

# TS-Reasoner: Aligning Time Series Foundation Models with LLM Reasoning

Anonymous ACL submission

## Abstract

Time series reasoning is crucial to decision-making across domains like finance, energy, and scientific discovery. While existing time series foundation models (TSFMs) excel at capturing numerical dynamics, they lack the high-level contextual reasoning inherent in Large Language Models (LLMs). Conversely, without expensive post-training, LLMs often struggle with the numerical understanding of time series data. Although it is intuitive to integrate the two types of models, developing effective training recipes that align the two modalities for reasoning tasks is still an open challenge. To this end, we propose TS-REASONER that aligns the latent representations of TSFMs with the textual inputs of LLMs for downstream understanding/reasoning tasks. Specifically, we propose a simple yet effective method to curate diverse, synthetic pairs of time series and textual captions for alignment training. We then develop a two-stage training recipe that applies instruction finetuning after the alignment pretraining. Unlike existing works that train an LLM to take time series as inputs, we leverage a pretrained TSFM and freeze it during training. Extensive experiments on several benchmarks demonstrate that TS-REASONER not only outperforms a wide range of prevailing LLMs, Vision Language Models (VLMs), and Time Series LLMs, but also achieves this with remarkable data efficiency, e.g., using less than half the training data.

## 1 Introduction

Time series analysis has long been fundamental to various real-world applications in finance, energy, and other domains (Prakarsha and Sharma, 2022; Xu et al., 2023; Nie et al., 2024). While numerical data form the bedrock of time series analysis, human decision-making is often complemented by rich prior knowledge and qualitative contextual information, such as news articles and expert assessments. Existing analytical models lack this

integration, preventing a truly contextualized understanding. By bridging numerical patterns with contextual information, we can develop automated systems that provide deeper insights into complex phenomena.

Recent advances in Time Series Foundation Models (TSFMs) have significantly enhanced the understanding of time series data through large-scale pretraining. These models are capable of generalizing across a wide variety of time series tasks and domains. Although TSFMs (Goswami et al., 2024; Das et al., 2024) demonstrate strong modeling capabilities, most are pre-trained exclusively on unimodal numerical time series and cannot therefore comprehend or integrate textual information. On the other hand, large Language Models (LLMs) and Vision Language Models (VLMs) can take texts and images as input context, and have demonstrated remarkable reasoning and problem-solving abilities across various tasks (Wei et al., 2022; Yao et al., 2023; Hao et al., 2023; Yu et al., 2024; Ho et al., 2025), sparking interest in transferring their capabilities to time series analysis. Some studies (Gruber et al., 2023; Liu et al., 2024c; Jia et al., 2024) transform numerical time series into string form and perform time series forecasting on LLMs by prompting them with the strings. However, despite their strong reasoning abilities, LLMs struggle to capture temporal dependencies due to their inherent lack of temporal understanding (Fons et al., 2024; Merrill et al., 2024) and limited ability to interpret numerical values. These limitations hinder their understanding of time series data. As shown in the figure 1, TSFM and LLM have complementary strengths; the former specializes in temporal understanding, while the latter excels at text understanding. To combine the complementary strengths of TSFMs and LLMs while overcoming their respective limitations, we propose TS-REASONER, a Time Series Large Language Model (TSLLM) designed to enhance time



Figure 1: Time series forecasting vs. reasoning. Time series reasoning requires both contextual reasoning (e.g., news) by LLMs and numerical understanding by TSFM.

series reasoning by aligning a TSFM with an LLM. Specifically, we first employ the TSFM to extract rich temporal representations from numerical time series data. To effectively incorporate this temporal information into the LLM, TS-REASONER introduces a TS-to-Text adapter, which projects the TSFM-extracted temporal features into the LLM’s input embedding space. This enables seamless integration of the TSFM’s temporal understanding with the LLM’s powerful linguistic and reasoning capabilities. Our training framework consists of two stages: pretraining and fine-tuning. In the pretraining stage, we finetune TS-REASONER to produce textual captions of input time series and achieve a fundamental alignment. To this end, we propose a simple yet effective prompting strategy to curate high-quality captions for diverse time series data using advanced LLMs/VLMs. In the fine-tuning stage, we further enhance the model’s reasoning abilities through instruction tuning, ensuring robust performance in downstream tasks.

Our work formulates the connection between LLMs and TSFMs, facilitating time series reasoning through the integration of rich contextual information and LLM reasoning. Second, we address a critical data bottleneck by a simple yet effective time series captioning method, which diversifies the training data for aligning LLMs and TSFMs. Finally, we offer empirical insights into the strengths and limitations of existing approaches.

We evaluate the understanding and reasoning capabilities of TS-REASONER on two standard benchmarks: TimeSeriesExam (Cai et al., 2024a) and MTBench (Chen et al., 2025). TS-REASONER significantly outperforms a wide range of baseline models, including LLMs, VLMs, and the TSLLMs, as shown in Figure 1. Finally, comprehensive analyses and ablation studies validate the effectiveness of our key designs and establish the superiority of our model in generalization performance,

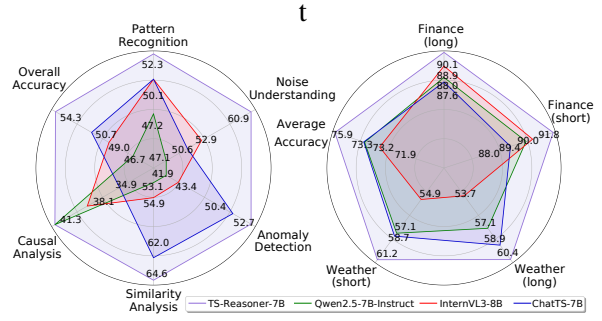


Table 1: Results on time series understanding and reasoning benchmarks. TS-REASONER demonstrates a consistent advantage over the prevailing LLMs, VLMs, and TSLLMs.

scalability, and training data efficiency.

## 2 Related Work

**LLMs for Time Series.** Shifting from traditional statistical (RB, 1990) and neural architectures (Liu et al., 2021; Lim et al., 2021; Wu et al., 2021; Zhou et al., 2022; Li et al., 2023b, 2024b), recent research adapts LLMs via prompting (Cao et al., 2023; Chuang\* et al., 2024), embedding alignment (Jin et al., 2023; Sun et al., 2023; Pan et al., 2024), reprogramming (Jin et al., 2023; Pillai et al., 2025) or vision-based modeling (Chen et al., 2024c; Zhong et al., 2025). Yet, given LLMs’ limitations in intrinsic temporal reasoning (Merrill et al., 2024), focus has turned to enhancing contextual understanding (Chow et al., 2024; Zhang et al., 2025a; Xie et al., 2024). Our framework targets deep semantic alignment to bridge the cognitive gap, empowering the model to ground the LLM’s reasoning in robust temporal features.

**Modality Alignment.** Modality alignment methods are widely studied in the multimodal domain (Li et al., 2022; Lai et al., 2024; Li et al., 2023a; Liu et al., 2024b). Inspired by the success of multimodal alignment, recent works treat time series as another modality and align it to the LLM (Xie et al., 2024; Zhang et al., 2025a). However, these approaches often rely on scratch-trained encoders and are restricted to narrow domains.

**Time Series Foundation Models.** Pre-training advancements have catalyzed the development of time series foundation models. While early works employed BERT-style masking (Zerveas et al., 2021; Nie et al., 2022), recent large-scale pre-training initiatives (Das et al., 2024; Liu et al., 2024e; Ekambaram et al., 2024; Ansari et al., 2024; Shi et al., 2024; Liu et al., 2024d; Qin et al., 2025;

