* synthetic generation provides a stable and controllable foundation for large-scale data construction. We then generate natural language descriptions from multiple perspectives through meticulously crafted design prompts with GPT-5.2. The resulting textual descriptions can be broadly categorized as structured or unstructured. Structured descriptions provide detailed, high-quality characterizations of time series properties from specific perspectives, such as trend, seasonality, and peak behavior. In contrast, unstructured descriptions consist of concise captions or summaries that emphasize the most salient temporal characteristics. These two forms of textual data are complementary, enabling the model to learn both fine-grained temporal patterns and high-level semantic abstractions.

Text-to-Time Series Based on the structured descriptions obtained in the time series-to-text stage, we construct text-to-time series pairs by exchanging the source and target roles of the time series and description. Each structured description specifies explicit temporal characteristics and is paired with its corresponding time series generated in the previous stage. This construction does not introduce additional synthesis and maintains a consistent alignment between the text and the time series.

Textual Time Series Knowledge Additionally, we incorporate open-source time series knowledge from two authoritative references: *Forecasting: Principles and Practice* (Hyndman & Athanasopoulos, 2018) and *Time Series Analysis and Its Applications* (Shumway & Stoffer, 2006). These sources provide a solid theoretical foundation and practical guidelines for time series analysis and forecasting. We further employ GPT-4o-mini for data cleaning, including the removal of irrelevant content and logical refinement, to improve the quality and consistency of the training data.

To mitigate catastrophic forgetting during mid-training, we adopt a standard mid-training strategy by mixing a small proportion of general-purpose LLM pre-training data into the training corpus.

Specifically, we include samples from the C4 (Raffel et al., 2020) and No Robots Rajani et al. (2023). This data mixture helps preserve general linguistic and world knowledge while enabling effective adaptation to time series–specific tasks. The proportions of each component are shown in Figure 3.

4.2 THOTH: MID-TRAINED LLMS

Architecture Thoth is a model family built upon the Qwen3 architecture and mid-trained on two model variants: Qwen3-30B-A3B-Instruct and Qwen3-8B (Yang et al., 2025). Both variants share the core Qwen3 design, including Grouped Query Attention (GQA) and Rotary Positional Embeddings (RoPE) for efficient sequence modeling, as well as QK-Norm and RMSNorm for training stability.

Training We follow the default Qwen3 training recipe and keep the backbone architecture unchanged. All models are trained via full-parameter fine-tuning, with DeepSpeed ZeRO-3 employed for memory-efficient optimization. Thoth is optimized using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ under a cosine learning rate schedule. The peak learning rate is set to 1×10^{-5}, with a warmup ratio of 0.1. Training is performed in `bfloat16` precision with a maximum sequence length of 4096 tokens. We use a global batch size of 32 and training runs for 3 epochs, resulting in approximately 1,600 optimization steps. Training is conducted on 8 NVIDIA A800 GPUs with 80GB of memory.

5 KNOTS

Existing time series question answering (TSQA) benchmarks (Kong et al., 2025; Wang et al., 2025a) predominantly focus on a limited set of temporal abilities, such as trend analysis, outlier identification, short-term forecasting, and simple judgment questions. Moreover, many existing benchmarks are constructed from a small collection of real-world data or rely heavily on synthetic time series and recurring question templates. Such settings often require limited domain knowledge and may allow models to exploit regularities in question patterns instead of interpreting temporal observations in their application context. To address these limitations, we introduce **KnoTS** (**Kno**wledge-intensive **T**ime **S**eries QA), a benchmark for evaluating whether models can jointly reason over real-world time series and domain knowledge. KnoTS is constructed from publicly available time series in six application domains: finance, weather and climate, power systems, hydrology, space weather, and air quality. For each time series sample, human annotators provide domain-specific background information to support interpretation of the observed temporal patterns. Questions are then constructed such that producing a correct answer requires grounding in both the time series and the associated background knowledge, rather than relying on either source alone. Figure 4 provides a representative example from KnoTS.

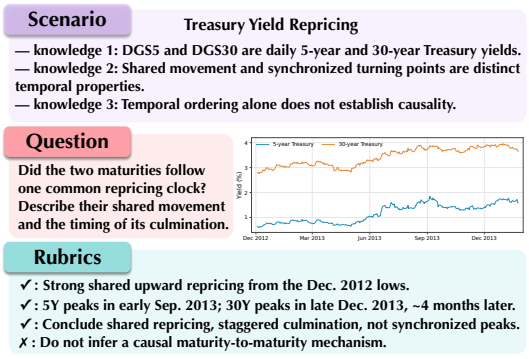


Figure 4: A brief case from the KnoTS benchmark. The complete version is provided in Figure 11.

To support reliable evaluation of open-ended responses, each question is further paired with a human-written rubric that decomposes the expected answer into explicit evaluation criteria. Given a response from a model under evaluation, a state-of-the-art language model serves as the judge and determines whether each rubric criterion is satisfied. We report two complementary metrics. The **binary score** assigns a score of 1 only when all rubric criteria are satisfied and 0 otherwise, providing a strict measure of answer correctness. The **partial score** instead awards the points associated with each satisfied criterion, allowing partially correct responses to receive proportional credit. This rubric-based evaluation enables fine-grained assessment while retaining the flexibility needed for questions that admit open-form answers. Further details of KnoTS are provided in Appendix E.

Table 1: Performance comparison across existing benchmarks. For the Time-MQA tasks, we report evaluation metrics including SMAPE for forecasting and imputation tasks, LLM-as-a-judge for open-ended tasks, and Accuracy (ACC) for others. For the ChatTime benchmark, the table reports the average results across all input lengths. The best results are highlighted in **bold** and the second best are underlined.

Model	ChatTime Benchmark				Time-MQA Benchmark						
	Trend ↑	Volatility ↑	Seasonality ↑	Outliers ↑	Forecast ↓	Impu. ↓	Clas. ↑	Anom. ↑	Judge. ↑	MCQ ↑	Open. ↑
Proprietary Models											
Gemini-3-Flash-Preview	0.870	**0.887**	0.520	0.747	0.642	0.263	0.34	0.70	0.62	0.52	6.60
GPT-4o-mini	0.850	0.433	0.427	0.673	0.546	0.252	0.28	**0.80**	0.76	0.60	7.32
Grok-4.1-Fast	0.647	0.383	**0.620**	0.747	0.503	0.303	0.20	0.68	**0.78**	0.60	7.62
Open-source Large Language Models											
Qwen3-235B-A22B-Instruct	0.867	0.513	0.447	0.667	0.458	**0.220**	0.36	0.72	0.72	0.64	**7.76**
Qwen3-30B-A3B-Instruct	0.780	0.467	0.337	0.443	0.531	0.426	0.04	0.66	0.42	0.56	6.62
Deepseek-R1-32B	0.533	0.450	0.447	0.623	0.554	0.361	0.42	0.52	0.56	0.48	6.14
Mistral-Small-24B-Instruct	0.850	0.513	0.357	0.627	0.593	0.276	0.18	0.36	0.42	0.46	6.54
Llama-3.1-8B-Instruct	0.500	0.350	0.317	0.383	0.587	0.356	0.48	0.60	0.64	0.62	4.22
Qwen3-8B	0.490	0.337	0.497	0.640	0.671	0.533	0.02	0.48	0.62	0.54	6.28
Open-source Vision Language Models											
Qwen3-VL-30B-A3B-Instruct	0.860	0.460	0.350	0.470	0.610	0.278	0.08	0.62	0.72	0.48	7.08
Mistral-Small-3.2-24B-Instruct	0.430	0.373	0.597	0.563	0.578	0.270	0.36	0.66	0.74	0.52	6.54
Task-Specific Time Series Language Models											
ChatTS-14B	-	-	-	-	1.166	-	0.24	0.44	0.58	0.46	4.18
ChatTime-7B-Chat	0.980	0.870	0.730	0.963	-	-	-	-	-	-	-
Time-MQA(LLaMA3-8B)	0.653	0.433	0.300	0.393	0.528	0.307	0.44	0.58	0.72	0.64	4.80
Time-MQA(Qwen2.5-7B)	0.297	0.370	0.343	0.390	0.480	0.251	0.28	0.64	0.66	0.46	5.06
Ours											
Thoth-8B	**0.973**	0.750	0.603	**0.753**	0.522	0.333	0.48	0.58	0.72	0.64	5.98
Thoth-30B-A3B	0.957	0.770	0.573	0.740	**0.432**	0.247	**0.52**	0.70	**0.78**	**0.70**	7.42

