

ChatTS: Aligning Time Series with LLMs via Synthetic Data for Enhanced Understanding and Reasoning

Zhe Xie

Tsinghua University
China, Beijing

xiez22@mails.tsinghua.edu.cn

Zeyan Li

Xiao He
ByteDance
China, Beijing

Longlong Xu

Tsinghua University
China, Beijing

Xiaodao Wen

BizSeer
China, Beijing

Tieying Zhang

Jianjun Chen
ByteDance
USA, San Jose

Rui Shi

ByteDance
China, Beijing

Dan Pei

Tsinghua University
China, Beijing

ABSTRACT

Understanding time series is crucial for its application in real-world scenarios. Recently, large language models (LLMs) have been increasingly applied to time series tasks, leveraging their strong language capabilities to enhance various applications. However, research on multimodal LLMs (MLLMs) for time series understanding and reasoning remains limited, primarily due to the scarcity of high-quality datasets that align time series with textual information. This paper introduces ChatTS, a novel MLLM designed for time series analysis. ChatTS treats time series as a modality, similar to how vision MLLMs process images, enabling it to perform both understanding and reasoning with time series. To address the scarcity of training data, we propose an attribute-based method for generating synthetic time series with detailed attribute descriptions. We further introduce Time Series Evol-Instruct, a novel approach that generates diverse time series Q&As, enhancing the model's reasoning capabilities. To the best of our knowledge, ChatTS is the first MLLM that takes multivariate time series as input for understanding and reasoning, which is fine-tuned exclusively on synthetic datasets. We evaluate its performance using benchmark datasets with real-world data, including six alignment tasks and four reasoning tasks. Our results show that ChatTS significantly outperforms existing vision-based MLLMs (e.g., GPT-4o) and text/agent-based LLMs, achieving a 46.0% improvement in alignment tasks and a 25.8% improvement in reasoning tasks. We have open-sourced the source code, model checkpoint and datasets at <https://github.com/NetManAIOps/ChatTS>.

1 INTRODUCTION

Multimodal large language models (MLLMs) have recently achieved significant progress in vision-language tasks, showing exceptional performance even in scenarios requiring complex understanding and reasoning [9, 31, 37, 58]. However, this success has not been replicated in the time series domain. Even though some studies have attempted to integrate LLMs with time series, such as TimeLLM [26], they usually only focus on specific classical time series tasks (e.g., forecasting) rather than understanding, reasoning, and dialogue based on time series attributes, as well as integrating into existing LLM workflows. Moreover, recent studies indicate that LLMs still struggle with zero-shot reasoning about time series [43]. This is particularly significant because time series analysis, widely applied

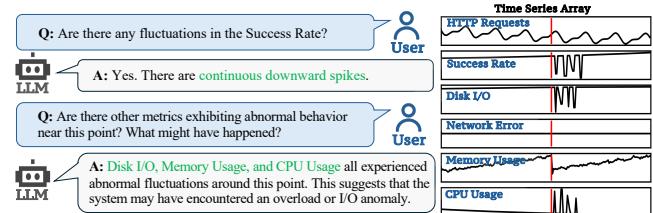


Figure 1: Example of an AIOps application of time series-related dialogue.

in domains such as electricity [53], healthcare [46], traffic [33], weather [36], and finance [50], frequently requires understanding and reasoning about time series patterns. Therefore, the ability to reason using both text and time series data is a critical capability for MLLMs, enabling them to support human decision-making by providing natural language explanations that align with human logic. Figure 1 illustrates such an example in an AIOps [66] scenario where understanding and reasoning about multivariate system monitoring time series are achieved through natural language dialogue, thereby improving the diagnostic and troubleshooting process.

Existing LLM-based methods for understanding and reasoning about time series attributes can be broadly categorized into text-based, vision-based, and agent-based approaches. Text-based methods directly use LLMs by structuring historical observations as raw text [8]. However, these methods are often constrained by the limitation of prompt length and generally perform poorly in understanding the global features of time series compared to vision-based methods. Vision-based methods utilize vision MLLMs, which accept plot figures of time series data [43], such as GPT-4o [5] or Qwen-VL [9]. While these methods can better capture global features, they are limited by the resolution of the plotted figures and face challenges in accurately interpreting the details. Recent works also show how agents can leverage time series analysis tools to interact with LLMs [52, 68]. However, the ability of agents to understand time series is restricted by the functionality of the tools.

Therefore, there is a strong need for TS-MLLM, a MLLM that can naturally handle time series modality, akin to how vision MLLMs process images. Such models have the potential to unlock valuable insights from time series by providing intuitive, question-driven analysis capabilities. Specifically, TS-MLLMs can capture global and

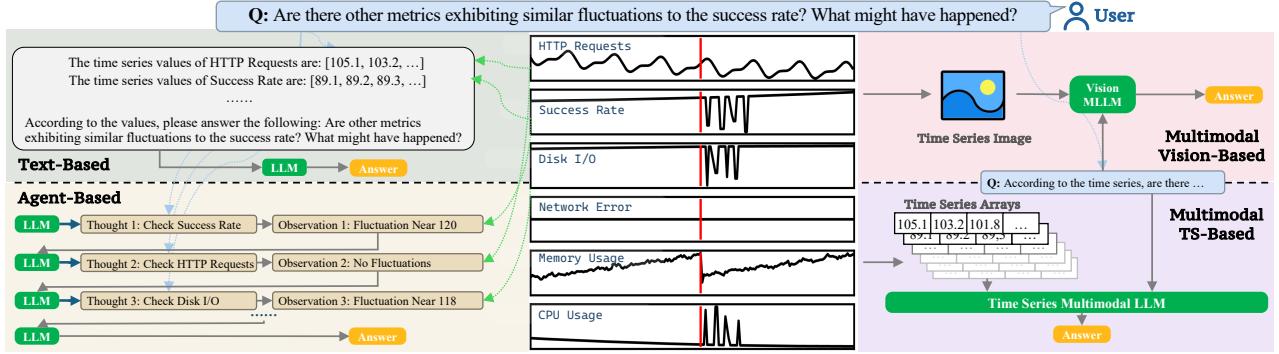


Figure 2: Comparison of four kinds of LLM-based methods for time series understanding and reasoning.