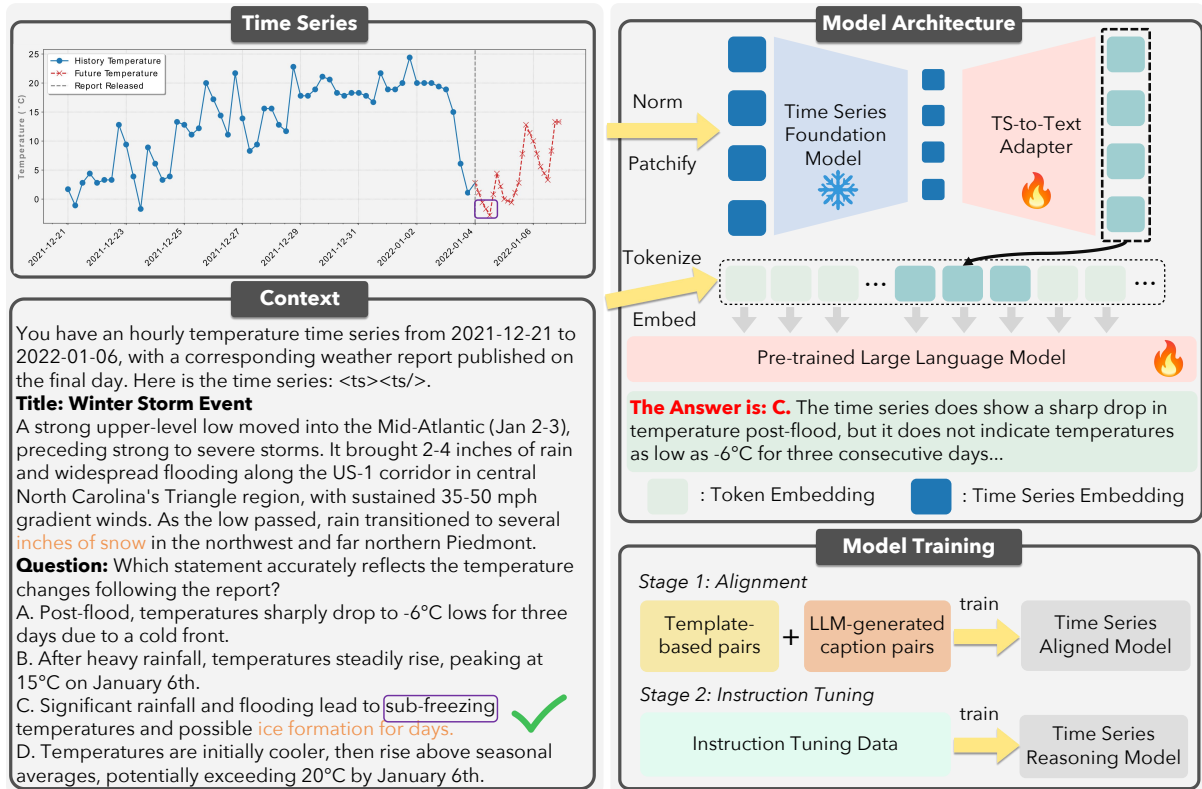


Figure 2: Overview of TS-REASONER architecture and training pipeline. Input time series are encoded by a pretrained TSFM, then mapped into the LLM’s embedding space via a trainable TS-to-Text Adapter. Training occurs in two stages: (1) alignment pretraining using code-synthesized and LLM-generated captions (§3.4), and (2) instruction-tuning to enhance complex reasoning capabilities.

Zhang et al., 2025b; Liu et al., 2025) demonstrate significant gains in forecasting performance. Beyond pure forecasting, MOMENT (Goswami et al., 2024) leverages T5 encoders for multi-task capabilities, while ChronoSteer (Wang et al., 2025a) explores alignment, utilizing LLM revisions to enhance model predictions.

### 3 TS-REASONER for Temporal Reasoning

#### 3.1 Problem Formulation

We formulate time series reasoning as a conditional generation task. Given a multimodal input comprising a natural language context  $\mathcal{X}$  (e.g., background knowledge or instructions) and a set of time series  $\mathcal{S} = \{\mathcal{T}_0, \dots, \mathcal{T}_K\}$ , the goal is to produce an output sequence  $\mathcal{V}$ . To explicitly capture the underlying rationale, we decompose  $\mathcal{V}$  into a reasoning path  $\mathcal{R}$  and a final answer  $\mathcal{A}$ , denoted as  $\mathcal{V} = [\mathcal{R}; \mathcal{A}]$ . Accordingly, the generation probability  $P(\mathcal{V}|\mathcal{X}, \mathcal{S})$  allows for a sequential factorization:

$$P(\mathcal{V}|\mathcal{X}, \mathcal{S}) = \underbrace{P(\mathcal{A}|\mathcal{R}, \mathcal{X}, \mathcal{S})}_{\text{Answer Generation}} \cdot \underbrace{P(\mathcal{R}|\mathcal{X}, \mathcal{S})}_{\text{Reasoning Process}} \quad (1)$$

#### 3.2 Overview

As illustrated in Figure 2, TS-REASONER is composed of (1) a pretrained TSFM to encode normalized, non-overlapping time series patches into embeddings; (2) a pretrained LLM, and (3) a TS-to-Text adapter that projects the TSFM features into the LLM’s input space. Time series features are interleaved with text token embeddings, demarcated by special token “ $\langle ts \rangle \langle ts / \rangle$ ”. Training follows a two stage pipeline: (1) Alignment Pretraining, utilizing LLM/VLM-synthesized captions to bridge TSFM and LLM modalities; and (2) Instruction Tuning, to refine complex reasoning for downstream tasks.

#### 3.3 Model Architecture

Consider a natural language context  $\mathcal{X}$  and a corresponding set of time series  $\mathcal{S} = \{\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_K\}$ , we first project both into a shared embedding space. Specifically, for each time series  $\mathcal{T}_i \in \mathbb{R}^{L_i}$ , where  $L_i$  is the length of the series, we first apply instance normalization to standardize its distribution to zero mean and unit variance, and incorporate these statistical features (mean and variance) into the natural language context  $\mathcal{X}$  in textual form. This prepro-

cessing step ensures that the model is robust to shifts and scales in the input data. Subsequently, we partition the normalized time series into a sequence of non-overlapping patches, each of a fixed length  $P$ . This patching strategy yields a sequence of  $N_i = \lfloor L_i/P \rfloor$  patches, transforming the time series into a tensor  $\mathcal{T}_i^p \in \mathbb{R}^{N_i \times P}$ . These patches are then encoded using the TSFM, which acts as our time series feature extractor. The TSFM processes the sequence of patches and produces a sequence of embedding vectors:

$$\mathcal{Z}_i^T = \text{TSFM}(\mathcal{T}_i^p) \in \mathbb{R}^{N_i \times d_{ts}}, \quad (2)$$

where  $d_{ts}$  denotes the dimension of the time series embeddings. Concurrently, the natural language context  $\mathcal{X}$  is tokenized and fed into the pre-trained LLM’s embedding layer. This process converts the textual input into a sequence of contextualized token embeddings:

$$\mathcal{Z}^L = \text{LLM}_{\text{embed}}(\mathcal{X}) \in \mathbb{R}^{M \times d_{\text{text}}}, \quad (3)$$

where  $M$  is the number of tokens in the instruction, and  $d_{\text{text}}$  is the dimensionality of the LLM’s hidden states. To align the dimension and semantics of embeddings between LLM and TSFM, we use a multilayer perceptron (MLP) as a TS-to-Text Adapter to transform the time series embedding into the text embedding space:

$$\mathcal{H}_i^T = \text{MLP}(\mathcal{Z}_i^T) \in \mathbb{R}^{N_i \times d_{\text{text}}}, \quad (4)$$

To form a unified input sequence for the LLM that accommodates multiple time series, we structure the natural language instruction  $\mathcal{X}$  to include  $K$  indicators,  $\{K \cdot \langle \text{ts} \rangle \langle \text{ts} / \rangle\}$ . The  $i$ -th placeholder  $\langle \text{ts} \rangle \langle \text{ts} / \rangle$  marks the insertion point for the corresponding  $i$ -th time series  $\mathcal{T}_i$ .

Let  $\{\mathcal{H}_i^T \in \mathbb{R}^{N_i \times d_{\text{text}}}\}_{i=1}^K$  be the set of projected time series embeddings. The final input sequence  $H$  is constructed by sequentially inserting the embedding to each  $\langle \text{ts} \rangle \langle \text{ts} / \rangle$  with its corresponding time series embedding sequence  $\mathcal{H}_i^T$ . This substitution process results in a composite sequence where language and time series representations are interleaved. The total length of this fused sequence is  $M + \sum_{i=1}^K N_i$ . The final tensor fed to the LLM’s transformer layers is therefore:  $\mathcal{H} \in \mathbb{R}^{(M + \sum_{i=1}^K N_i) \times d_{\text{text}}}$ . This strategy enables the LLM to process multiple, arbitrarily placed time series within a single, coherent context and capture complex inter-series and text-series dependencies. After the combination, the input embedding  $H$  is fed to LLM to produce the final prediction  $\mathcal{Y}$ .

### 3.4 Training Recipe

Our training process consists of two sequential stages: the first stage aligns time series data with the LLM to establish a foundational understanding of temporal-textual relationships, while the second stage refines the LLM’s reasoning capabilities to interpret and analyze these aligned representations. Throughout both stages, we keep the parameters of the TSFM frozen to preserve its pretrained temporal knowledge, while allowing the LLM’s parameters to remain trainable, ensuring adaptive learning without compromising the integrity of the encoded time series features.

**Stage 1: Pre-training for Language-Timeseries Alignment.** In this stage, our primary objective is to align temporal data with textual information. We initially leverage synthesized data from (Xie et al., 2024), which provides predefined templates to describe time series attributes. However, while this template-based data offers accurate numerical information, its focus on specific time series patterns limits diversity, and the caption structure is monotonous. This lack of diversity can lead to overfitting to the templates, encouraging the model to learn shallow patterns and resulting in poor generalization ability (Dong et al., 2025; Choi et al., 2024). To alleviate this problem, we draw inspiration from captioning techniques in multimodal LLMs (Chen et al., 2024a). We synthesize comprehensive captions using advanced LLMs (e.g., GPT-4.1) to enrich our alignment data. Specifically, we collect time series from two sources: (Merrill et al., 2024), which includes contextual information, and synthetic data from Chronos (Ansari et al., 2024), which provides pure numerical time series.

**Attribute-aware Captioning.** Caption generation has been extensively investigated in visual domains (Cheng et al., 2023, 2025; Chen et al., 2024b), playing a crucial role in multimodal alignment. However, time series captioning remains largely underexplored, presenting a significant impediment to achieving comprehensive alignment. To address this gap, we introduce a straightforward approach for generating scalable time series captions, as shown in Figure 3.