6 EXPERIMENT

Datasets To comprehensively evaluate the time series understanding capabilities of Thoth, we adopt two well-acknowledged benchmarks: ChatTime (Wang et al., 2025a) and Time-MQA (Kong et al., 2025). ChatTime is designed for time series pattern recognition, focusing on four fundamental temporal characteristics: trend, volatility, seasonality, and outliers. All tasks are formulated as multiple-choice questions, with ground-truth answers precisely defined by the data generation process. Time-MQA is a real-world multi-task question-answering benchmark that encompasses seven diverse tasks, including Forecasting, Imputation, Classification, Anomaly Detection, Judgment, MCQ, and Open-Ended question answering. Together, these benchmarks provide a comprehensive evaluation framework, enabling us to assess both fine-grained discriminative understanding of temporal patterns and reasoning abilities in real-world time series scenarios.

Baselines We evaluate Thoth against 15 advanced baseline models, spanning four categories: (1) *Proprietary models*: Gemini-3-Flash-Preview, GPT-4o-mini, Grok-4.1-Fast, (2) *Open-source Large Language Models*: Qwen3-235B-A22B-Instruct (Yang et al., 2025), Qwen3-30B-A3B-Instruct (Yang et al., 2025), Deepseek-R1-32B (Guo et al., 2025), Mistral-Small-24B-Instruct, LLaMA3.1-8B-Instruct, Qwen3-8B (Yang et al., 2025), (3) *Open-source Vision Language Models*: Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025), Mistral-Small-3.2-24B-Instruct, and (4) *Task-Specific Time Series Language Models*: ChatTS-14B (Xie et al., 2024), Time-MQA (Llama3-8B) (Kong et al., 2025), Time-MQA (Qwen2.5-7B) (Kong et al., 2025), ChatTime-7B-Chat (Wang et al., 2025a).

Evaluation Details To evaluate the time series understanding capabilities of Thoth while ensuring a fair comparison with baselines, we employ the widely recognized *n-shot in-context learning* protocol (Brown et al., 2020). Specifically, we provide 3-5 fixed examples per task as demonstrations in the prompt for all models. This range is chosen to maintain an optimal balance between context window constraints and the need for class representation across different categories. These examples serve to guide the LLM in following complex instructions and adhering to specific response formats, such as structured multiple-choice answers.

In terms of evaluation metrics, the four sub-tasks within ChatTime are evaluated using Accuracy (ACC), as all tasks are formulated as multiple-choice questions. For Time-MQA, metrics are task-specific: Forecasting and Imputation are evaluated using Symmetric Mean Absolute Percentage Error (SMAPE), which is robust to scale variations commonly observed in real-world time series. Classification, Anomaly Detection, Judgement, and Multiple-Choice Question tasks are evaluated

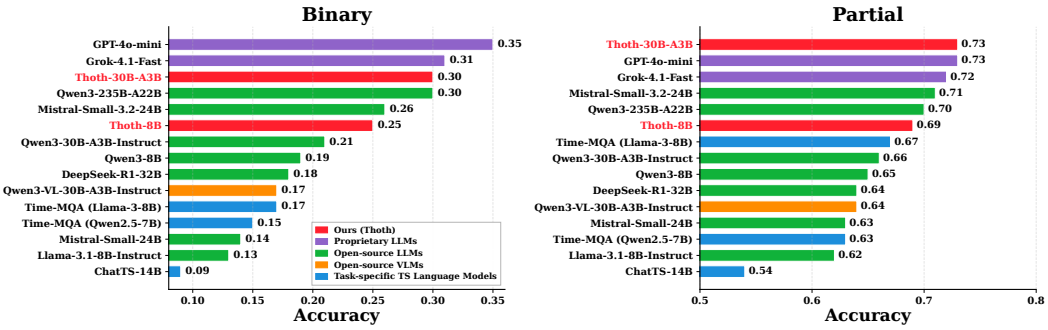


Figure 5: Performance of various models on the KnoTS benchmark. The reported results are averaged over the six KnoTS subsets, with the complete subset-level results provided in Appendix G.

via Accuracy (ACC). For the Open-Ended task and our newly introduced KnoTS, where explicit ground-truth labels are unavailable, we employ the advanced Gemini-3.5-Pro as an automated evaluator, which assigns scores along multiple dimensions, including factual correctness, temporal logic, completeness, and conciseness.

6.1 Main Results

Experimental results are presented in Table 1. Thoth demonstrates remarkable capabilities in time series question answering. Thoth-30B-A3B achieves overall performance comparable to, and in some cases exceeding, the 235B-parameter Qwen3 model. Meanwhile, Thoth-8B performs on par with open-source models at around 30B parameters, and substantially outperforms the base Qwen3-8B.

Downstream performance for Time Series Language Models is shown in gray. These models are directly supervised fine-tuned on downstream datasets, which often leads to pronounced cross-task imbalance and unstable performance across different tasks. Notably, Open-source VLMs that incorporate time series plots as visual inputs consistently outperform parameter-matched LLMs, highlighting the importance of visual information for time series understanding. This observation further underscores the effectiveness of constructing *Book-of-Thoth* with dual inputs of time series text and images when using GPT-5.2 for data generation.

The ChatTime benchmark further stratifies evaluations by input length, and the complete results are provided in Appendix G. As LLM-based generation is inherently probabilistic, we additionally conduct an error bar analysis to quantify uncertainty, with detailed results reported in Appendix C.1.

Beyond existing public TSQA benchmarks, we further evaluate models on KnoTS, which requires joint reasoning over real-world time series and domain-specific background knowledge. Model responses are evaluated using the human-written rubrics, with Gemini-3.5-Pro serving as the judge model. To reduce potential evaluator bias, we exclude Gemini-family models from the baselines.

As shown in Figure 5, Thoth exhibits strong performance on both evaluation metrics. Thoth-30B-A3B achieves the highest partial score of 0.73, matching GPT-4o-mini and outperforming Grok-4.1-Fast and Qwen3-235B-A22B. On the stricter binary metric, Thoth-30B-A3B reaches 0.30, matching Qwen3-235B-A22B and remaining competitive with proprietary LLMs. Thoth-8B also shows a clear advantage at a smaller model scale, achieving 0.25 binary and 0.69 partial accuracy. These results suggest that the gains from Thoth are not limited to conventional time series understanding, but extend to settings that require combining temporal evidence with domain knowledge.

6.2 Supervised Fine-tuning Results

As a mid-trained LLM, Thoth is designed to provide a critical warm-up for post-training and specific downstream tasks. In this section, we fine-tune Thoth on a randomly sampled 5% subset from Time-MQA and ChatTime, respectively, while preserving the original data distribution within each benchmark. We compare the results with those obtained by directly fine-tuning the original Qwen3-8B model using the same data subset. As illustrated in Figure 6, after supervised fine-tuning, Thoth significantly outperforms the baseline LLMs on the vast majority of tasks, demonstrating its effectiveness for warm-up downstream tasks.