local features and relationships between multivariate time series (MTS), areas where existing LLMs and MLLMs have struggled. By incorporating textual modalities as input, these models can broaden their applicability and better contextualize time series data, aligning the analysis with user queries. If successful, TS-MLLMs could perform novel tasks such as citing patterns and events in time series as evidence for observations and inferences, drawing interpretable conclusions from complex dynamical systems, and recognizing and responding to temporal patterns [43].

However, developing TS-MLLMs with effective understanding and reasoning ability for time series attributes faces several core challenges. First, multimodal time series data, especially language-time series pair data, is extremely scarce [15, 27, 43]. Unlike modalities such as images and audio, almost no research focuses on the language alignment of time series. As a result, there is a significant lack of time-series + text data, which makes the construction of time-series dialogue and reasoning datasets challenging. This is fundamental for TS-MLLMs to develop temporal understanding and reasoning capabilities. Second, time-series data contains abundant shape and numerical attributes (*i.e.*, the types of local fluctuations and their amplitudes). Therefore, a diverse range of text is needed to comprehensively describe these attributes while ensuring accuracy to achieve effective alignment. Third, real-world time-series data are usually variable in length, multivariate, and of uncertain quantity. The correlations among MTS are often a focus of attention (as illustrated in Figure 1). In MLLMs for other modalities, such as images, few methods emphasize the relationships between multiple samples. However, such relationships are indispensable for understanding and reasoning about time series. Finally, there is a lack of evaluation data and methods for TS-MLLMs. Developing comprehensive and reasonable datasets and methodologies to evaluate their performance is necessary.

To address the challenges above, we innovatively propose a method to fine-tune a pre-trained LLM for TS-MLLMs solely using synthetic time series and text data. An important reason is that synthetic time series data for time series model training has shown good results [21]. However, current methods are difficult to apply directly because time series-text alignment tasks require both *precise* and *diverse* time series attribute descriptions. Therefore, we propose an attribute-based method for generating synthetic time series and precise text attributes to facilitate the modal alignment

of time series with LLMs. Compared with existing studies on synthetic time-series generation [21, 61], the proposed attribute-based time-series generation method provides precise textual attributes for each detailed pattern of the time series, laying a foundation for generating diverse text data. Furthermore, to equip MLLM with enhanced time series understanding and reasoning capabilities, we propose the Time Series Evol-Instruct (TSEvol) algorithm. Through the diverse combinations of attributes and tasks, TSEvol can generate diverse time series Q&A datasets through evolutions, thereby enhancing the model’s overall performance. To handle multivariate time-series inputs and fully preserve semantic information, we propose ChatTS, trained using the generated synthetic datasets. ChatTS employs a context-aware time-series encoder capable of encoding time series of (theoretically) arbitrary length and quantity while retaining their original numerical information. Finally, to support comprehensive evaluation regarding both language alignment and time series reasoning, we have collected evaluation datasets comprising both real and synthetic time series. These datasets include both alignment and reasoning tasks with uni/multivariate time series, ensuring a thorough assessment of the model’s performance. **Our contributions.** This paper makes the following contributions.

- We propose to align LLMs with time series using attribute-based synthetic time series and text data. Building on this, we further introduce Time Series Evol-Instruct (TSEvol), an algorithm that generates diverse, accurate, and multimodal training datasets of time series and text entirely through synthetic data generation.
- We propose a context-aware TS-MLLM, ChatTS, designed for variable-length, multivariate time series input and trained using the generated synthetic data. To the best of our knowledge, ChatTS is the *first TS-MLLM with multivariate time series as input for understanding and reasoning about time series attributes*.
- We have collected evaluation datasets containing real-world time series data, including six alignment tasks and four reasoning tasks. Evaluation results across multiple datasets demonstrate that ChatTS significantly outperforms baseline models, including GPT-4o, in both time series alignment and reasoning tasks.
- We have open-sourced the model, source code, and evaluation datasets to support future research: <https://github.com/NetManAIOps/ChatTS>.

2 PRELIMINARY AND MOTIVATION

2.1 Problem Definition

The task of a TS-MLLM is to generate text-based responses based on the input textual query and MTS array. Given a set of time series $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$, where each $T_i = \{t_{i,1}, t_{i,2}, \dots, t_{i,m_i}\}$ represents a sequence of m_i observed values over time for the i -th metric, and a natural language question Q , the goal is to generate an answer A that captures relevant patterns or relationships across \mathcal{T} based on the context of Q . Formally, it can be defined as follows:

- **Input:**
 - A set of time series $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$, where $T_i \in \mathbb{R}^{m_i}$ represents the values of the i -th metric over m_i time points.
 - A natural language query Q specifies the information of interest within the time series data.
- **Output:** A text answer A derived from the \mathcal{T} analysis, providing insights based on Q .