Given a time series  $\mathcal{T}$  with a temporal context  $\mathcal{C}$ , we begin by defining a fundamental captioning instruction, denoted as  $\mathcal{I}_{\text{base}}$ . To facilitate enhanced comprehension by LLMs, we transform the time series into an image plot via Python code,

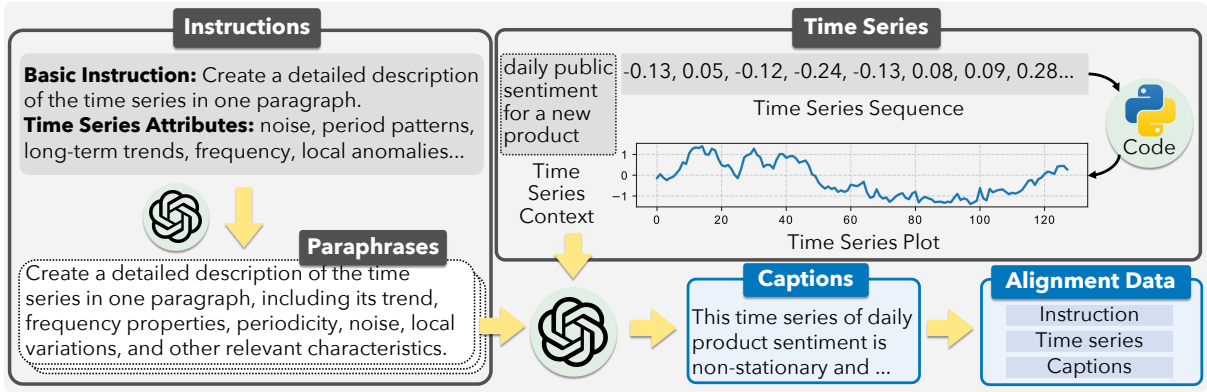


Figure 3: Workflow for attribute-aware caption synthesis. This process enriches basic instructions with key attributes and generates diverse paraphrases to produce high-fidelity captions for Stage 1 alignment.

$I_{TS} = \Phi(\mathcal{T})$ . As evidenced in Table 2 (Section 4.1), presenting the time series as an image to advanced LLMs (e.g., GPT-4.1) demonstrates a substantial advantage in understanding compared to providing it as a raw numerical series.

To enrich the generated captions, we first identify a set of  $G$  pertinent attributes of the time series, denoted as  $\{a_1, a_2, \dots, a_G\}$  (e.g., trend, frequency, periodicity, noise, local variations). These attributes are then incorporated into the basic instruction, yielding an augmented instruction  $\mathcal{I}' = \mathcal{I}_{\text{base}} \cup \{a_1, a_2, \dots, a_G\}$ . To further promote caption diversity, we leverage the LLM to paraphrase  $\mathcal{I}'$  into  $R$  distinct instructions, forming a candidate set of prompts  $\mathcal{P} = \{\mathcal{I}'_1, \mathcal{I}'_2, \dots, \mathcal{I}'_R\}$ . For each time series  $\mathcal{T}$ , a single prompt  $\mathcal{I}''$  is uniformly sampled from this set. The final caption is then generated conditioned on the sampled prompt and the time series visualization:

$$\text{Caption} = \text{LLM}(\mathcal{I}'', I_{TS}), \quad (5)$$

where  $\mathcal{I}'' \sim \mathcal{U}(\mathcal{P})$ . The prompts are shown in the Figure 10 in Appendix F. We randomly sample 10K time series from each of two distinct sources: the Chronos synthetic dataset (Ansari et al., 2024), which contains purely numerical time series, and a dataset of text-attributed time series from Merrill et al. (Merrill et al., 2024), which provides contextual backgrounds. The construction of data offers two benefits: (1) Pure time series data enables the model to build a foundational understanding of temporal patterns by focusing solely on the intrinsic characteristics of the data. (2) Context-augmented time series enhances domain-specific comprehension by linking numerical trends to real-world scenarios, thereby improving the model’s ability to generalize across diverse applications.

**Stage 2: Instruction Finetuning for Time Series Reasoning.** To elevate the model’s capabilities from foundational understanding to complex reasoning, we employ an instruction fine-tuning stage based on the instruction tuning dataset (Xie et al., 2024), which encompasses a wide range of Q&As and instruction-following tasks. This training facilitates TS-REASONER with two critical abilities: the fidelity to adhere to complex instructions and structured response formats, and the capacity for nuanced, context-driven reasoning on time series-specific queries.

## 4 Experiments

**Datasets.** Our evaluation incorporates TimeSeriesExam (Cai et al., 2024a), which systematically evaluates time series understanding and reasoning abilities across several key aspects: Pattern Recognition (PR), Noise Understanding (NU), Anomaly Detection (AD), Similarity Analysis (SA), and Causality Analysis (CA). See Table 8 for question examples. We also evaluate on MTBench (Chen et al., 2025), a benchmark for evaluating time series reasoning in financial and weather domains, featuring questions that span both short-term (7-day) and long-term (14-day) temporal horizons. An additional experiment on open-ended inductive reasoning is reported in Appendix A.

**Baselines and Evaluation Metrics.** We compare our method against three types of baselines: closed-source LLMs / VLMs, open-source LLMs / VLMs, and TSLLMs. Specifically, for closed-source models, we include GPT-4o, GPT-4.1 (gpt, 2024; Achiam et al., 2023), Claude-Sonnet-3.7 (The), and DeepSeek-Chat (Liu et al., 2024a). For open source LLM, we evalu-

ated Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen-2.5-7B-Instruct (Yang et al., 2024), GLM-4-9B-Chat (GLM et al., 2024), InternLM3-8B-Instruct (Cai et al., 2024b), and Ministral-8B-Instruct (Jiang et al., 2024). Time series are transformed into textual sequences of numbers for LLMs. For open-source VLM models, we compare Qwen2.5-VL-7B (Bai et al., 2025), Phi-4-Multimodal-Instruct (Abouelenin et al., 2025), Llama3-LLaVA-Next-8B (Li et al., 2024a), InternVL3-8B (Zhu et al., 2025), and MiniCPM-V-2.6 (Yao et al., 2024). Time series are transformed into plots via code for VLMs. For TSLLMs models, we compare with ChatTime-7B (Wang et al., 2025b), ChatTS-14B (Xie et al., 2024), and we use the official training data and code to fine-tune a 7B model for a fair comparison. As all benchmarks are multiple-choice Q&As, we use accuracy as the evaluation metric and run experiments for three repetitions. See more implementation details in Appendix B.

## 4.1 Main Results

Table 2 presents the performance of all models on the two benchmarks. The best results are in bold, and the second-best results are underlined. Based on the results, we have the following key observations:

(i) **TS-REASONER demonstrates superior performance among same-sized models.** TS-REASONER demonstrates outperforms same-sized models on TimeSeriesExam, surpassing the best-performing best LLM, VLM, and TSLLM by 7.60%, 5.25%, and 3.54%, respectively. Compared to the backbone model, TS-REASONER improves on our backbone LLM performance by a substantial 16.29%. On MTBench, TS-REASONER exceeds the top baseline by approximately 2.7%. These results confirm the efficacy of integrating TSFM temporal features into LLMs.

(ii) **TS-REASONER shows consistent gains across time series reasoning subtasks.** TS-REASONER consistently leads across reasoning tasks, with absolute improvements over the second-best baseline in multiple subtasks like *Pattern Recognition* (2.16%), *Noise Understanding*(8.05%). It also gains 2% in financial and weather reasoning. Such improvements stem from robust alignment and the enhanced ability to reason over numerical patterns within textual contexts.

## 4.2 Analysis of Data Scaling and Efficiency

Figure 4 presents our data scaling analysis on the TimeSeriesExam benchmark. TS-REASONER demonstrates superior data efficiency compared to the ChatTS-7B baseline. In the alignment stage, our model achieves higher accuracy using only 60K samples, less than half the baseline requirement. This advantage is even more pronounced in instruction tuning, where 10K samples suffice to outperform ChatTS-7B. This efficiency stems from leveraging a pretrained TSFM for a robust temporal foundation, enabling advanced reasoning with minimal labeled data

## 4.3 Choices of Captioning Model for Alignment

To assess the impact of caption quality on alignment efficacy, we trained TS-REASONER using captions from three models with varying capabilities: GPT-4.1, and two VLMs, InternVL3-8B and Qwen2.5-VL-7B-Instruct. As illustrated in Figure 6, the results demonstrate that the performance of TS-REASONER is directly correlated with the fidelity of the captioning model. The model trained on GPT-4.1 captions consistently outperforms the one trained on InternVL3-8B captions, which in turn surpasses the one trained on Qwen2.5-VL-7B-Instruct captions. The higher performance gain from GPT-4.1 is attributed to its advanced capability in time series understanding. It is not surprising that the captions generated by InternVL3-8B achieve higher performance than Qwen2.5-VL-7B-Instruct, as its better time series understanding capability is shown in Table 2.