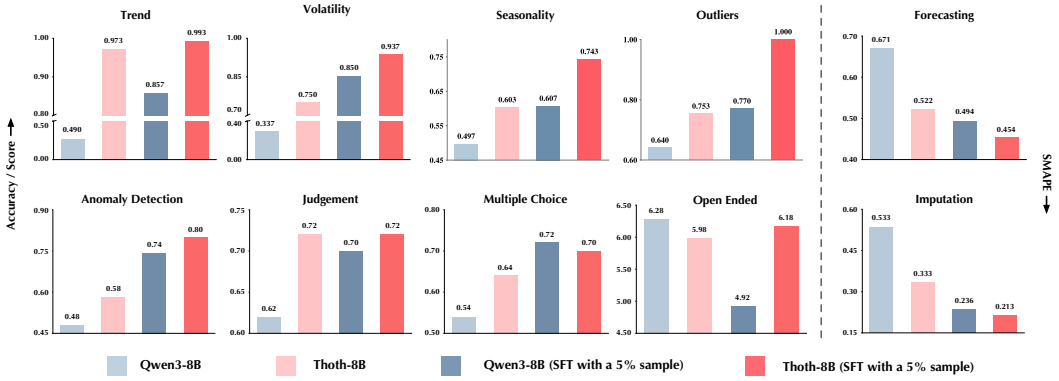


Figure 6: Performance of Supervised Fine-tuning on ChatTime and Time-MQA benchmark using a 5% sampled training set. For each benchmark, an equal proportion of data is randomly selected from each sub-task for training, with no separate model trained per sub-task.

Table 2: Ablation results on *Book-of-Thoth*. We separately remove the three types of corpora, *Time series-to-text pairs* (TS-to-Text), *Text-to-time series pairs* (Text-to-TS), and *General pre-training data* (General), for training and compare them with Thoth-8B.

Model	ChatTime Benchmark				Time-MQA Benchmark						
	Trend ↑	Volatility ↑	Seasonality ↑	Outliers ↑	Forecast ↓	Impu. ↓	Clas. ↑	Anom. ↑	Judge. ↑	MCQ ↑	Open.
Qwen3-8B	0.490	0.337	0.497	0.640	0.671	0.533	0.02	0.48	0.62	0.54	6.28
– TS-to-Text	0.360	0.487	0.347	0.453	**0.495**	**0.286**	0.14	0.48	0.68	0.46	3.32
– Text-to-TS	0.967	0.677	**0.630**	0.733	0.532	0.476	**0.50**	0.56	0.68	0.58	5.10
– General	0.963	0.547	0.387	**0.763**	0.538	0.363	0.48	**0.58**	0.70	0.60	**6.12**
Thoth-8B	**0.973**	**0.750**	0.603	0.753	0.522	0.333	0.48	**0.58**	**0.72**	**0.64**	5.98

An interesting observation is that performance on open-ended tasks decreases sharply after fine-tuning Qwen3-8B, suggesting that directly fine-tuning on multiple downstream tasks can induce an undesirable capability trade-off, where gains on certain skills come at the expense of others. In contrast, fine-tuned Thoth-8B achieves a stable performance improvement. This is attributed to its mid-training phase, which allows the model to leverage its established understanding to assimilate task-specific knowledge during multi-task fine-tuning rather than resorting to rote memorization.

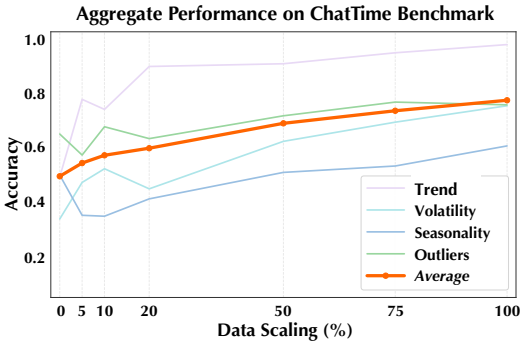


Figure 7: Aggregate performance of Thoth-8B on ChatTime benchmark at different mid-training data scales. As the data scale increases, Thoth-8B exhibits stronger average performance.

6.3 Data Scaling

In this section, we examine the empirical scaling behavior of our model by conducting mid-training experiments across varying data regimes. Specifically, we utilize subsets of *Book-of-Thoth* representing 10%, 20%, 50%, 75%, and 100% of the total volume, while maintaining a consistent data distribution. The results presented in Figure 7 reveal a clear trend that the model performance improves steadily as the dataset size increases, highlighting the significant potential of data scaling.

6.4 Ablation Study

To verify the effectiveness of *Book-of-Thoth* components, we provide detailed ablations by removing each of the three components individually. The results are listed in Table 2. Upon removing the *TS-to-Text* component, since the remaining dataset predominantly consists of time series generation, the performance of two tasks related to time series generation, forecasting and imputation, improves, although their overall performance remains suboptimal.

Interestingly, while *Text-to-TS* enhances overall time series understanding, the improvement is less pronounced compared to *TS-to-Text*. This discrepancy likely arises from the fact that reconstructing precise time series from text presents a much greater challenge than generating a textual description from time series. In contexts with limited model capacity, simpler auxiliary tasks often yield greater benefits. A typical instance is the Masked Language Modeling (MLM) task in BERT (Devlin et al., 2019), which, by capturing bidirectional contextual information at the token level, proves highly efficient for feature extraction in smaller models. However, as model size increases, the more complex Next Token Prediction (NTP) task, following the GPT paradigm, demonstrates its advantages, aligning with the enhanced generative capabilities of larger models (Brown et al., 2020). Although the improvement from *Text-to-TS* may not currently be as pronounced as that from *TS-to-Text*, it still offers valuable enhancement and remains an indispensable potential for advancing time series understanding.

6.5 Case Study

In this section, we provide a case study in Figure 8, showing the simplified inference results of Thoth-30B-A3B on the KnoTS benchmark. As illustrated, the model infers the rainfall-runoff across two distinct stages based on time series analysis. These results are consistent with the ground truth, demonstrating Thoth's robust time series reasoning capabilities in domain-specific scenarios.

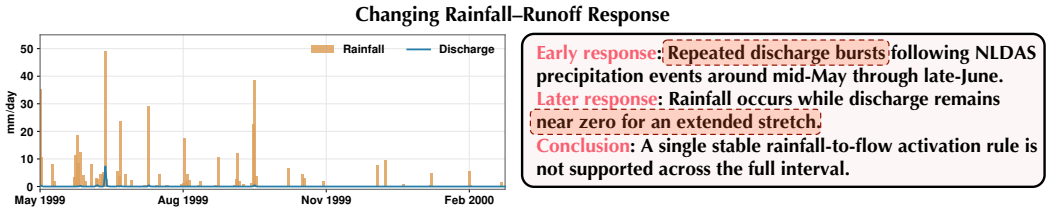


Figure 8: Case study of Thoth-30B-A3B model response extracted on the KnoTS benchmark. Detailed comparative results of various models can be found in Appendix F for further reference.

7 Discussions

Given the prevalence of time series data in real-world applications, empowering LLMs with time series understanding capability is of increasing importance. This paper treats mid-training as a practical bridge that adapts LLM toward time series understanding. To this end, we construct *Book-of-Thoth*, a large-scale, task- and domain-agnostic, time series–centric corpus that aligns time series with natural language, and use it to train Thoth, a family of general-purpose LLMs with substantially improved time series understanding. To better evaluate the time series reasoning capabilities of LLMs, we further propose a knowledge-intensive time series QA benchmark that emphasizes reasoning and decision-making grounded in temporal evidence. Extensive experiments demonstrate that Thoth delivers strong few-shot performance on a broad range of time series tasks. Moreover, a supervised fine-tuning study shows that mid-training effectively warms up downstream time-series QA, enabling consistent gains with limited task-specific data. Analysis experiment demonstrates strong scalability and provides valuable insights.