The task of TS-MLLM can be expressed as a function:

$$f(Q, \mathcal{T}) \rightarrow A,$$

where f denotes the model or algorithm responsible for interpreting the text query Q and generating the text answer A by analyzing relevant patterns and relationships across the time series in \mathcal{T} .

2.2 Existing Methods

Although mainstream LLMs currently do not support the direct input of time series modality data, time series information can be provided to LLMs through alternative methods to do simple understanding and reasoning about time series attributes, as shown in Figure 2. Existing approaches can be broadly categorized into text-based, vision-based, and agent-based, each with distinct limitations.

Text-based methods encode time series values as raw text [8]. However, these methods are constrained by the length of prompts, limiting their global analysis capabilities and often resulting in an incomplete understanding of the data context (refer to Section 4).

Vision-based approaches, which use visual representations of time series data (e.g., time series plots) processed by vision MLLMs [5, 9], may face challenges in accurately capturing detailed information in time series, resulting in lower accuracy for data-intensive tasks and high computational overhead (refer to Section 4).

Agent-based methods employ a reasoning and action strategy, breaking down complex tasks into a sequence of thoughts, observations, and actions conducted by external tools to analyze time series. While potentially more flexible, this approach is heavily dependent on expert knowledge and effectiveness of tools, token-intensive, and time-consuming, often requiring extensive token chains to handle MTS data. Additionally, hallucination becomes a significant problem [59] as the chains grow longer, reducing reliability in complex analytical tasks.

2.3 Time Series Multimodal LLM

TS-MLLM is a new type of MLLM that aims at overcoming the limitations of existing methods by *natively* integrating both textual and time series inputs (see Figure 2). It can process multiple time series data and textual descriptions, enabling a unified analysis that captures complex, multivariate relationships. Unlike previous methods,

it does not rely on lengthy token chains or visual representations, thereby reducing computational overhead and mitigating issues with hallucination. Through the alignment of time series and text, TS-MLLM can perform both global and local analysis of the shape and numerical information of time series. This capability allows it to achieve higher accuracy and greater potential than existing methods.

3 METHODOLOGY

3.1 Overview

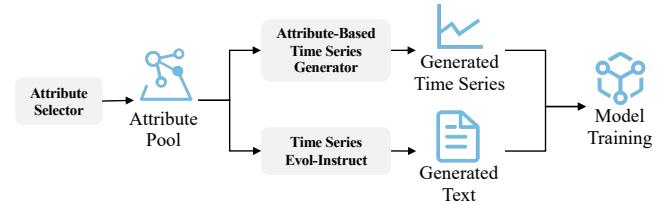


Figure 3: Overview of ChatTS.

Due to the scarcity of high-quality datasets that align time series with textual information, we propose to generate synthetic text-time series pairs for model training. Synthetic data is a common approach when there is a lack of sufficient real training data, and its effectiveness has been well validated in various fields [21, 39, 49]. However, as discussed earlier, "time series + text" data for TS-MLLM requires sufficient accuracy to ensure alignment precision, comprehensive coverage of time series attributes to guarantee effective multimodal alignment, and task diversity in the text to enhance QA and reasoning abilities. Unfortunately, existing time series generation methods [21, 61] fail to achieve these goals. A key reason is that we need a *diverse* set of time series and *precise, detailed* descriptions of time series patterns. Therefore, in this paper, we propose an attribute-based method to generate time series + text data, as illustrated in Figure 3:

- **Attribute Selector** (Section 3.2): To produce highly controllable time-series data with precise attributes, we use a detailed feature set to describe time series. These attributes are aligned with real-world settings through an LLM selection.
- **Attribute-Based Time Series Generator** (Section 3.2): Construct time series that correspond exactly to the attribute pool using a rule-based approach.
- **Time Series Evol-Instruct** (Section 3.3): A novel Time Series Evol-Instruct module for creating large, diverse, and accurate datasets of time-series and text question-answering pairs for complex reasoning.
- **Model Design** (Section 3.4): To handle MTS, we design a context-aware MLLM encoding for multiple time series input, along with a value-preserved time series encoding method.
- **Model Training** (Section 3.5): A large-scale training and a SFT are conducted to perform language alignment and improve time series-related reasoning ability.

As shown in Figure 3, the framework in ChatTS integrates synthetic data generation and model training into a pipeline that ensures effective time series attributes understanding and reasoning with

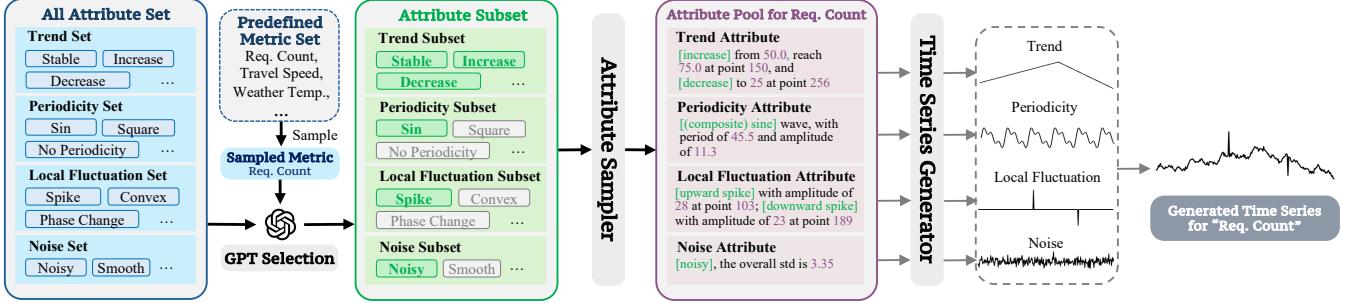


Figure 4: Attribute selector and attribute-based time series generator in ChatTS.

only synthetic data. First, building on the attribute-based time series generation and TSEvol described in Sections 3.2 and 3.3, the pipeline generates synthetic data that captures intricate numerical and textual information for effective multimodal alignment. This data is then used to train the model (Section 3.4), where the context-aware time series encoder preserves the time series values while aligning attributes with textual semantics accurately. Finally, by alignment training and SFT (Section 3.5), ChatTS achieves precise alignment between time series encoding and text embeddings, along with enhanced reasoning capabilities.