## 4.4 Choices of TSFM and LLM in TS-REASONER

**Different choices of TSFMs.** To investigate the performance of TS-REASONER with different TSFMs, we replaced TimesFM (200M) with MOMENT-1-base (200M) and Chronos-base (200M), TSFMs of the same size, and re-evaluated its performance on the TimeSeriesExam benchmark. Results presented in Table 3 reveal a substantial performance degradation when using MOMENT and Chronos, with overall accuracy falling from 54.26% to 45.74% and 53.21% respectively. These results align with established forecasting benchmarks (Shi et al., 2024; Mulayim et al., 2024) where TimesFM demonstrates higher fidelity. This suggests that TimesFM provides better time series

Model	TimeSeriesExam (Cai et al., 2024a)						MTBench (Chen et al., 2025)			
	PR	NU	AD	SA	CA	OA	Finance (long)	Finance (short)	Weather (long)	Weather (short)
<i>Proprietary models</i>										
DeepSeek-Chat	65.23	55.17	52.71	63.71	42.86	59.89	89.15	90.02	59.75	58.76
DeepSeek-R1	74.66	63.22	63.56	65.49	41.27	67.36	65.31	60.69	49.45	46.36
Claude-Sonnet-3.7	62.26	55.17	48.06	72.57	50.79	59.63	84.11	88.56	51.24	47.91
GPT-4o	59.03	55.17	53.49	62.83	31.75	55.96	84.30	82.69	48.07	48.22
GPT-4o (vision)	67.12	62.07	62.79	64.60	26.98	62.12	84.11	80.65	46.43	48.53
GPT-4.1 (vision)	69.81	68.97	68.22	75.22	41.27	67.89	93.41	91.45	56.04	55.35
<i>Open-source Large Language Models</i>										
Llama-3.1-8B-Instruct	37.73	37.93	30.23	36.28	28.57	35.52	63.37	35.52	40.25	40.00
Qwen2.5-7B-Instruct	47.17	47.13	41.86	53.10	41.27	46.66	87.98	89.41	57.14	58.44
GLM-4-9B-chat	41.78	39.08	37.21	47.79	38.09	41.28	71.31	77.19	50.27	50.85
InternLM3-8B-Instruct	43.93	51.72	26.35	52.21	34.92	42.33	71.70	71.08	45.05	46.67
Minstral-8B-Instruct	43.13	37.93	39.53	44.25	36.51	41.55	46.32	50.71	39.15	40.93
<i>Open-source Vision Language Models</i>										
Qwen2.5-VL-7B-Instruct	25.34	32.18	19.38	42.48	12.70	26.61	81.98	86.35	52.06	46.82
Phi-4-Multimodal-Instruct	36.39	34.48	30.23	38.94	14.28	33.68	70.35	74.54	48.35	49.77
Llama3-LLaVA-Next-8B	31.27	35.63	29.46	30.09	38.09	31.85	52.14	51.50	47.53	47.29
InternVL3-8B	50.13	52.87	43.41	54.87	38.09	49.01	88.95	90.00	53.71	54.88
MiniCPM-V-2.6	29.11	39.08	27.13	51.33	31.75	33.42	81.78	83.09	48.63	45.12
<i>Time Series Large Language Models</i>										
ChatTime-7B	42.85	49.42	35.65	44.24	34.92	41.94	25.97	28.10	47.80	42.79
ChatTS-7B	50.13	50.57	50.38	61.95	34.92	50.72	87.60	88.01	58.92	58.75
ChatTS-14B*	59.30	54.02	51.16	62.83	41.27	56.36	89.22	91.22	59.61	59.22
TS-REASONER-7B (ours)	<b>53.46</b> $\pm 1.58$	<b>60.15</b> $\pm 3.51$	<b>53.23</b> $\pm 1.61$	<b>63.42</b> $\pm 1.02$	<b>43.39</b> $\pm 2.42$	<b>54.83</b> $\pm 0.98$	<b>92.00</b> $\pm 1.74$	<b>93.28</b> $\pm 1.28$	<b>60.44</b> $\pm 0.14$	<b>61.55</b> $\pm 0.31$
$\Delta$ Over Best 7B	+3.33	+7.28	+2.85	+1.47	+2.12	+4.11	+3.05	+3.28	+1.52	+2.80

Table 2: Performance of LLMs, VLMs, TSLLMs, and proprietary models on time series understanding and reasoning benchmarks. Our baselines also include ChatTS-14B, which uses a larger base model.

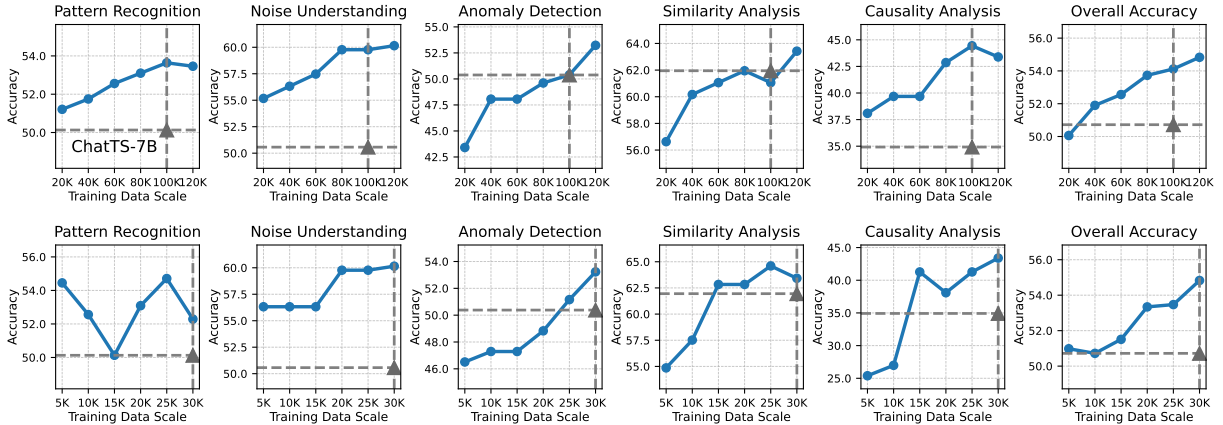


Figure 4: Data scaling and efficiency of TS-REASONER. The top (bottom) row depicts the performance of TS-REASONER when increasing the training data for alignment (instruction tuning). The columns correspond to sub-tasks in TimeSeriesExam. ChatTS-7B (Xie et al., 2024) is included for reference, denoted by the gray triangle.

Table 3: Comparison of TS-REASONER using different TSFMs on the TimeSeriesExam benchmark.

Model	PR	NU	AD	SA	CA	OA
MOMENT	46.90	47.13	41.86	54.87	28.57	45.74
Chronos	51.75	59.77	51.93	<b>63.71</b>	36.51	53.21
TimesFM	<b>53.46</b>	<b>60.15</b>	<b>53.23</b>	63.42	<b>43.39</b>	<b>54.83</b>

features, enabling TS-REASONER to better understand and reason on time series.

**Different choices of LLMs.** To investigate the scalability and robustness of our approach with different LLM backbones, we evaluate TS-REASONER against across three distinct sizes of the Qwen-2.5-Instruct: 0.5B, 3B, and 7B. The results,

shown in Figure 5, confirm that TS-REASONER is both highly effective and robustly scalable. We observe a clear positive scaling law for both TS-REASONER and a baseline. More importantly, TS-REASONER maintains a consistent and significant lead across all models. This demonstrates that our approach performs robustly across different LLM backbones for complex time series reasoning.

#### 4.5 Ablation Studies

Table 4 validates the contribution of each component in TS-REASONER. We summarize key findings as follows:

**Significance of Attribute-aware Captioning.** Alignment data quality and quantity are critical. Re-

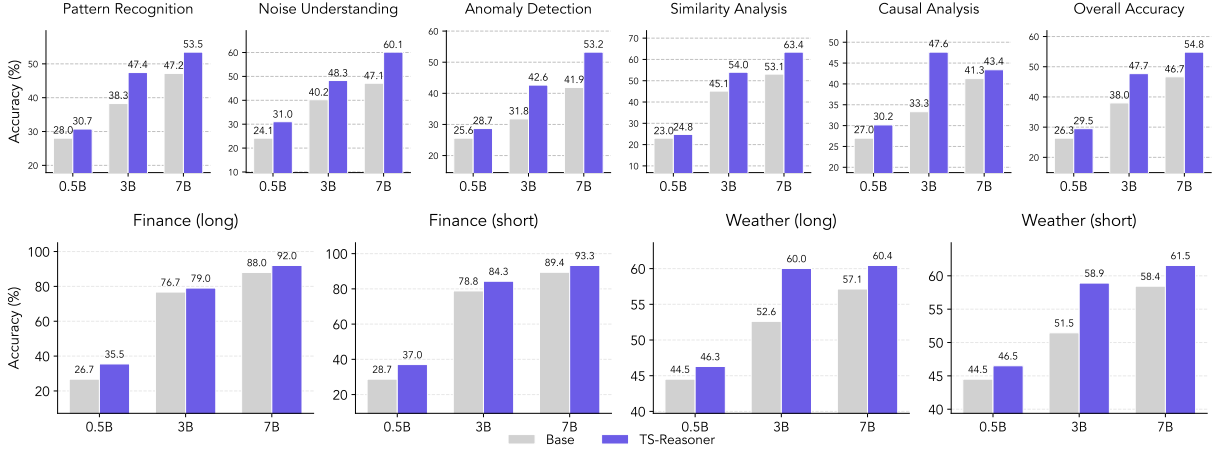


Figure 5: Performance of TS-REASONER and their associated LLM backbones (Qwen2.5 series). The top row and bottom row report the performance on TimeSeriesExam and MTBench, respectively.

Model	TimeSeriesExam (Cai et al., 2024a)						MTBench (Chen et al., 2025)			
	PR	NU	AD	SA	CA	OA	Finance (long)	Finance (short)	Weather (long)	Weather (short)
TS-REASONER-7B (ours)	<b>53.46</b>	<b>60.15</b>	<b>53.23</b>	<b>63.42</b>	<b>43.39</b>	<b>54.83</b>	<b>92.00</b>	<b>93.28</b>	<b>60.44</b>	<b>61.55</b>
<i>Ablation on Training Data</i>										
- LLM-caption	51.21	56.32	52.71	56.54	36.51	51.25	88.67	89.40	58.24	59.69
- Attributes	52.02	57.47	48.83	62.83	39.68	52.69	89.71	89.20	57.28	59.07
<i>Ablation on Training Stages</i>										
- Stage 1	47.98	54.02	37.98	57.52	30.16	46.92	80.24	83.71	52.88	55.34
- Stage 2	33.42	28.73	13.95	25.67	1.59	25.81	88.07	86.76	56.86	58.60
<i>Ablation on Model Architecture</i>										
- TSFM	51.48	52.87	51.16	63.71	38.09	51.76	89.43	89.70	58.65	60.62
- Freeze	51.48	56.32	53.48	60.18	41.27	52.81	89.15	91.02	58.24	60.15

Table 4: Ablation study results of different components in TS-REASONER.