Limitations and Future Work This paper focuses on bridging LLMs to time series understanding via mid-training. Looking ahead, it is valuable to explore further advancements in post-training to enhance time series understanding and reasoning. Figure 1 illustrates the powerful capability of mid-training in warming up downstream tasks. However, it is evident that direct fine-tuning of multi-task learning for downstream tasks may lead to performance degradation in certain tasks, due to the imbalanced requirements across tasks. Therefore, reinforcement learning, which has been widely proven effective in prior work on question-answering tasks, is worth investigating.

Table 2 demonstrates the impact of different components of the *Book-of-Thoth* framework on model performance. The text-to-time series training task has shown promising potential; however, as discussed in Section 6.4, there is still room for improvement on larger-scale models and datasets. Moreover, it remains an open question whether the training signal of generated time series should be based on its alignment with textual information, rather than the traditional next-token prediction loss.

AI USE STATEMENT

In this work, we used generative AI tools to support distinct parts of the research process: GPT-5.2 generated structured and unstructured descriptions of synthetic time series for Book-of-Thoth, GPT-4o-mini assisted with cleaning and refining textual time series knowledge, and Gemini-3.5-Pro evaluated open-ended responses in Time-MQA and KnoTS, using human-written rubrics for KnoTS. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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.

Nans Addor, Andrew J Newman, Naoki Mizukami, and Martyn P Clark. The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313, 2017.

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.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. *arXiv preprint arXiv:2509.24803*, 2025.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

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.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

J. H. King and N. E. Papitashvili. Solar wind spatial scales in and comparisons of hourly Wind and ACE plasma and magnetic field data. *Journal of Geophysical Research: Space Physics*, 110(A2): A02104, 2005. doi: 10.1029/2004JA010649.

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.

Zhouhan Lin, Cheng Deng, Le Zhou, Tianhang Zhang, Yi Xu, Yutong Xu, Zhongmou He, Yuanyuan Shi, Beiya Dai, Yunchong Song, et al. Geogalactica: A scientific large language model in geoscience. *arXiv preprint arXiv:2401.00434*, 2023.

Marco Lippi, Matteo Bertini, and Paolo Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):871–882, 2013.

Diego Mendez-Carbajo and Genevieve M. Podleski. Federal reserve economic data: A history. *The American Economist*, 66(1):61–73, 2021. doi: 10.1177/0569434520973989.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3512–3533, 2024.

Andrew J Newman, Martyn P Clark, Kevin Sampson, Andrew Wood, Lauren E Hay, Andy Bock, Roland J Viger, David Blodgett, Levi Brekke, JR Arnold, et al. Development of a large-sample watershed-scale hydrometeorological data set for the contiguous usa: data set characteristics and assessment of regional variability in hydrologic model performance. *Hydrology and earth system sciences*, 19(1):209–223, 2015.

Juntong Ni, Shiyu Wang, Qi He, Ming Jin, and Wei Jin. Streasoner: Empowering llms for spatio-temporal reasoning in time series via spatial-aware reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15369–15416, 2026.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Nazneen Rajani, Lewis Tunstall, Edward Beeching, Nathan Lambert, Alexander M. Rush, and Thomas Wolf. No robots. `https://huggingface.co/datasets/HuggingFaceH4/no_robots`, 2023.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

Robert H Shumway and David S Stoffer. *Time series analysis and its applications: with R examples*. Springer, 2006.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Ruey S Tsay. *Analysis of financial time series*. John wiley & sons, 2005.

Chengying Tu, Xuemiao Zhang, Rongxiang Weng, Rumei Li, Chen Zhang, Yang Bai, Hongfei Yan, Jingang Wang, and Xunliang Cai. A survey on llm mid-training. *arXiv preprint arXiv:2510.23081*, 2025.

U.S. Energy Information Administration. EIA's hourly electric grid monitor provides timely data about electricity usage patterns. Today in Energy, April 2020. URL `https://www.eia.gov/todayinenergy/detail.php?id=43295`. Published April 6, 2020.

U.S. Environmental Protection Agency. Air quality system data mart. Internet database, 2026. URL `https://www.epa.gov/outdoor-air-quality-data`. Accessed September 24, 2026.

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.

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. *arXiv preprint arXiv:2506.20093*, 2025b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843, 2024.

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, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A Downstream Tasks

We evaluate our model on two prominent multi-task time series question answering (QA) benchmarks: Time-MQA (Kong et al., 2025) and ChatTime (Wang et al., 2025a). These benchmarks cover a wide range of time series analysis tasks, spanning both traditional numerical analysis and more advanced reasoning tasks. The datasets include various domains, such as healthcare, finance, energy, transportation, and more, providing a comprehensive foundation for model evaluation. The following sections describe the details of these benchmarks and their associated tasks.

A.1 TIME-MQA BENCHMARK

The Time-MQA framework is a multi-task time series QA system that integrates natural language queries across several time series tasks. It contains approximately 200,000 question-answering pairs derived from 12 diverse domains, including healthcare, energy, finance, traffic, IoT, and the web. This extensive dataset is designed not only to support traditional numerical tasks, such as forecasting and anomaly detection, but also to handle more complex reasoning tasks through text-enhanced queries. The multi-task structure of Time-MQA makes it an ideal benchmark for evaluating the versatility and reasoning ability of models across a range of time series problems.

The specific tasks within the Time-MQA dataset are as follows:

Forecasting The forecasting task in Time-MQA involves predicting future time points based on known time series data. This task spans multiple domains, such as healthcare and energy, and requires models to accurately predict future trends from historical observations. For instance, forecasting heart rate trends or energy consumption based on previous data points.

Imputation The imputation task addresses the challenge of filling in missing values within time series data. Time series data often suffer from gaps due to various reasons such as sensor malfunctions or incomplete data collection. The imputation task requires models to predict the missing values, thus restoring the continuity of the time series. This is especially important in fields like healthcare, where data integrity is critical.

Anomaly Detection Anomaly detection is essential in identifying unusual patterns or deviations in time series data. This task requires models to detect outliers or anomalous events that deviate from expected behavior. In industrial and financial settings, detecting anomalies in real-time data can signal critical events, such as equipment failure or market disruptions.

Classification The classification task involves categorizing time series data into predefined classes. For example, sensor data from wearable devices could be classified into different human activities, such as walking, sitting, or jogging. This task tests the model's ability to recognize patterns in time series and assign them to appropriate categories.

True/False Questions The True/False questions assess the model's ability to verify simple assertions based on the time series data. These questions ask the model to determine whether a statement about the time series is correct or not.

Multiple Choice Questions In the Multiple Choice task, the model is asked to choose the correct answer from a set of predefined options. This task evaluates the model's ability to classify and compare specific characteristics of the time series data.

Open-Ended QA Open-ended QA tasks in Time-MQA require the model to generate detailed textual explanations about the time series data. These tasks go beyond simple numerical predictions, asking the model to interpret trends, identify causal relationships, and provide insights based on the time series patterns.

These tasks collectively make Time-MQA a comprehensive benchmark for evaluating multi-task models. By integrating numerical and textual reasoning, Time-MQA challenges models to provide more advanced insights into time series data, bridging the gap between classical numerical analysis and modern language model-driven approaches.

A.2 CHATTIME BENCHMARK

The ChatTime framework is another state-of-the-art multimodal time series foundation model that incorporates both time series data and textual information. ChatTime uniquely handles both bimodal inputs and outputs, allowing models to leverage textual context alongside time series data. It offers zero-shot time series forecasting capabilities and facilitates multimodal time series question answering tasks, making it particularly useful for testing models on complex, context-sensitive tasks.