3.2 Attribute-Based Time Series Generator

Diverse time series and precise, detailed textual attribute descriptions are essential to achieve accurate time series language alignment. Time series have rich pattern attributes, which can be roughly categorized into trend, periodicity, and remainder [24, 47]. Much existing research on the generation of time series [20, 21] also adopts similar approaches to classify these attributes. Therefore, following existing studies, we classify time series attributes into four major categories, Trend, Periodicity, Noise, and Local Fluctuation, to construct the corresponding attribute set for time series.

Based on this, we propose an attribute selector and an attribute-based time series generator that produces synthetic time series data (see Figure 4). First, we define an “All Attribute Set”, which includes many specific attributes under different attribute categories. The All Attribute Set includes 4 types of Trend, 7 types of Seasonality, 3 types of Noise, and 19 types of local fluctuations. The complete list can be found in the source code. Different attributes within the same category can be combined. A time series can include multiple segments of trends and several local fluctuations by combining the same type of attributes (see Figure 4). Additionally, by combining sine waves, we can generate a diverse range of periodic fluctuation patterns. Therefore, the proposed time series generator can theoretically generate an infinite number of different time series, ensuring the richness of attributes. We also introduced a GPT Selector. Specifically, when generating an attribute set for time series, we randomly sample a metric from a large “Metric Set” that contains 567 predefined metric names from real-world scenarios and use GPT to choose a *attribute subset* from the all attribute set, based on the actual physical meaning of the metric and the predefined scenario. This helps align time series with real-world physical meanings.

Then, the *Attribute Sampler* randomly samples a combination of attributes from the Attribute Subset. It also assigns specific numerical values, like position and amplitude, based on rules and constraints from the GPT Selector. These details are stored in the “Attribute Pool”, which records all the detailed information about a time series. The *Time Series Generator* finally creates time series arrays that *exactly* match the attributes from the pool in a rule-based manner (more details can be found in the source code). This process allows us to generate diverse synthetic time series with precise attribute descriptions.

3.3 Time Series Evol-Instruct

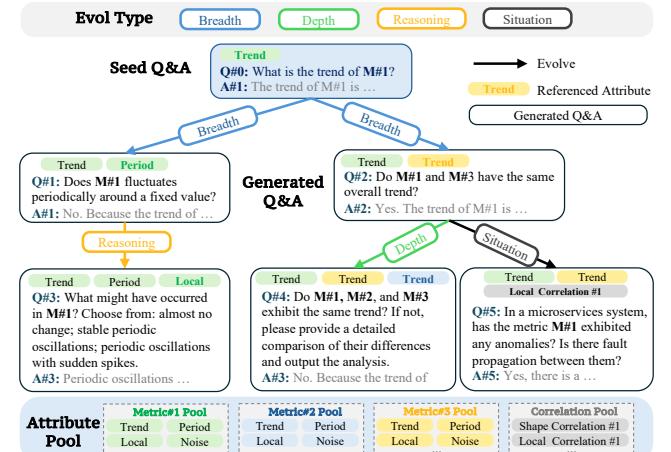


Figure 5: Time Series Evol-Instruct

To improve the model’s question-answering and reasoning abilities, it is essential to have high-quality SFT training data that is diverse in format and tasks. However, due to the lack of time-series + text data, it is challenging to obtain sufficiently diverse time-series-related training data directly. To generate accurate time-series + text SFT data with rich question-answering formats, inspired by Evol-Instruct [55] and its multimodal version MM-Evol [40], we innovatively propose Time Series Evol-Instruct (TSEvol).

Evol-Instruct [55] is a data generation approach that incrementally evolves instructional prompts and their outputs to enhance the diversity and complexity of training datasets for LLMs. TSEvol