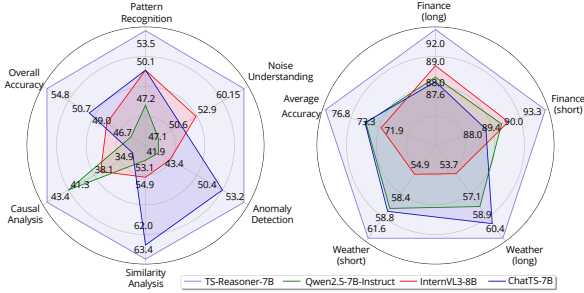


Figure 6: Comparison of VLMs used for time series captions to train TS-REASONER. **Left:** performance on TimeSeriesExam. **Right:** Performance on MTBench.

moving the captioning data or stripping attributes from instructions degrades performance by up to 3.01% on TimeSeriesExam and  $\sim 2\%$  on MTBench. This confirms that fine-grained linguistic descriptions are essential for capturing nuanced temporal patterns. Detailed qualitative and quantitative analyses of the diversity of captions are provided in Appendix D and E.

**Necessity of Two-Stage Training.** Eliminating either stage harms results. Removing Stage 1 (Alignment) primarily impacts MTBench’s cross-modal

reasoning tasks, while omitting Stage 2 (Instruction Tuning) causes a 28.45% drop on TimeSeriesExam. This suggests that while alignment grounds the model, instruction tuning is essential for activating the ability to follow specific analytical commands. **Efficacy of Pretrained TSFM.** Both replacing the frozen TSFM with a linear projection of time series patches and fine-tune TSFM result in consistent performance degradation. This underscores the value of pretrained knowledge in TSFMs to extract robust temporal features.

## 5 Conclusion

We introduce TS-REASONER, a framework that advances the ability of LLMs to understand and reason about time series via bridging with the TSFM. To mitigate the intrinsic semantic gap, we further developed an attribute-aware captioning method that enriches time-series alignment data, fostering a more robust alignment. Extensive experiments demonstrate that TS-REASONER substantially outperforms a wide range of baselines on time series understanding and reasoning benchmarks.

528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
  
538  
539  
540  
541  
542  
  
543  
544  
545  
546  
547  
548  
  
549  
550  
551  
552  
  
553  
554  
555  
  
556  
557  
558  
559  
  
560  
561  
562  
563  
  
564  
565  
566  
567  
568  
  
569  
570  
571  
572  
573  
574  
  
575  
576  
577  
578  
579

## References

The claude 3 model family: Opus, sonnet, haiku.

2024. [Openai gpt-4o](#).

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

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

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, and 1 others. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yves Bestgen. 2024. Measuring lexical diversity in texts: The twofold length problem. *Language Learning*, 74(3):638–671.

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

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, and 1 others. 2024b. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Defu Cao, Furong Jia, Serkan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassioulas, Yifeng Gao, and Rex Ying. 2025. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024a. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, and 1 others. 2024b. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495.

Mouxian Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. 2024c. Visions: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*.

Kanzhi Cheng, Wenpo Song, Jiabin Fan, Zheng Ma, Qiushi Sun, Fangzhi Xu, Chenyang Yan, Nuo Chen, Jianbing Zhang, and Jiajun Chen. 2025. Caparena: Benchmarking and analyzing detailed image captioning in the llm era. *arXiv preprint arXiv:2503.12329*.

Kanzhi Cheng, Wenpo Song, Zheng Ma, Wenhao Zhu, Zixuan Zhu, and Jianbing Zhang. 2023. Beyond generic: Enhancing image captioning with real-world knowledge using vision-language pre-training model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5038–5047.

Juhwan Choi, Junehyoung Kwon, JungMin Yun, Seung-guk Yu, and YoungBin Kim. 2024. Voldoger: Llm-assisted datasets for domain generalization in vision-language tasks. *arXiv preprint arXiv:2407.19795*.

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

Yu-Neng Chuang\*, Songchen Li\*, Jiayi Yuan\*, Guanchu Wang\*, Kwei-Herng Lai\*, Leisheng Yu, Sirui Ding, Chia-Yuan Chang, Qiaoyu Tan, Daochen Zha, and Xia Hu. 2024. Understanding different design choices in training large time series models. *arXiv preprint arXiv:2406.14045*.

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

Hao Dong, Moru Liu, Kaiyang Zhou, Eleni Chatzi, Juho Kannala, Cyrill Stachniss, and Olga Fink. 2025. Advances in multimodal adaptation and generalization: From traditional approaches to foundation models. *arXiv preprint arXiv:2501.18592*.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. 2024. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chap-*

580  
581  
582  
583  
584  
585  
  
586  
587  
588  
589  
590  
  
591  
592  
593  
594  
595  
  
596  
597  
598  
599  
600  
601  
  
602  
603  
604  
605  
  
606  
607  
608  
609  
  
610  
611  
612  
613  
614  
615  
  
616  
617  
618  
619  
  
620  
621  
622  
623  
624  
  
625  
626  
627  
628  
629  
630  
631  
  
632  
633  
634  
635

636			691
637		<i>ter of the Association for Computational Linguistics: System Demonstrations</i> , pages 150–158.	692
638	Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetenko. 2024. Evaluating large language models on time series feature understanding: A comprehensive taxonomy and benchmark. <i>arXiv preprint arXiv:2404.16563</i> .		693
644	Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. <i>arXiv preprint arXiv:2406.12793</i> .		694
649	Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models. In <i>International Conference on Machine Learning</i> .		695
650			696
651			697
652			698
653			699
654	Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. <i>arXiv preprint arXiv:2407.21783</i> .		700
655			701
656			702
657			703
658			704
659	Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2023. Large language models are zero-shot time series forecasters. <i>Advances in Neural Information Processing Systems</i> , 36:19622–19635.		705
660			706
661			707
662			708
663	Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. <i>arXiv preprint arXiv:2305.14992</i> .		709
664			710
665			711
666			712
667	Matthew Ho, Chen Si, Zhaoxiang Feng, Fangxu Yu, Zhijian Liu, Zhiting Hu, and Lianhui Qin. 2025. Arcmemo: Abstract reasoning composition with lifelong llm memory. <i>arXiv preprint arXiv:2509.04439</i> .		713
668			714
669			715
670			716
671	Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. 2024. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> , volume 38, pages 23343–23351.		717
672			718
673			719
674			720
675			721
676	Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. <i>arXiv preprint arXiv:2401.04088</i> .		722
677			723
678			724
679			725
680			726
681			727
682	Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and 1 others. 2023. Time-llm: Time series forecasting by reprogramming large language models. <i>arXiv preprint arXiv:2310.01728</i> .		728
683			729
684			730
685			731
686			732
687			733
688	Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, and		734
689			735
690			736
			737
			738
			739
			740
			741
			742
			743
			744
			745
			746
			747
			748
			749
			750
			751
			752
			753
			754
			755
			756
			757
			758
			759
			760
			761
			762
			763
			764
			765
			766
			767
			768
			769
			770
			771
			772
			773
			774
			775
			776
			777
			778
			779
			780
			781
			782
			783
			784
			785
			786
			787
			788
			789
			790
			791
			792
			793
			794
			795
			796
			797
			798
			799
			800