The ChatTime dataset includes tasks specifically designed to evaluate the model's ability to process time series data alongside contextual textual information. The four main tasks within this dataset are as follows:

Trend Analysis This task involves identifying the overall direction of a given time series. The model is required to determine whether the data is exhibiting an increasing, decreasing, or stable trend over time. The objective is to recognize the general movement in the data points and categorize the series accordingly.

Volatility Analysis Volatility analysis focuses on assessing the extent of fluctuations within the time series. The model is tasked with identifying periods of high volatility, where the data points show significant variations, or periods of low volatility, where the data remains relatively stable. This task requires recognizing how the data's variation changes over time.

Seasonality Analysis In seasonality analysis, the model is asked to determine if the time series displays periodic or cyclic behavior. This includes recognizing repeating patterns or trends that occur at regular intervals, indicating a seasonal influence on the data over time.

Outlier Detection Outlier detection requires the model to identify data points in the time series that significantly deviate from the general pattern or trend. These outliers might represent unusual events or anomalies, and the model needs to pinpoint these points based on their departure from expected behavior.

Each of these tasks challenges the model to integrate both time series data and textual information, requiring not only numerical analysis but also a deeper understanding of the context within which the data is generated. The multimodal nature of ChatTime makes it an ideal candidate for evaluating models on their ability to reason across both domains simultaneously.

B Implementation Details of Evaluation

We provide the implementation details for our evaluation. For the ChatTime benchmark, given the massive scale of synthesized time series question-answering pairs, we randomly sample 300 instances per task as the test set to evaluate the accuracy of LLMs in identifying time series patterns. For the Time-MQA benchmark, we follow the experimental settings in the original text by extracting 50 samples per task as test cases.Specifically, for forecasting and imputation tasks, we employ the Symmetric Mean Absolute Percentage Error (SMAPE) as the evaluation metric. This choice is justified by the fact that the dataset spans multiple application domains with extreme numerical variances and diverse sequence stationarity; SMAPE effectively assesses the quality of generation across these heterogeneous samples. The formula is defined as follows:

$$\text{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \tag{2}$$

where A_t represents the actual value and F_t denotes the forecast value. In rare cases where the sequence length generated by the LLM deviates from the instructions, we introduce a penalty factor of 1.1. The adjusted error, E_{adj}, is calculated as:

$$E_{adj} = \begin{cases} E \times 1.1, & \text{if } L_{gen} \neq L_{inst} \\ E, & \text{if } L_{gen} = L_{inst} \end{cases} \tag{3}$$

where L_{inst} and L_{gen} refer to the instructed and generated lengths, respectively. Except for open-ended questions, all other tasks utilize Accuracy (ACC) as the evaluation metric. For the open-ended task in Time-MQA and the KnoTS benchmark, we use Gemini-3.5-Pro for scoring. The model provides a comprehensive score by considering multiple dimensions, including factual correctness, temporal logic, completeness, and conciseness.

C Additional Analyses

C.1 Error Bar Analysis

To ensure the reproducibility and statistical reliability of our results, we conducted three independent runs ($N = 3$) for our main experiments using different random seeds. The values are reported as **mean** $\pm$ **standard deviation**, which accounts for the inherent stochasticity in LLM decoding and sampling processes. This analysis confirms that the performance gains observed in our study are consistent across multiple iterations and are not a result of favorable random seeding, demonstrating the stability of our proposed method.

Table 3: Robustness of Thoth-30B-A3B and Thoth-8B performance on all benchmarks. The results are obtained from three random runs.

Model	ChatTime Benchmark				Time-MQA Benchmark							KnoTS Benchmark	
	Trend ↑	Volatility ↑	Seasonality ↑	Outliers ↑	Forecast ↓	Impu. ↓	Clas. ↑	Anom. ↑	Judge. ↑	MCQ ↑	Open. ↑	Binary ↑	Partial ↑
Thoth-8B	0.973±0.003	0.750±0.005	0.603±0.007	0.753±0.003	0.522±0.019	0.333±0.013	0.48±0.00	0.58±0.01	0.72±0.02	0.64±0.02	5.98±0.17	0.25±0.01	0.69±0.02
Thoth-30B-A3B	0.957±0.003	0.770±0.010	0.573±0.010	0.740±0.007	0.432±0.021	0.247±0.014	0.52±0.00	0.70±0.00	0.78±0.01	0.70±0.02	7.42±0.11	0.30±0.01	0.73±0.01

C.2 General-Domain Capability Retention

To assess whether time-series-centric mid-training compromises the general capabilities of the base LLM, we further evaluate Qwen3-8B, Thoth-8B, and a variant trained without general-domain data (*Thoth-8B w/o General*) on two general-domain benchmarks, MMLU-Redux (Gema et al., 2025) and CommonSenseQA (Talmor et al., 2019).

As shown in Table 4, Thoth-8B retains most of the general-domain performance of the original Qwen3-8B after mid-training. More importantly, removing general-domain data consistently leads to further degradation: compared with *Thoth-8B w/o General*, Thoth-8B improves by 0.66 points on MMLU-Redux and 0.90 points on CommonSenseQA. These results indicate that mixing general-domain data during mid-training effectively mitigates capability forgetting while allowing the model to acquire time-series-specific capabilities.

Table 4: General-domain capability retention after mid-training. *Thoth-8B w/o General* denotes the variant trained without general-domain data. Higher is better.

Model	MMLU-Redux ↑	CommonSenseQA ↑
Qwen3-8B	79.50	78.46
Thoth-8B w/o General	75.40	74.45
Thoth-8B	76.06	75.35

D Prompts of *Book-of-Thoth* Generation

In this section, we present the details for constructing the mid-training general-purpose dataset, *Book-of-Thoth*. For the diverse time series data synthesized via the KernelSynth method, we design a set of carefully crafted prompts to elicit high-quality and diverse time series–text pairs from GPT-5.2. For both structured and unstructured text, we design two versions of prompts, which generate detailed and brief texts, respectively. This significantly enhances the diversity of the generated data. The prompting scheme and examples are shown in Figure 9 and Figure 10.

E Details of KnoTS

E.1 Data Sources

KnoTS is constructed from publicly available real-world time series across six application domains. For each subset, we sample domain-relevant variables from established data sources and organize them into time series of 300 time steps. Each sample contains two variables and is paired with

Structured Prompt (Detailed)

Act as an expert data scientist. Provide a high-density technical analysis of the time series provided. Input: Data={ts}, Length={len(ts)}

Output Requirements:

- **Style: Professional, analytical, and concise. No conversational filler.**
- **Objectives: Focus on structural properties and signal dynamics.**

1 Descriptive Summary

- **Stats: [Length: Value | Start: V1 | End: V2 | Range: [Min, Max] | Mean/SD: Value]**
- **Stationarity: Assess if the mean and variance are time-invariant (Stationary vs. Non-stationary).**

2 Trend & Momentum

- **Global Trend: Overall direction, slope intensity, and linearity.**
- **Local Dynamics: Identify specific segments of acceleration, deceleration, or plateaus.**
- **Structural Breaks: Note any permanent level shifts or regime changes.**

3 Pattern & Noise

- **Seasonality/Periodicity: Identify cyclic components or frequency of recurrence.**
- **Autocorrelation: Describe the degree of self-similarity and lag dependency.**
- **Volatility: Characterize noise levels and whether volatility is constant or clustered (Heteroscedasticity).**

4 Synthesis & Outliers

- **Anomalies: Locate specific point outliers or transient spikes.**
- **Technical Verdict: One-paragraph synthesis of the data's generative process and predictability.**

Use telegraphic professional prose: omit unnecessary subjects and auxiliary verbs while maintaining grammatical clarity.