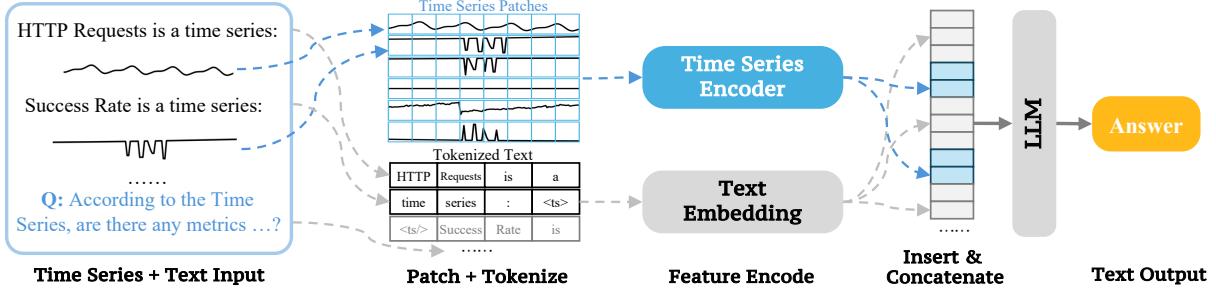


Figure 6: Model Structure of the Multimodal LLM in ChatTS

builds upon Evol-Instruct by introducing a mechanism to incorporate time series attributes dynamically into each evolutionary step (see Figure 5). TSEvol relies on *attribute pools* of multivariate time series (see Section 3.2). Additionally, to enhance the model’s ability to analyze correlations, we introduce a correlation pool, which records time series with related attributes (refer to the source code for details). During each step of the evolution process, a subset of attributes is randomly selected from the *attribute pool* and added as *additional context*, guiding the LLMs to generate Q&As about a broader set of time series attributes according to the *evolution type*. With TSEvol, generated Q&As can cover more attributes in the time series and avoid repetitive questions. We also added an attribute-based eliminator to ensure the Q&As match the time series attributes. In addition to the commonly used evolution types, we also add two more types, reasoning (reasoning-based questions) and situation (situation-based questions), to enhance the model’s ability to handle complex questions.

3.4 Time Series Multimodal LLM

In this subsection, we introduce the model structure of the proposed ChatTS, as shown in Figure 6. ChatTS takes multivariate time series and text, along with their *contextual information* as the input.

3.4.1 Context-Aware Time-Series Multimodal LLM. To handle the multimodal inputs, ChatTS first separates the input time series arrays and the text. Following the established practice in encoding time series for LLMs [26], the input time series arrays are divided into fixed-size patches, which enables the model to handle and encode temporal patterns more effectively. We employ a simple 5-layer MLP to encode each patch of the time series, as time series inherently have sequential patterns. Therefore, a simple structure can map the patch features to a space aligned with the text embedding. For text input, they are tokenized and then encoded through a text embedding layer. In this way, each patch of the time series and each text token are mapped to the same space.

To fully retain the contextual information of multivariate time series, we performed token-level concatenation based on the position of the time series in the original input. Specifically, the encoded patches corresponding to each time series were inserted between the surrounding text tokens. Unlike the method used in TimeLLM [26], this approach ensures that the contextual information of the time series is fully preserved. This is especially important in multivariate scenarios, where referencing the corresponding time series in textual form is often necessary. This process results in a sequence that reflects the multivariate structure of the data, enabling

the LLM to capture both temporal and contextual dependencies across different metrics. This sequence is then fed into the LLM, which generates an answer that incorporates insights from both the time series data and the natural language query, achieving a multimodal understanding suited for complex question-answering tasks.

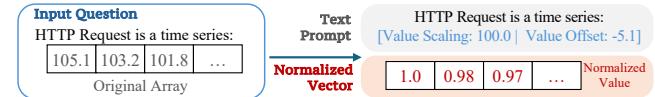


Figure 7: Value-Preserved Time Series Normalization

3.4.2 Value-Preserved Time Series Normalization. The numerical features of time series are essential, as real-world applications often involve specific numerical queries (e.g., asking for the maximum CPU utilization). However, normalization of time series data can lead to losing original numerical information. To address this, we introduce a value-preserved time series normalization scheme (as shown in Figure 7). First, we apply standard min-max normalization (0-1 scaling) to each time series array. Then, for each time series, we include the normalization parameters—“Value Scaling” (the scaling factor during normalization) and “Value Offset” (the offset applied during normalization)—in the text as part of a prompt. This approach leverages the numerical understanding capabilities of LLMs, enabling us to normalize time series features while preserving the original numerical information. To further enhance numerical understanding, numerical tasks are included in the training dataset (see Section 3.5).

3.5 Model Training

Table 1: Training Datasets

Stage	Alignment			SFT	
	UTS	MTS-Shape	MTS-Local	TSEvol	Instruct Follow
# Samples	35,000	35,000	35,000	24,270	5,050

ChatTS is trained based on QWen2.5-14B-Instruct [56]¹, with a two-stage fine-tuning process: large-scale alignment training and supervised fine-tuning (SFT). Table 1 shows the datasets we use during training.

¹<https://huggingface.co/Qwen/Qwen2.5-14B-Instruct>

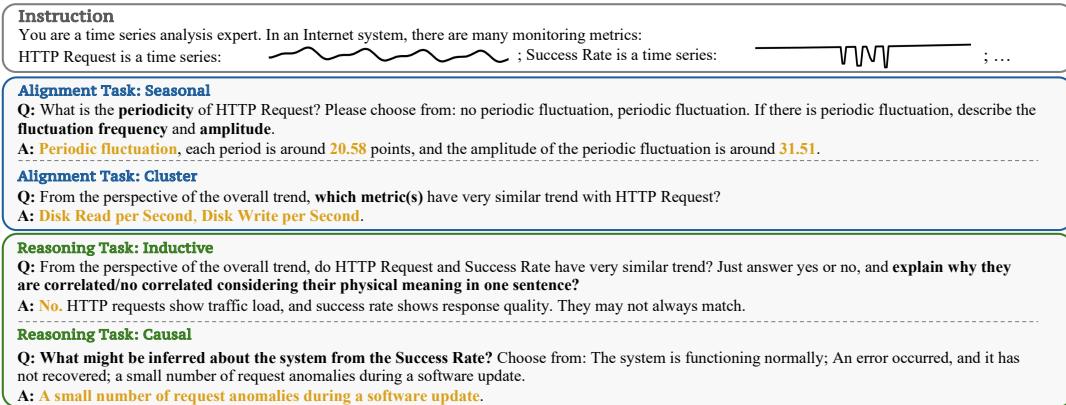


Figure 8: Example QAs in some evaluation tasks.