747	Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin	Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou	803
748	Yang, Xiangdong Huang, Jianmin Wang, and Ming-	Ye, Qingsong Wen, and Ming Jin. 2024. Time-moe:	804
749	sheng Long. 2025. Sundial: A family of highly ca-	Billion-scale time series foundation models with mix-	805
750	pable time series foundation models. <i>arXiv preprint</i>	ture of experts. <i>arXiv preprint arXiv:2409.16040</i> .	806
751	<i>arXiv:2502.00816</i> .		
752	Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong	Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong.	807
753	Huang, Jianmin Wang, and Mingsheng Long.	2023. Test: Text prototype aligned embedding to	808
754	2024e. Timer: Generative pre-trained transform-	activate llm’s ability for time series. <i>arXiv preprint</i>	809
755	ers are large time series models. <i>arXiv preprint</i>	<i>arXiv:2308.08241</i> .	810
756	<i>arXiv:2402.02368</i> .		
757	Mike A Merrill, Mingtian Tan, Vinayak Gupta, Tom	Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob	811
758	Hartvigsen, and Tim Althoff. 2024. Language mod-	Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz	812
759	els still struggle to zero-shot reason about time series.	Kaiser, and Illia Polosukhin. 2017. Attention is all	813
760	<i>arXiv preprint arXiv:2404.11757</i> .	you need. <i>Advances in neural information processing</i>	814
		<i>systems</i> , 30.	815
761	Ozan Baris Mulayim, Pengrui Quan, Liying Han, Xi-	Chengsen Wang, Qi Qi, Zhongwen Rao, Lujia Pan,	816
762	aomin Ouyang, Dezhi Hong, Mario Bergés, and Mani	Jingyu Wang, and Jianxin Liao. 2025a. Chronos-	817
763	Srivastava. 2024. Are time series foundation models	teer: Bridging large language model and time series	818
764	ready to revolutionize predictive building analytics?	foundation model via synthetic data. <i>arXiv preprint</i>	819
765	In <i>Proceedings of the 11th ACM International Con-</i>	<i>arXiv:2505.10083</i> .	820
766	<i>ference on Systems for Energy-Efficient Buildings,</i>	Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun,	821
767	<i>Cities, and Transportation</i> , pages 169–173.	Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin	822
768	Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mul-	Liao. 2025b. Chattime: A unified multimodal time	823
769	vey, H Vincent Poor, Qingsong Wen, and Stefan	series foundation model bridging numerical and tex-	824
770	Zohren. 2024. A survey of large language models	tual data. In <i>Proceedings of the AAAI Conference</i>	825
771	for financial applications: Progress, prospects and	<i>on Artificial Intelligence</i> , volume 39, pages 12694–	826
772	challenges. <i>arXiv preprint arXiv:2406.11903</i> .	12702.	827
773	Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	828
774	Jayant Kalagnanam. 2022. A time series is worth	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,	829
775	64 words: Long-term forecasting with transformers.	and 1 others. 2022. Chain-of-thought prompting elic-	830
776	<i>arXiv preprint arXiv:2211.14730</i> .	its reasoning in large language models. <i>Advances</i>	831
777	Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schnei-	<i>in neural information processing systems</i> , 35:24824–	832
778	der, Yuriy Nevmyvaka, and Dongjin Song. 2024.	24837.	833
779	<i>s<sup>2</sup>IP-LLM: Semantic space informed prompt learn-</i>	Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng	834
780	<i>ing with LLM for time series forecasting</i> . In <i>Proceed-</i>	Long. 2021. Autoformer: Decomposition transform-	835
781	<i>ings of the 41st International Conference on Machine</i>	ers with auto-correlation for long-term series fore-	836
782	<i>Learning</i> , volume 235 of <i>Proceedings of Machine</i>	casting. <i>Advances in Neural Information Processing</i>	837
783	<i>Learning Research</i> , pages 39135–39153. PMLR.	<i>Systems</i> , 34:22419–22430.	838
784	Arvind Pillai, Dimitris Spathis, Subigya Nepal,	Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen,	839
785	Amanda C Collins, Daniel M Mackin, Michael V	Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei.	840
786	Heinz, Tess Z Griffin, Nicholas C Jacobson, and An-	2024. Chatts: Aligning time series with llms via syn-	841
787	drew Campbell. 2025. Time2lang: Bridging time-	thetic data for enhanced understanding and reasoning.	842
788	series foundation models and large language models	<i>arXiv preprint arXiv:2412.03104</i> .	843
789	for health sensing beyond prompting. <i>arXiv preprint</i>	Jingwen Xu, Fei Lyu, and Pong C Yuen. 2023. Density-	844
790	<i>arXiv:2502.07608</i> .	aware temporal attentive step-wise diffusion model	845
791	Kandukuri Ratna Prakarsha and Gaurav Sharma. 2022.	for medical time series imputation. In <i>Proceedings of</i>	846
792	Time series signal forecasting using artificial neural	<i>the 32nd ACM International Conference on Informa-</i>	847
793	networks: An application on ecg signal. <i>Biomedical</i>	<i>tion and Knowledge Management</i> , pages 2836–2845.	848
794	<i>Signal Processing and Control</i> , 76:103705.	An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui,	849
795	Guo Qin, Zhi Chen, Yong Liu, Zhiyuan Shi, Haixuan	Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu,	850
796	Liu, Xiangdong Huang, Jianmin Wang, and Ming-	Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.	851
797	sheng Long. 2025. Cora: Covariate-aware adaptation	5 technical report. <i>arXiv preprint arXiv:2412.15115</i> .	852
798	of time series foundation models. <i>arXiv preprint</i>	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran,	853
799	<i>arXiv:2510.12681</i> .	Tom Griffiths, Yuan Cao, and Karthik Narasimhan.	854
800	CLEVELAND RB. 1990. Stl: A seasonal-trend de-	2023. Tree of thoughts: Deliberate problem solving	855
801	composition procedure based on loess. <i>J Off Stat</i> ,	with large language models. <i>Advances in neural</i>	856
802	6:3–73.	<i>information processing systems</i> , 36:11809–11822.	857

858 Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo  
859 Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin  
860 Zhao, Zhihui He, and 1 others. 2024. Minicpm-v:  
861 A gpt-4v level mllm on your phone. *arXiv preprint*  
862 *arXiv:2408.01800*.

863 Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and  
864 Lianhui Qin. 2024. Flow of reasoning: Training llms  
865 for divergent problem solving with minimal exam-  
866 ples. *arXiv preprint arXiv:2406.05673*.

867 George Zerveas, Srideepika Jayaraman, Dhaval Patel,  
868 Anuradha Bhamidipaty, and Carsten Eickhoff. 2021.  
869 A transformer-based framework for multivariate time  
870 series representation learning. In *Proceedings of*  
871 *the 27th ACM SIGKDD conference on knowledge*  
872 *discovery & data mining*, pages 2114–2124.

873 Haochuan Zhang, Chunhua Yang, Jie Han, Liyang Qin,  
874 and Xiaoli Wang. 2025a. Tempogpt: Enhancing tem-  
875 poral reasoning via quantizing embedding. *arXiv*  
876 *preprint arXiv:2501.07335*.

877 Haoran Zhang, Yong Liu, Yunzhong Qiu, Haixuan  
878 Liu, Zhongyi Pei, Jianmin Wang, and Mingsheng  
879 Long. 2025b. Timesbert: A bert-style foundation  
880 model for time series understanding. *arXiv preprint*  
881 *arXiv:2502.21245*.

882 Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qing-  
883 song Wen, and Yuxuan Liang. 2025. Time-vlm:  
884 Exploring multimodal vision-language models for  
885 augmented time series forecasting. *arXiv preprint*  
886 *arXiv:2502.04395*.

887 Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang,  
888 Liang Sun, and Rong Jin. 2022. Fedformer:  
889 Frequency enhanced decomposed transformer for  
890 long-term series forecasting. *arXiv preprint*  
891 *arXiv:2201.12740*.

892 Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu,  
893 Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan,  
894 Weijie Su, Jie Shao, and 1 others. 2025. Internv13:  
895 Exploring advanced training and test-time recipes  
896 for open-source multimodal models. *arXiv preprint*  
897 *arXiv:2504.10479*.

898 Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan  
899 Zhang, Jun Wang, and Yong Yu. 2018. Tegygen: A  
900 benchmarking platform for text generation models.  
901 In *The 41st international ACM SIGIR conference*  
902 *on research & development in information retrieval*,  
903 pages 1097–1100.

Table 5: Results on time series inductive reasoning.

Model	Accuracy
GPT-4o-mini	33.30
GPT-4o	33.60
Qwen2.5-14B	18.40
GPT-4o-mini (vision)	32.30
GPT-4o (vision)	32.20
ChatTS-14B	51.80
TS-REASONER-7B	<b>54.70</b>

## A Experiments on Open-Ended Time Series Reasoning

To investigate TS-REASONER on open-ended time series reasoning tasks, we evaluate on the Inductive Reasoning dataset (Xie et al., 2024), which requires the model to summarize the physical reasoning reflected by a uni/multivariate time series. The performance are evaluated using RAGAS (Es et al., 2024), a keyword-matching approach through LLM-based fuzzy matching.

**Results.** As shown in Table 5, TS-REASONER outperforms all baselines on inductive reasoning, demonstrating strong generalization to open-ended reasoning tasks.

## B Implementation Details.

TS-REASONER uses the Qwen-2.5-7B-Instruct as the LLM backbone across all the experiments with an embedding dimension of 5120, and uses the TimesFM-1.0-200M (Das et al., 2024) as our backbone TSFM with an embedding dimension of 1080. Parameters of the LLM are fine-tuned while those of the TSFM remain frozen during training. The detailed derivation of these time series embeddings from TimesFM can be found in the Appendix C. All training and inference procedures for TS-REASONER were conducted locally on 8 × L40s GPUs. In stage 1, training data is composed of 100K template-based pairs and 20K LLM-generated caption pairs. In stage 2, we employ an instruction tuning data from (Xie et al., 2024). Comprehensive training parameters are further detailed in Table 6.

## C TimesFM for Time Series Embedding

Given a time series  $\mathcal{T} \in \mathbb{R}^L$ , where L is the length of the time series. We first normalize it to have a mean of zero and a variance of one. We then segment  $\mathcal{T}$  into consecutive, non-overlapping

Table 6: Training details of TS-REASONER.

	Stage-1	Stage-2
Patch Size	32	32
Dataset	Captions	Instructions
#Samples	120K	30K
TFSM	TimesFM-1.0-200M	
LLM Backbone	Qwen2.5-7B-Instruct	
Trainable Params.	7.3B	7.3B
Batch Size	64	32
Learning Rate:	$1 \times 10^{-5}$	$2 \times 10^{-5}$
Epoch	1	2

Table 7: Comparison of lexical diversity between template-based pairs and LLM-generated pairs.

Metrics	MTLD $\uparrow$	Self-BLEU-4 $\downarrow$
Template-based pairs	42.95	0.82
LLM-generated pairs	<b>133.30</b>	<b>0.45</b>

patches of fixed length  $P$ , resulting in a total of  $N = \lfloor L/P \rfloor$  patches. This yields a patched time series  $\mathcal{T}_p \in \mathbb{R}^{N \times P}$ .