Structured Prompt (Brief)

Act as an expert data scientist. Extract evidence-only observations from the given time series.
Input: Data={ts}, Length={len(ts)}

Output Requirements:
- Output MUST contain exactly 4 lines total:
1) A single Stats line (see format below)
2-4) EXACTLY 3 bullet points, each starting with "- "
- Stats line format (MUST match exactly):
Stats: [Length: Value | Start: V1 | End: V2 | Range: [Min, Max]]

Evidence Bullet Requirements (lines 2-4):
- Content: Evidence ONLY. Do NOT provide any overall verdict, diagnosis, recommendation, or predictability statement.
- Each bullet must be specific and falsifiable from the sequence:
- Refer to concrete localizations (e.g., "early/mid/late segment", "near the start/end", "around the largest peak/trough").
- Describe observable properties such as: monotonic drift, slope change, level shift, periodic recurrence, amplitude modulation, volatility clustering, abrupt spikes/drops, prolonged plateaus, changes in variance.
- Consistency constraints:
- No contradictions across bullets.
- Avoid redundant bullets; each must cover a different axis (e.g., one for trend/regime, one for periodicity/autocorr, one for volatility/anomalies).
- Style: Professional, concise, telegraphic; no conversational filler.
- Forbidden:
- No causal claims that cannot be read from the series.
- No words implying a final judgment (e.g., "therefore", "overall", "suggests that", "implies", "predictable/unpredictable").

Return only the 4 lines.

Figure 9: Examples of carefully designed prompts used for constructing structured data of *Book-of-Thoth*.

manually written domain knowledge and questions that require joint interpretation of the temporal observations and the provided context. Table 5 summarizes the benchmark composition. We describe the data source for each subset below.

Finance. The Finance subset is constructed from financial and macroeconomic series retrieved through Federal Reserve Economic Data (FRED), maintained by the Federal Reserve Bank of St. Louis. FRED provides access to economic and financial time series collected from a wide range of

Unstructured Prompt (Feature Emphasis)

Act as an expert data scientist. Analyze the following time series data with two core objectives: first, provide a concise holistic overview of key observations, and second, prioritize emphasizing the most profound, defining feature that distinguishes this sequence.

The defining feature may include (but is not limited to) strong monotonic trends, distinct seasonality/periodicity, abrupt outliers/ anomalies, extreme volatility/stability, sharp peaks/troughs, prolonged plateaus, or consistent rate of change—focus on what makes the time series uniquely identifiable.

To ensure diversity, use varied descriptive angles, terminology, and sentence structures. Avoid repetitive phrasing or overly technical jargon (keep it accessible but precise).

Input: Data={ts}, Length={len(ts)}.

Generate a single, coherent description (50-150 words) that integrates the holistic overview with a clear emphasis on the most striking feature.

Unstructured Prompt (Brief Summary)

Act as an expert data scientist. Produce an ultra-compressed single-sentence technical verdict for the given time series.

Input: Data={ts}, Length={len(ts)}

Output Requirements:
- Format: EXACTLY ONE sentence. No bullet points, no line breaks.
- Style: Professional, high-density, telegraphic; avoid conversational filler.
- Content MUST include all three components (in one sentence):
1) Generative process hypothesis: identify the dominant structure (e.g., trend / seasonality / regime switching / mean-reversion / random walk / heteroscedastic noise).
2) Predictability assessment: state predictability level (high/medium/low) with a brief justification tied to structure.
3) Risk point: name the primary failure mode for forecasting (e.g., structural break, volatility clustering, outliers, non-stationarity, weak autocorrelation).

- Constraints:
- Do NOT output numbers beyond what is necessary; prefer qualitative descriptors.
- Do NOT mention the prompt, the word "Input", or "Output Requirements".
- Do NOT hedge with multiple conflicting hypotheses; commit to the most plausible dominant mechanism.

Return only the single-sentence verdict.

Figure 10: Examples of carefully designed prompts used for constructing unstructured data of *Book-of-Thoth*.

source institutions (Mendez-Carbajo & Podleski, 2021). We select variables related to financial stress, market volatility, Federal Reserve balance sheet conditions, policy rates, and U.S. Treasury yields to capture changes in financial conditions and policy environments.

Weather. The Weather subset uses ERA5 reanalysis data distributed through WeatherBench 2 (Hersbach et al., 2020; Rasp et al., 2024). ERA5 provides a globally consistent reconstruction of atmospheric and surface conditions with hourly temporal resolution, while WeatherBench 2 provides standardized access to data for weather modeling and evaluation. We select meteorological variables that capture weather extremes and interactions between land and atmosphere.

Power. The Power subset uses hourly electricity system data released by the U.S. Energy Information Administration (EIA) through Form EIA-930 and the Hourly Electric Grid Monitor (U.S. Energy Information Administration, 2020). The data include electricity demand, demand forecasts, net generation, generation by energy source, and power interchange between balancing authorities. We use these measurements to characterize electricity supply, demand, and operational conditions.

Hydrology. The Hydrology subset is based on the CAMELS-US dataset (Newman et al., 2015; Addor et al., 2017). CAMELS-US provides hydrometeorological records for a large collection of catchments across the contiguous United States, including meteorological forcing and streamflow observations. We use these variables to characterize runoff responses to rainfall, flow persistence, and related hydrological dynamics.

Table 5: Overview of KnoTS benchmark.

Subset	Domain	Data Source	Questions	Description
Finance	Financial markets	FRED	42	Financial stress and policy transitions.
Weather	Weather and climate	ERA5	39	Weather extremes and land–atmosphere interactions.
Power	Power systems	EIA-930	46	Electricity supply, demand, and operational warnings.
Hydrology	Catchment hydrology	CAMELS-US	40	Runoff responses to rainfall and flow persistence.
Space	Space weather	NASA OMNI	37	Solar wind disturbances and geomagnetic activity.
AirQuality	Air quality	EPA AQS	38	Pollution dynamics and monitoring consistency.

Space. The Space subset is constructed from the NASA OMNI hourly dataset (King & Papitashvili, 2005). OMNI provides time-aligned measurements of solar wind plasma and the interplanetary magnetic field, together with commonly used geomagnetic quantities. We use these variables to characterize solar wind disturbances and their relationship with geomagnetic activity.

Air Quality. The AirQuality subset uses measurements from the U.S. Environmental Protection Agency (EPA) Air Quality System (AQS) (U.S. Environmental Protection Agency, 2026). AQS is a repository of ambient air pollution measurements collected by EPA, state, local, and tribal air pollution control agencies, together with meteorological data, monitoring-site information, and data quality information. We use pollutant and monitoring variables to characterize pollution dynamics and consistency across measurements.

E.2 Benchmark and Rubric Construction

KnoTS is constructed from publicly available real-world time series across six application domains. The Finance subset uses data from FRED (Mendez-Carbajo & Podleski, 2021); Weather uses ERA5 data distributed through WeatherBench 2 (Hersbach et al., 2020; Rasp et al., 2024); Power uses EIA-930 electricity system records (U.S. Energy Information Administration, 2020); Hydrology is based on CAMELS-US (Newman et al., 2015; Addor et al., 2017); Space uses the NASA OMNI dataset (King & Papitashvili, 2005); and AirQuality uses the EPA Air Quality System (U.S. Environmental Protection Agency, 2026). The benchmark statistics and data sources are summarized in Table 5.

For each instance, human annotators select a pair of domain-relevant time series and write the accompanying background information and question. The background information specifies the meaning of the variables and relevant domain conventions, while the question is designed to require joint interpretation of the temporal observations and the provided context. Annotators then write a reference answer and a question-specific rubric that identifies the key requirements of a correct response. The rubric covers the main conclusion, supporting temporal or numerical evidence, and important limits on the interpretation. It focuses on semantic correctness and allows minor numerical or temporal differences that do not alter the underlying conclusion. All benchmark instances and their rubrics are manually reviewed for consistency between the time series, background information, question, and evaluation criteria.