3.5.1 Large-Scale Alignment Training. In the first stage, we perform large-scale alignment training using the attribute-based synthetic time series data to establish an initial alignment between the text and time series modalities within the LLM. This stage enables ChatTS to align textual descriptions with time series attributes effectively. During the alignment stage, we created three datasets for large-scale training based on a series of manually designed templates and LLM refinement. The *UTS* dataset includes tasks for basic attribute descriptions of univariate time series (both global and local attribute tasks are included). The *MTS-Shape* dataset consists of multivariate data with *global* trend correlations designed to enhance the model’s ability to analyze multivariate correlations. The *MTS-Local* dataset contains multivariate data with correlated *local* fluctuations, aiming to improve the model’s capability in analyzing local features of multivariate data. Given MTS’s more complex feature combinations, we set the training data size for MTS and UTS at an approximately 2:1 ratio. We conduct a dataset scaling study in Section 4.5 to investigate the impact of training dataset size.

3.5.2 Supervised Fine-Tuning. In the second stage, we use SFT to develop the LLM’s ability to perform complex question-answering and reasoning tasks. This stage utilizes two main types of training data: the datasets generated with TSEvol, designed to enhance the model’s question answering and reasoning ability about time series, and an instruction-following (IF) dataset, constructed based on a series of predefined templates, designed to enhance the model’s ability to follow specific response formats. For TSEvol, we used the dataset from alignment training along with LLM-generated QAs as the seed data. Together, these datasets train the multimodal LLM to respond accurately to time series-specific queries and follow task instructions, strengthening its capacity for complex, context-driven question-answering and reasoning tasks. In both Alignment and SFT stages, we enhance ChatTS’s numerical capabilities through a series of numerical tasks. Specifically, we explicitly train the model to learn various aspects, such as maximum/minimum values, segmented averages, local features (e.g., spike positions and amplitudes), seasonality and trend amplitudes, and raw numerical values at individual time points. The numerical evaluation metrics in our experimental results further demonstrate ChatTS’s strong performance in time series numerical analysis.

3.5.3 Training Settings. We use QA pairs as the data format for both training stages. During alignment training, we mixed in a small amount of IF data and found that this mitigates the decline in the model’s IF ability. In the SFT stage, we mixed 30% of the alignment training dataset to reduce overfitting. The training dataset includes time series with lengths ranging from 64 to 1024 to ensure that ChatTS can handle varying time series lengths. Full-parameter SFT is used for ChatTS with DeepSpeed [2] and LLaMA-Factory [65], with Qwen2.5-14B-Instruct [6, 56] as the base model. Inference for both Qwen and ChatTS is also conducted with DeepSpeed.

4 EVALUATION

In this section, we will comprehensively evaluate the performance of ChatTS by answering the following research questions (RQs):

- **RQ1.** How well does ChatTS align with time series?
- **RQ2.** How does ChatTS perform in time series reasoning tasks?
- **RQ3.** Are attribute-based data and TSEvol effective?
- **RQ4.** How does the training set size affect model performance?
- **RQ5.** Is the time series modality in ChatTS truly useful?
- **RQ6.** Does ChatTS, with its native time-series multimodal capabilities, have advantages over agent-based methods?

4.1 Experimental Setup

4.1.1 Evaluation Tasks. To comprehensively evaluate the model’s performance, we set two categories of evaluation tasks: alignment tasks and reasoning tasks, following the general evaluation methods of multimodal LLMs [10, 37, 40]. For each type of evaluation task, we designed a series of subtasks based on existing work. Some example QAs are shown in Figure 8 (more details can be found in the source code). Specific tasks that rely heavily on domain-specific knowledge (e.g., classification and etiological reasoning) were excluded due to the lack of high-quality datasets that provide sufficient background information. Therefore, we primarily focused on the following tasks:

- Alignment tasks are divided into univariate and multivariate:
- **Univariate tasks.** Identify trends, seasonality, noise, and local fluctuations. These tasks include both *categorical* subtasks and *numerical* subtasks.
 - **Multivariate tasks.** Correlation and clustering. These tasks are all categorical.

The reasoning tasks include inductive reasoning, deductive reasoning, causal reasoning, and comparison reasoning (MCQ2):

- **Inductive reasoning.** Q&A task. Inductive summarization of the physical meaning reflected by a uni/multivariate time series.
- **Deductive reasoning.** True/False (T/F) task. Reasoning based on a predefined condition in conjunction with univariate time series.
- **Causal reasoning.** Multiple-choice task. Based on univariate time series, select the most likely cause.
- **Comparison reasoning (MCQ2).** Multiple-choice task. Compare two time series and select the correct answer.

More details about the evaluation tasks can be found in the source code and the evaluation dataset.

4.1.2 Evaluation Metrics. For categorical tasks in alignment evaluation, we match labels from the responses of LLMs using rule-based matching and use F1-Score as the metric. For numerical tasks in alignment evaluation, we extract numbers from the responses of LLMs and use *relative accuracy* (1.0 - relative error) as the metric:

$$\text{relative_accuracy} = \max \left(1.0 - \frac{|V_{\text{answer}} - V_{\text{label}}|}{|V_{\text{label}}|}, 0.0 \right)$$

We set a minimum value of 0.0 for relative accuracy to mitigate the impact of outlier results. For Q&A tasks in inductive reasoning, answers are evaluated using RAGAS [19], a keyword-matching approach through LLM-based fuzzy matching. T/F and MC tasks are directly evaluated through choice matching and the accuracy is calculated. All evaluation metrics are the higher, the better.