Following the approach of (Das et al., 2024),  $j$ -th patch  $\mathcal{T}_p^j$  is passed through a residual block to project it into the model dimension. This block is implemented as a two-layer MLP with a skip connection, processing each patch independently. The input token for the  $j$ -th patch is computed as:

$$\mathcal{E}_p^j = \text{InputResidualBlock}(\mathcal{T}_p^j) + \text{PE}_j, \quad (6)$$

where  $\text{PE}_j$  is the position encoding for the  $j$ -th patch, as defined in the original transformer (Vaswani et al., 2017). These encoded patch representations are then fed into an  $M$ -layer stacked Transformer to produce the final sequence of time series features:

$$\mathcal{Z}_T = \text{StackedTransformer}([\mathcal{E}_p^{(0)}, \mathcal{E}_p^{(1)}, \dots, \mathcal{E}_p^{(N)}]), \quad (7)$$

where  $\mathcal{Z}_T \in \mathbb{R}^{N \times d_{ts}}$  and  $d_{ts}$  denotes the embedding dimension for each time series patch. Refer to more details of TimesFM in (Das et al., 2024).

## D Caption Analysis

A critical limitation of synthetic datasets is the risk of models learning spurious correlations from similar templates. To mitigate this, our attribute-aware generation process is designed to produce captions that are both lexically diverse. To quantitatively

validate the richness of our approach, we compare it against the template-based method. We evaluate both lexical diversity using the Measure of Textual Lexical Diversity (MTLD) (Bestgen, 2024) and Self-BLEU-4 (Zhu et al., 2018) on a random sample of 1K captions from each dataset. The results presented in Table 7 show that our attribute-aware captions achieve an MTLD score of 133.30, a nearly 3 times increase over the template-based score of 42.95. Furthermore, the Self-BLEU-4 score is almost halved from 0.82 to 0.45. This substantial improvement in lexical diversity confirms that our method generates a significantly more expressive and diverse set of captions, crucial for training robust and generalizable models.

To ensure comprehensive data coverage, we curated time series with context from a wide range of domains. The distribution of these domains is visualized in Figure 7.

## E Analysis of Captioning

To qualitatively evaluate the distinct advantages of our approach, we conduct a case study comparing three methods: (1) our proposed attribute-aware captioning, which leverages visual time series plots and explicit attribute guidance; (2) a basic captioning baseline that operates on visual plots but lacks attribute guidance; and (3) LLM prompted with the raw textual (numerical) time series data. Our analysis, illustrated in Figure 9, yields two key insights.

(i) **Attribute-Aware Captions Provide Semantically Richer Descriptions.** A primary limitation of basic captioning is its tendency to produce superficial, chronological narrations of the data. As shown in Figure 9, the captioner describes the series' movements (e.g., "the value increases, then decreases sharply") but fails to extract deeper, underlying characteristics. While factually correct, this description omits properties crucial for a comprehensive understanding. In contrast, our attribute-aware captioning enriches this chronological account with critical semantic attributes. It not only captures the temporal dynamics but also identifies and articulates the series' overall trend, periodicity, and noise level. This multifaceted analysis provides a more holistic understanding of the time series, which is essential for TS-REASONER to conduct reasoning on downstream tasks.

(ii) **Visual Representation is Crucial for Capturing Global Temporal Patterns.** When comparing our visually-grounded method to an LLM

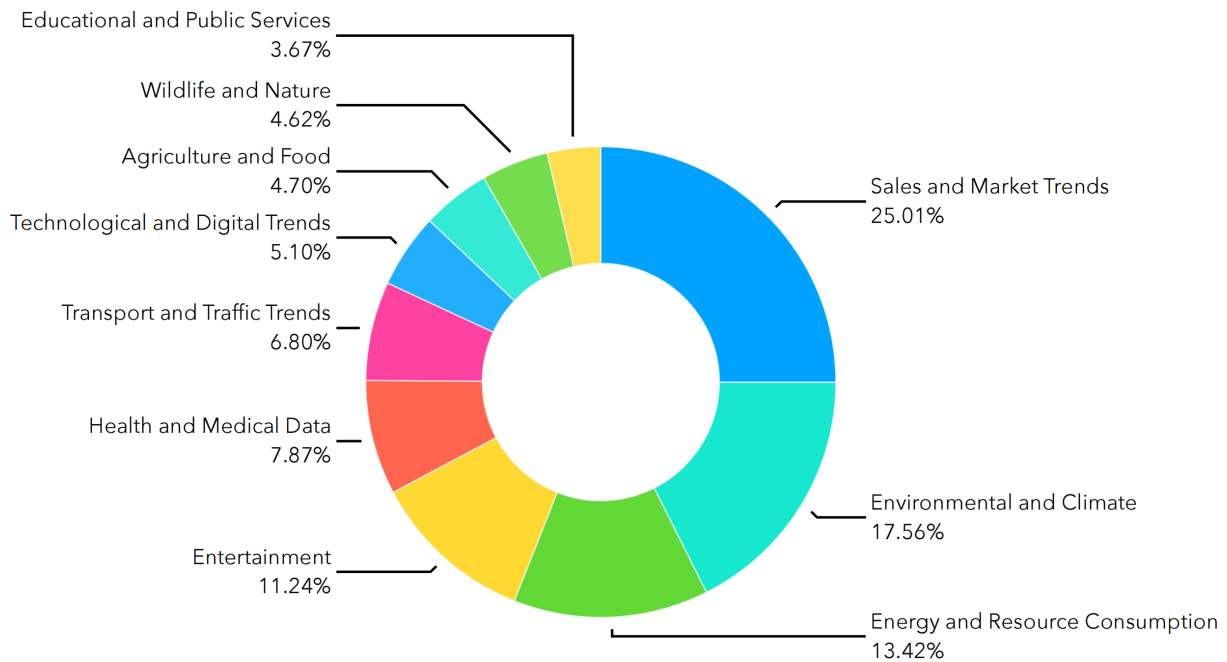


Figure 7: Domain distribution of LLM-generated time series with context.

processing raw numerical data, a significant gap emerges in the ability to identify global patterns. The text-based LLM, while capable of discerning local features like high-frequency oscillations or noise within a limited window, consistently fails to recognize the overarching periodicity of the entire series. We hypothesize that this failure stems from the inherent inefficiency of representing long numerical sequences as text. The exceeding length of input may distract the LLM, preventing it from observing the complete pattern. Conversely, a time series plot serves as a highly compressed, holistic representation. It enables the model to perceive the entire sequence as a single input, making global structures like periodicity visually salient and readily identifiable.

(iii) **Visual representations yield superior alignment captions compared to textual representations.** We employed GPT-4.1 to synthesize training pairs using raw numerical sequences (textual) versus visual plots (visual). As illustrated in Figure 8, TS-REASONER trained on visually-derived captions consistently outperforms its text-derived counterpart across most tasks. This indicates that visual representations enable LLMs to produce higher-fidelity captions for model alignment.

## F Prompts

Below, we detail all the prompts we used. Figure 10 shows our captioning prompts. Figure 11 shows the prompt for the TimeSeriesExam benchmark. Figure 12 and Figure 13 show the prompt for MTBench.

## G Data Leakage Examination

To guarantee evaluation integrity and prevent data leakage, particularly given LLM-synthesized data, we strictly prevent data leakage at both the source and semantic levels. We enforced source isolation by deriving training data exclusively from synthetic sources (Ansari et al., 2024) and public archives (Merrill et al., 2024) for alignment, and the instruction tuning dataset in (Xie et al., 2024), explicitly verifying no numerical overlap with the MTBench and TimeSeriesExam benchmarks. Furthermore, we ensured narrative and template isolation; our synthesized training captions utilize attribute-aware prompts focusing on generic properties like trends and periodicity, which are structurally and semantically distinct from the domain-specific financial analyses and weather reports in MTBench and the systematic property evaluation questions in TimeSeriesExam.

## Limitation

Our current experimental setup utilizes LLMs with up to 7B parameters, primarily due to existing re-

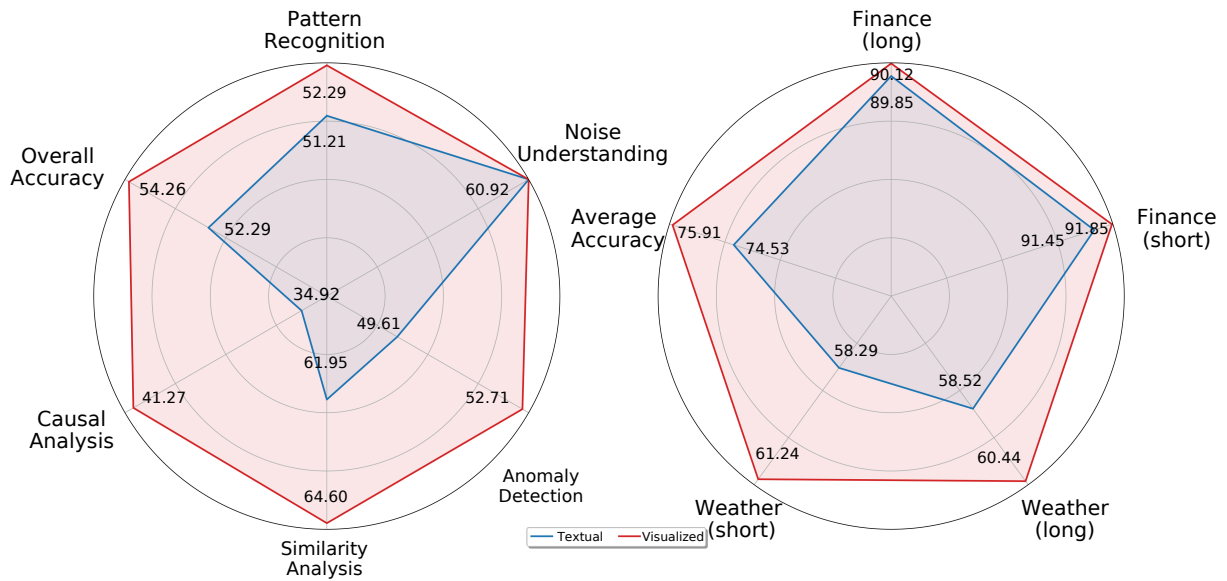
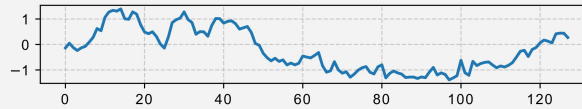


Figure 8: Results of TS-REASONER on TimeSeriesExam (left) and MTBench (right) using textual and visualized time series as training data.