E.3 Evaluation Protocol

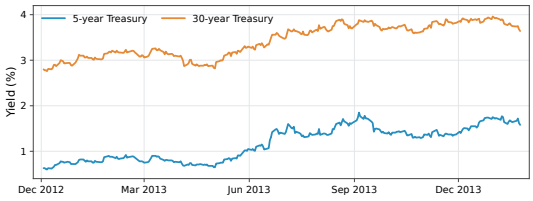
We evaluate open-ended responses using the human-written rubrics described above. For each question, Gemini-3.5-Pro serves as the judge model and determines whether the response satisfies each rubric criterion. The judge is instructed to assess semantic consistency with each criterion rather than requiring exact lexical overlap. We use the same judge prompt and inference configuration for all evaluated models.

We report two complementary metrics. The **partial score** assigns credit for each satisfied rubric criterion according to its predefined weight, allowing partially correct responses to receive proportional credit. The **binary score** is stricter and assigns a score of 1 only when all rubric criteria are satisfied; otherwise, the score is 0.

Further, we present one representative example in Figure 11. It shares two key characteristics: they are tightly grounded in domain background knowledge, and they require models to reason and act based on the real-world time series evidence. As such, KnoTS enables evaluation of time series

understanding while also testing whether incorporating time series compromises the model's general reasoning competence.

Scenario Description: The observations are frozen records from the Federal Reserve Bank of St. Louis FRED database. Weekly records use retained Friday labels. TOTLL is loans and leases in bank credit, measured in billions of U.S. dollars; WRESBAL is reserve balances and WLCFLPCL is Primary Credit outstanding, both measured in millions of U.S. dollars. NFCI and ANFCI are Chicago Fed financial-conditions indexes, with ANFCI adjusted for prevailing economic conditions. STLFSI4 is the St. Louis Fed Financial Stress Index, and VIXCLS is a weekly mean of the CBOE Volatility Index. Higher index values generally represent tighter financial conditions, greater financial stress, or greater expected equity-market volatility, respectively. DGS3MO, DGS2, DGS5, DGS10, and DGS30 are daily Treasury constant-maturity yields for the stated maturities, expressed in percent. Source-date missing values are retained and are not interpolated. The data are retrospective snapshots rather than historical real-time vintages. Temporal order can support descriptions of sequence, persistence, state change, and operational association, but it cannot by itself identify a causal policy, credit, or market mechanism. Yield-curve configurations likewise do not determine a particular future macroeconomic outcome.



Question: Did the two Treasury maturities follow one common repricing clock through this interval? Give the most defensible account of their shared movement and its culmination.

Rubrics:

R1: Hard failure if the two window maxima are treated as simultaneous or their order is reversed.
R2: Hard failure if the strong shared upward repricing is denied altogether.
R3: A strong answer must reconcile the common broad rise with the later 30-year culmination.
R4: Hard failure if the peak sequence is claimed to prove a causal maturity-to-maturity mechanism.

Figure 11: A complete example of reasoning questions in the KnoTS benchmark. It is the full version of Figure 4 in the main text.

F SHOWCASES

In this section, we present two complete examples of model inference, as illustrated in Figures 12 and 13. We select representative problems to compare the inference results of Thoth with the baseline LLMs.

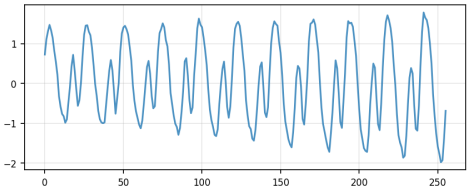
G FULL RESULTS

The ChatTime benchmark categorizes input sequences into four lengths: 64, 128, 256, and 512. However, since the 512 length sequences significantly challenge the input capacity of LLMs under n-shot in-context learning settings, we restrict our evaluation to lengths of 64, 128, and 256. The complete results are detailed in Table 6. Both parameter versions of Thoth demonstrate robust performance across all tested input lengths.

Table 7 reports the complete results on the six KnoTS subsets. We report both Binary and Partial scores for each domain. Thoth-30B achieves strong and consistent performance across the six domains, with average scores of 0.30 and 0.73, respectively. In particular, it obtains the best Binary

Question:
Given the following definitions:
- Seasonal pattern: The time series shows a repetitive and predictable fluctuation throughout the period.
- Fixed seasonal pattern: The timing and magnitude of the seasonal fluctuation remain constant over time.
- Shifting seasonal pattern: The timing or magnitude of the seasonal fluctuation changes over time.
Select one of the following answers that best describes the provided time series:
(a) This time series has no obvious seasonal pattern.
(b) This time series has a fixed seasonal pattern.
(c) This time series has a shifting seasonal pattern.
Only answer (a), (b), or (c).

Ground Truth: (b)



Models Answer:
— Gemini-3-flash: (c)
— GPT-4o-mini: (c)
— Grok-4.1-fast: (c)
— Qwen3-235B-A22B-Instruct: (c)
— Qwen3-30B-A3B-Instruct: (c)
— Qwen3-VL-30B-A3B-Instruct: (c)
— Thoth-30B-A3B: (b)
— Thoth-8B: (b)

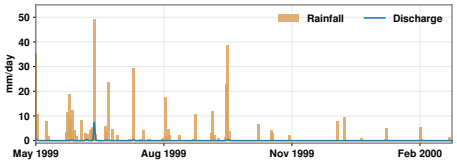
Figure 12: A showcase of seasonality recognition question from various LLMs in the ChatTime benchmark.

Table 6: Full results of ChatTime benchmark (Wang et al., 2025a). There are three fixed time series lengths of 64, 128, and 256 of the four time series pattern recognition tasks. *Avg* means the average results from the three lengths.

Model	Trend				Volatility				Seasonality				Outliers			
	64	128	256	*Avg*	64	128	256	*Avg*	64	128	256	*Avg*	64	128	256	*Avg*
Proprietary Models																
Gemini-3-Flash-Preview	0.970	0.800	0.840	0.870	**0.880**	**0.860**	**0.920**	**0.887**	0.550	0.550	0.460	0.520	0.700	**0.800**	0.740	0.747
GPT-4o-mini	0.890	0.790	0.870	0.850	0.380	0.500	0.420	0.433	0.540	0.370	0.370	0.427	0.660	0.660	0.700	0.673
Grok-4.1-Fast	0.680	0.700	0.560	0.647	0.460	0.380	0.310	0.383	0.590	**0.730**	0.540	**0.620**	0.700	0.750	**0.790**	0.747
Open-source Large Language Models																
Qwen3-235B-A22B-Instruct	0.940	0.820	0.840	0.867	0.500	0.560	0.480	0.513	0.550	0.350	0.440	0.447	0.710	0.650	0.640	0.667
Qwen3-30B-A3B-Instruct	0.840	0.700	0.800	0.780	0.490	0.480	0.430	0.467	0.400	0.310	0.300	0.337	0.530	0.450	0.350	0.443
Deepseek-R1-32B	0.540	0.670	0.390	0.533	0.530	0.460	0.360	0.450	0.510	0.350	0.480	0.447	0.720	0.640	0.510	0.623
Mistral-Small-24B-Instruct	0.860	0.870	0.820	0.850	0.530	0.550	0.460	0.513	0.380	0.340	0.350	0.357	0.690	0.580	0.610	0.627
Llama-3.1-8B-Instruct	0.800	0.410	0.290	0.500	0.360	0.380	0.310	0.350	0.250	0.400	0.300	0.317	0.360	0.410	0.380	0.383
Qwen3-8B	0.530	0.560	0.380	0.490	0.320	0.360	0.330	0.337	0.500	0.500	0.490	0.497	0.710	0.630	0.580	0.640
Open-source Vision Language Models																
Qwen3-VL-30B-A3B-Instruct	0.920	0.810	0.850	0.860	0.230	0.510	0.640	0.460	0.440	0.250	0.360	0.350	0.480	0.490	0.440	0.470
Mistral-Small-3.2-24B-Instruct	0.460	0.500	0.330	0.430	0.410	0.380	0.330	0.373	0.530	0.620	**0.640**	0.597	0.610	0.570	0.510	0.563
Task-Specific Time Series Language Models																
ChatTime-7B-Chat	0.990	0.980	0.970	0.980	0.820	0.910	0.880	0.870	0.760	0.720	0.710	0.730	0.980	0.990	0.920	0.963
Time-MQA(LLaMA3-8B)	0.720	0.580	0.660	0.653	0.410	0.500	0.390	0.433	0.330	0.230	0.340	0.300	0.480	0.360	0.340	0.393
Time-MQA(Qwen2.5-7B)	0.270	0.390	0.230	0.297	0.350	0.410	0.350	0.370	0.340	0.400	0.290	0.343	0.350	0.370	0.450	0.390
Ours																
Thoth-8B	**0.980**	**0.970**	**0.970**	**0.973**	0.770	0.770	0.710	0.750	**0.610**	0.580	0.620	0.603	0.780	0.760	0.720	**0.753**
Thoth-30B-A3B	**0.980**	0.920	**0.970**	0.957	0.770	0.790	0.750	0.770	0.600	0.520	0.600	0.573	**0.820**	0.720	0.680	0.740