Table 2: Tasks in Evaluation Dataset

Dataset	Tasks	# Questions
A	Alignment (Trend, Season, Noise, Local, Correlation, Cluster), Reasoning (Inductive, Deductive, Causal)	525
B	Alignment (Trend, Season, Noise, Local, Correlation, Cluster), Reasoning (Inductive)	1,616
MCQ2	Reasoning (Comparison - MCQ2)	100

4.1.3 Evaluation Datasets. Our evaluation is conducted on three datasets (see Table 2) to test the model’s performance across both real-world and synthetic time series scenarios. Dataset A and B are collected by us, and Dataset MCQ2 is an open-source dataset [43].

Dataset A includes real-world time series data collected from multiple domains, including AIOps [35], weather [3], the NAB (Numenata Anomaly Benchmark) [7], and Oracle system metrics [34]. We manually label and collect a total of 525 questions, including both alignment tasks and reasoning tasks.

To expand the size of the evaluation set, we used the attribute-based time series generator introduced in ChatTS to generate a series of time series and created alignment Q&A by applying a set of templates. We also develop a set of reasoning questions with LLM, resulting in a larger-scale *Dataset B* containing 1,616 questions. Considering the complexity of reasoning tasks, we have included only inductive reasoning tasks in the reasoning tasks of this dataset to ensure the quality of the questions.

MCQ2 [43] is an open-source dataset [4] that includes comparison reasoning tasks. The questions, answers, and time series in this dataset are all generated by LLMs. We did not use the etiological reasoning and forecasting datasets as they are not aligned with our evaluation settings. Furthermore, [43] suggests that the settings of the MCQ1 dataset are unsuitable for evaluating the performance of time series reasoning, so we also did not adopt it. Considering the inference cost, we randomly sampled 100 questions.

4.1.4 Baselines. Based on different modalities, we categorized the baseline methods into the following types:

- **Text-Based:** These methods convert time series arrays into textual prompts as inputs for LLMs. We choose several mainstream LLMs as our base model (GPT-4o/GPT-4o-mini/GPT-4-Turbo/QWen2.5-14B-Instruct) for evaluation.
- **Vision-Based:** These methods plot time series and input them into visual MLLMs. We choose mainstream vision MLLMs (GPT-4o/GPT-4o-mini) for evaluation.
- **Agent-Based:** These methods employ the ReAct [57] framework to interact with multiple tools to analyze the time series. The tools used include single-point/range query, STL decomposition, anomaly detection (autoregression AD in adtk [1]), and classification (Rocket [18]) for UTS; trend/fluctuation correlation (based on Pearson correlation & rules), multivariate version of AD and classification for MTS. We choose GPT-4o/GPT-4o-mini for the agent. More details about the tools’ implementation can be found in the source code. *We also conducted additional experiments to explore further the capabilities of agent-based methods (Section 4.7), which studies the impact of tool accuracy.*

4.1.5 Implementation. For GPT-based models, we used OpenAI’s API to infer and track token consumption. For ChatTS and QWen-based models, the training and inference are conducted locally on 8×A800 GPUs. The token consumption for ChatTS is calculated after the “Reorder & Concat” step.

4.2 RQ1. Alignment Tasks

The evaluation results on alignment tasks are shown in Table 3. ChatTS consistently outperforms all baseline models across nearly all tasks and datasets, achieving 46.0%–75.9% improvement in categorical metrics and 80.7%–112.7% in numerical metrics compared to industry-leading models like GPT-4o. This demonstrates that synthetic training data can effectively enable strong alignment with real-world time series.

Among the baselines, GPT-4o (Vision) performs best, suggesting vision-based MLLMs possess some capability to analyze shape characteristics of time series, though they remain limited by image resolution when interpreting details. Text-based methods struggle with the constraints of prompt length, while agent-based approaches performed below expectations (see Section 4.7 for detailed analysis).

ChatTS’s advantages are particularly pronounced in multivariate tasks, where text-based models face challenges with excessively long prompts and vision-based models struggle to distinguish features across multiple time series plotted simultaneously. In contrast, ChatTS’s context-aware time series encoding accurately analyzes referenced time series based on contextual information.

Table 3: Comparison of different models in terms of performance and cost of input tokens on alignment tasks (*image tokens are converted in some models according to price). “Cate.” and “Num.” denotes categorical and numerical tasks respectively. F1-Score and relative accuracy are used in evaluating categorical and numerical tasks, respectively.