1072 source constraints. While we anticipate that our  
 1073 findings will extend to larger models, future work  
 1074 should aim to validate these results with more sub-  
 1075 stantial LLMs. Additionally, exploring alternative  
 1076 architectural designs, such as different encoder ar-  
 1077 chitectures, and investigating more effective train-  
 1078 ing strategies for time series text alignment would  
 1079 be valuable avenues for future research. Future  
 1080 research could also explore incorporating RL ap-  
 1081 proaches to elicit reflection behaviors.

### Time Series



**Input:** Create a detailed description of the time series in one paragraph.

**Output:** The time series plot shows a fluctuating trend with distinct phases of increase and decrease over time. Initially, the series begins near zero and exhibits a sharp upward movement, peaking around the 10-20 time mark, where it reaches values greater than 1. This is followed by a volatile but generally declining trend that persists through to approximately the 60th time unit, dipping below -1. From this low point, the series remains subdued and oscillates around a low mean until about time step 100. After that, a gradual upward recovery begins, with the series rising back toward positive values and ending slightly above zero near time step 130. The fluctuations suggest a non-stationary process with alternating periods of high volatility and relative stability.

**Input with visualized time series:** Create a detailed description of the time series in one paragraph, including its trend, frequency properties, periodicity, noise, local variations, and other relevant characteristics.

**Output:** This time series displays a **prominent, low-frequency pattern without a clear long-term linear trend**. The series appears **non-stationary**, as its mean value shifts significantly over time, completing roughly one full oscillation over the observed interval of approximately 125 time units. The primary cycle involves a rise to a peak around time unit 20, followed by a decline to a trough near time unit 85, and a subsequent recovery. Superimposed on this dominant wave are higher-frequency, smaller-amplitude fluctuations, which introduce **considerable noise** and give the line a jagged, irregular texture. The volatility seems relatively constant, although there are **local variations in the magnitude and sharpness of these smaller oscillations**. Strong positive autocorrelation is evident, as consecutive data points are closely related, resulting in a continuous, flowing path rather than random jumps.

**Input with textual time series:** Here is a time series: [0.13, 0.05, -0.12, -0.24, ...]. Create a detailed description of the time series in one paragraph, including its trend, frequency properties, periodicity, noise, local variations, and other relevant characteristics.

**Output:** The time series exhibits a strong **quasi-periodic behavior** without a discernible long-term trend, characterized by a distinct cycle of a sharp rise to a peak, a steep fall into a trough, and a subsequent gradual recovery. This dominant low-frequency pattern is superimposed with considerable high-frequency noise and local variations, resulting in fluctuations that **make each cycle similar** but not identical. Overall, the data represents a noisy cyclical signal, where **a clear underlying periodicity** is modulated by random, short-term movements.



Figure 9: Case study comparing time series captioning methods. Basic visual captioning (row 1) gives a chronological description. Our attribute-aware visual captioning (row 2) provides a richer description including key attributes. Textual time series captioning (row 3) identifies some attributes but fails to capture the periodicity pattern.

Table 8: Example template questions for different reasoning tasks. Each subcategory covers a specific aspect of time series understanding, guiding the model to reason about comparative, anomalies, and causal relationships.

<b>Category</b>	<b>Subcategory</b>	<b>Example question</b>
Pattern Recognition	Trend	What is the most likely linear trend coefficient of the given time series?
	Cyclic	The given time series has sine wave pattern. How does its amplitude change from the beginning to the end?
	Stationarity	Is the given time series likely to be stationary after removing the cycle component?
	Regime Switching	Based on the given time series, how many different regimes are there?
	Statistical properties	Is the mean stable over time in the given time series?
	Random processes	Does the following time series exhibit a mean reversion property?
Noise Understanding	White Noise	Is the given time series a white noise process?
	Random Walk	Is the given time series likely to be a random walk process?
	Signal / Noise Ratio	You are given two time series with the same underlying pattern but different noise level. Which time series has higher magnitude of noise?
Anomaly Detection		The following time series has two types of anomalies appearing at different time points. What are the likely types of these anomalies?
Similarity Analysis	Shape	Despite the noise, do the given two time series have similar patterns?
	Distributional	You are given two time series which are generated using a random walk. Are they likely to have the same variance?
Causality Analysis	Granger Causality	Is there Granger causality between the two time series?

Instructions for prompting LLMs to generate time series captions.

1. Write a paragraph that analyzes the time series, covering its local behaviors, noise levels, periodic structures, overall trend, frequency content, and any other characteristics you consider important.
2. Create a detailed description of the time series in one paragraph, including its trend, frequency properties, periodicity, noise, local variations, and other relevant characteristics.
3. Provide a paragraph summarizing the time series characteristics such as noise, periodic patterns, long-term trends, frequency behavior, local anomalies, and any other significant features.
4. Compose a detailed caption describing the frequency characteristics, noise, trends, local variations, periodic structures, and any other meaningful patterns you observe in the time series.
5. Craft a one-paragraph summary of the time series, noting local fluctuations, periodic behavior, frequency features, trend, noise content, and any other insights you find important.
6. Generate a descriptive paragraph detailing the time series' key attributes, including frequency structure, noise patterns, trend direction, local features, periodic elements, and other notable aspects.
7. Give a thorough one-paragraph explanation of the time series, addressing periodicity, noise, frequency components, trend, local variations, and other relevant characteristics.
8. Write a narrative paragraph explaining the time series, focusing on noise, frequency characteristics, periodicity, localized structures, the overall trend, and other important features you identify.
9. Summarize the time series in a paragraph, describing its fluctuations, recurring patterns, noise levels, frequency-domain features, trend direction, and any additional traits you find significant.
10. Develop a paragraph that captures the key features of the time series, such as frequency traits, trend, noise, periodic components, local behaviors, and other characteristics worth noting.
11. Provide a one-paragraph caption analyzing the time series data in terms of noise, trend, periodicity, local features, frequency-related behavior, and any additional characteristics of interest.
12. Create a rich paragraph description of the time series, including its trend, local anomalies, periodic activity, noise artifacts, spectral content, and other important descriptive elements.
13. Write a descriptive paragraph for the time series, highlighting frequency properties, trend behavior, periodic patterns, local structures, noise, and other characteristics you consider relevant.
14. Generate a compact yet thorough paragraph explaining the time series in terms of periodicity, trend movement, noise level, frequency details, local dynamics, and any other key aspects.
15. Construct a one-paragraph analysis of the time series by examining its local variations, noise, trend, periodic elements, frequency spectrum, and other notable features you deem important.
16. Write a summary paragraph that discusses the time series' periodic features, trend behavior, local patterns, noise levels, frequency domain signals, and other characteristics worth mentioning.
17. Create a detailed one-paragraph commentary on the time series that outlines its noise characteristics, periodicity, frequency content, trends, localized behaviors, and other useful insights.
18. Prepare a paragraph-long description of the time series covering its trend, noise, frequency-related traits, local fluctuations, periodic structures, and any additional attributes of note.
19. Offer a one-paragraph interpretation of the time series, highlighting its frequency features, periodic nature, local patterns, noise, trend line, and any other important characteristics you observe.
20. Compose a detailed summary in one paragraph focusing on the time series' periodic behavior, frequency spectrum, localized fluctuations, overall trend, noise, and other relevant descriptive elements.

Figure 10: The list of instructions for attributes-aware time series captioning.

Instructions for prompting LLMs for time series understanding and reasoning tasks in Timeseriesexam.

You are a time series analysis expert. These are the time series data:  $\langle ts_1 \rangle \langle ts_1 \rangle \dots \langle ts_k \rangle \langle ts_k \rangle$ . Please answer the question and provide the correct option letter.  
Question:  $\langle \text{Question} \rangle$   
Choices:  $\langle \text{choices} \rangle$

Figure 11: The list of instructions for attributes-aware time series captioning.

Instructions for prompting TS-REASONER for financial reasoning tasks in MTBench.

You are an expert in finance and stock market analysis. Your task is to answer the question based on the given n-day historical stock price time series and a financial analysis published at the last timestamp of the time series. The time series is:  $\langle ts_1 \rangle \langle ts_1 \rangle$ .  
 $\langle \text{Context} \rangle$   
Question:  $\langle \text{Question} \rangle$   
Choices:  $\langle \text{Choices} \rangle$

Figure 12: Prompt example for time series reasoning in Finance.

Instructions for prompting TS-REASONER for weather reasoning tasks in MTBench.

You have an n-day temperature time series, and a weather event report published on the last day of the time series. The time series below is the 14-day temperature between <start time> to <end time>, and the time interval is 1 hour:  $\langle ts_1 \rangle \langle ts_1 \rangle$ .  
The following events are reported:  
<Context>  
Question: <Question>  
Choices: <Choices>

Figure 13: The prompt for time series reasoning in Weather.