and Partial results on AirQuality and matches the best Partial score on Space. Thoth-8B also remains competitive at a substantially smaller model scale, reaching average Binary and Partial scores of 0.25 and 0.69. The consistently lower Binary scores across models further illustrate the difficulty of satisfying all rubric criteria simultaneously.

Scenario Description: The observations come from CAMELS-US hydrometeorological time series. Each item contains two aligned daily records from one or two comparatively low-human-influence U.S. catchments. Daymet and NLDAS precipitation are alternative basin-mean gridded estimates rather than independent storms. Maximum temperature and shortwave radiation are atmospheric state or energy proxies; they do not directly observe soil moisture, snow storage, groundwater, or evapotranspiration. USGS discharge has been converted to basin-depth units of millimeters per day using catchment area, and a value of zero denotes recorded zero flow. The Sopchoppy, Merced, Justiceburg, and Sycamore records represent different hydroclimates, so a temporal pattern in one basin need not transfer to another. These observational series can support conclusions about order, persistence, state change, and association, but cannot by themselves identify a causal hydrologic mechanism.



Question: Does rainfall trigger active flow in the same way throughout this interval? Compare the clear early activations with the prolonged later zero-flow period in which rainfall still occurs, and state whether one stable activation rule is supported.

Rubrics:
R1: Hard failure if one stable precipitation threshold or activation rule is claimed to govern the full interval.
R2: The answer should recognize both the clear early flow activations and the prolonged later zero-flow regime despite later rainfall. Exact run length, counts, or boundary dates are not required.
R3: Credit requires at least one early responsive rainfall event and one later rainfall example with little or no flow response.
R4: Hard failure if a physical storage, infiltration, or moisture threshold is claimed as directly measured by these two series.

GPT-4o-mini: (Binary=0.0; Partial=0.5)
The observations suggest that rainfall does not trigger active flow uniformly throughout the specified interval. In the earlier part of the observed time series, there are clear activations of flow associated with rainfall events, as evidenced by the observed specific discharge values correlating with higher rainfall rates. For instance, on days when significant precipitation occurred, such as around May 5, 1999, there were corresponding increases in discharge. However, during the later period, particularly noted in June and July 1999, there were instances of rainfall that did not trigger any flow, leading to a prolonged zero-flow period despite ongoing rainfall events. This indicates that while rainfall initially resulted in flow activation, this relationship changed over time, likely due to factors such as saturation or other hydrological variables affecting discharge. Therefore, the data does not support one stable activation rule throughout the interval; instead, it reflects a changing relationship between rainfall and flow response.

Thoth-30B-A3B: (Binary=1.0; Partial=1.0)
Rainfall triggers active flow early (e.g., repeated discharge bursts following NLDAS precipitation events around mid-May through late-June), but later in the record (roughly late-August through mid-October) rainfall occurs while discharge remains near zero for an extended stretch. This suggests a shift in catchment response—early conditions support rapid runoff generation from storms, while later conditions do not—so a single stable rainfall-to-flow activation rule is not supported across the full interval.

Figure 13: A showcase from GPT-4o-mini and Thoth-30B-A3B in the KnoTS benchmark.

Table 7: Full results on the six KnoTS subsets. We report both *Binary* and *Partial* scores for each subset. *Avg* denotes the average performance over all six subsets.

Model	Finance		Weather		Power		Hydrology		Space		AirQuality		*Avg*	
	Binary	Partial	Binary	Partial	Binary	Partial	Binary	Partial	Binary	Partial	Binary	Partial	Binary	Partial
Proprietary Models														
GPT-4o-mini	0.29	0.66	0.46	0.81	0.46	0.79	0.40	0.76	0.16	0.64	0.32	0.74	0.35	0.73
Grok-4.1-Fast	0.31	0.70	0.26	0.67	0.33	0.76	0.50	0.82	0.16	0.66	0.29	0.72	0.31	0.72
Open-source Large Language Models														
Qwen3-235B-A22B	0.21	0.63	0.28	0.69	0.35	0.75	0.58	0.86	0.14	0.62	0.26	0.68	0.30	0.70
Qwen3-8B	0.21	0.68	0.28	0.72	0.22	0.66	0.15	0.64	0.00	0.51	0.26	0.71	0.19	0.65
Mistral-Small-24B	0.12	0.60	0.26	0.72	0.17	0.63	0.10	0.63	0.03	0.55	0.13	0.65	0.14	0.63
Llama-3.1-8B-Instruct	0.21	0.62	0.21	0.69	0.11	0.55	0.13	0.63	0.00	0.58	0.11	0.63	0.13	0.62
Qwen3-30B-A3B-Instruct	0.24	0.68	0.28	0.72	0.30	0.67	0.15	0.64	0.03	0.51	0.26	0.71	0.21	0.66
DeepSeek-R1-32B	0.26	0.64	0.21	0.72	0.20	0.64	0.13	0.64	0.08	0.52	0.18	0.66	0.18	0.64
Open-source Vision Language Models														
Qwen3-VL-30B-A3B-Instruct	0.19	0.68	0.23	0.67	0.09	0.63	0.15	0.64	0.08	0.53	0.26	0.70	0.17	0.64
Mistral-Small-3.2-24B	0.26	0.73	0.26	0.67	0.39	0.73	0.23	0.72	0.16	0.62	0.26	0.76	0.26	0.71
Task-Specific Time Series Language Models														
ChatTS-14B	0.12	0.57	0.15	0.61	0.07	0.43	0.13	0.63	0.05	0.49	0.03	0.51	0.09	0.54
Time-MQA (Llama-3-8B)	0.16	0.64	0.31	0.74	0.13	0.58	0.18	0.68	0.03	0.63	0.21	0.72	0.17	0.67
Time-MQA (Qwen2.5-7B)	0.14	0.60	0.18	0.70	0.17	0.57	0.20	0.66	0.03	0.59	0.18	0.64	0.15	0.63
Ours														
Thoth-30B-A3B	0.26	0.70	0.41	0.76	0.35	0.67	0.35	0.77	0.11	0.66	0.34	0.80	0.30	0.73
Thoth-8B	0.19	0.65	0.28	0.78	0.28	0.66	0.40	0.76	0.11	0.58	0.26	0.72	0.25	0.69