Dataset	Type	Model	Trend		Season		Noise		Local		Corr.	Clus.	Overall		Tokens	Est. Cost
			Task	Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Cate.	Cate.	Num.	\$
A	Text	GPT-4o-mini	0.585	0.752	0.649	0.264	0.952	0.312	0.263	0.187	0.357	0.254	0.464	0.310	1.3M	0.20
		GPT-4o	0.585	0.882	0.811	0.768	0.905	0.153	0.379	0.256	0.476	0.333	0.542	0.371	1.3M	3.25
		GPT-4-Turbo	0.526	0.699	0.649	0.131	0.900	0.339	0.303	0.247	0.417	0.269	0.490	0.353	1.3M	13.0
		QWen2.5-14B	0.707	0.709	0.622	0.205	0.833	0.231	0.137	0.099	0.571	0.349	0.464	0.241	1.3M	0.35
	Vision	GPT-4o-mini	0.610	0.501	0.432	0.205	0.667	0.201	0.242	0.184	0.357	0.330	0.404	0.248	2.2M*	0.33
		GPT-4o	0.659	0.613	0.811	0.559	0.810	0.248	0.537	0.414	0.476	0.480	0.609	0.436	0.13M*	0.32
	Agent	GPT-4o-mini	0.559	0.773	0.595	0.270	0.714	0.105	0.400	0.212	0.381	0.361	0.469	0.309	3.0M	0.45
		GPT-4o	0.537	0.650	0.405	0.000	0.595	0.088	0.232	0.136	0.429	0.417	0.390	0.220	2.7M	6.75
	TS	ChatTS	0.927	0.874	0.973	0.849	0.857	0.511	0.895	0.805	0.905	0.782	0.889	0.788	0.08M	0.02
B	Text	GPT-4o-mini	0.619	0.716	0.711	0.317	0.427	0.198	0.145	0.091	0.335	0.269	0.336	0.217	4.5M	0.67
		GPT-4o	0.690	0.825	0.732	0.474	0.573	0.331	0.191	0.136	0.324	0.281	0.366	0.284	4.5M	11.3
		GPT-4-Turbo	0.667	0.732	0.667	0.345	0.348	0.067	0.188	0.133	0.438	0.369	0.385	0.259	4.5M	45.0
		QWen2.5-14B	0.711	0.669	0.705	0.217	0.256	0.094	0.111	0.082	0.402	0.276	0.339	0.193	4.5M	1.22
	Vision	GPT-4o-mini	0.679	0.240	0.814	0.453	0.305	0.238	0.141	0.081	0.327	0.307	0.347	0.142	11.4M*	1.71
		GPT-4o	0.702	0.361	0.938	0.589	0.610	0.398	0.375	0.265	0.367	0.389	0.472	0.311	0.56M*	1.40
	Agent	GPT-4o-mini	0.612	0.591	0.455	0.605	0.375	0.000	0.043	0.022	0.654	0.585	0.372	0.125	8.5M	1.27
		GPT-4o	0.532	0.586	0.619	0.658	0.391	0.262	0.551	0.287	0.500	0.464	0.490	0.370	7.2M	10.8
	TS	ChatTS	0.976	0.902	1.000	0.930	0.927	0.572	0.828	0.752	0.818	0.834	0.862	0.787	0.34M	0.09

Table 4: Reasoning tasks. Inductive Reasoning is in the form of Q&A, evaluated with RAGAS. Other tasks are MC or T/F questions, which are evaluated with accuracy.

Type	Model	Induct.	Deduct.	Causal	MCQ2	Average
Text	GPT-4o-mini	0.333	0.326	0.576	0.480	0.429
	GPT-4o	0.336	0.628	0.685	0.470	0.530
	GPT-4-Turbo	0.280	0.581	0.644	0.490	0.499
	QWen2.5-14B	0.184	0.605	0.348	0.320	0.364
Vision	GPT-4o-mini	0.323	0.442	0.495	0.480	0.435
	GPT-4o	0.322	0.605	0.652	0.490	0.517
Agent	GPT-4o-mini	0.219	0.357	0.692	0.340	0.402
	GPT-4o	0.167	0.553	0.696	0.380	0.449
TS	ChatTS	0.518	0.744	0.804	0.600	0.667

From the efficiency perspective, ChatTS’s native multimodal encoding requires significantly fewer tokens to represent time series data, resulting in much lower costs compared with the baselines (see Table 3). This shows both the effectiveness and efficiency of treating time series as a native modality.

4.3 RQ2. Reasoning Tasks

The comparison results of our model and the baseline models for Reasoning Tasks are shown in Table 4. Reasoning tasks are typically more complex and better aligned with real-world application scenarios than alignment tasks. It can be found that ChatTS achieves consistent improvements over the baseline models across all reasoning tasks. In the Inductive Reasoning task, ChatTS achieved a 34.5% improvement compared to the baseline models, indicating that ChatTS can accurately associate time series attributes with their physical meanings in the real world. This demonstrates that the proposed attribute-based time series generation effectively enables the model to understand the patterns of the physical world

reflected in time series. Moreover, ChatTS also achieved notable improvements in other reasoning tasks, which indicates that even with only synthetic training data, the model can be equipped with good reasoning capabilities related to time series. This further demonstrates the effectiveness of the proposed attribute-based time series generation method and TSEvol.

4.4 RQ3. Studies of Synthetic Training Data

To evaluate the effectiveness of attribute-based time series generation and TSEvol, we conducted ablation studies with two variants: (1) *w/o Attribute-Based*, where all training datasets were replaced by GPT-generated datasets from [43], containing time series directly generated using GPT-produced Python code with corresponding GPT-generated Q&As; (2) *w/o TSEvol*, where SFT datasets were replaced with data directly generated using an LLM without the evolutionary approach, though with prompts designed to encourage diversity. Both variants included the instruct-following dataset to ensure fair comparison.

The evaluation results in Figures 9 and 10 reveal that models trained on GPT-generated data performed significantly worse across alignment tasks, particularly for local fluctuation detection and numerical analysis. This suggests the attribute-based generation method better captures precise feature details and numerical values. Meanwhile, models trained with TSEvol demonstrated substantial improvements in reasoning capabilities and modest gains in alignment tasks, indicating that TSEvol effectively diversifies question formats and generates tailored Q&As for different time series attributes, enhancing overall model performance.

4.5 RQ4. Scaling of Training Dataset

Figure 11a illustrates the relationship between ChatTS performance and training data size. The results show that increasing the Phase 1

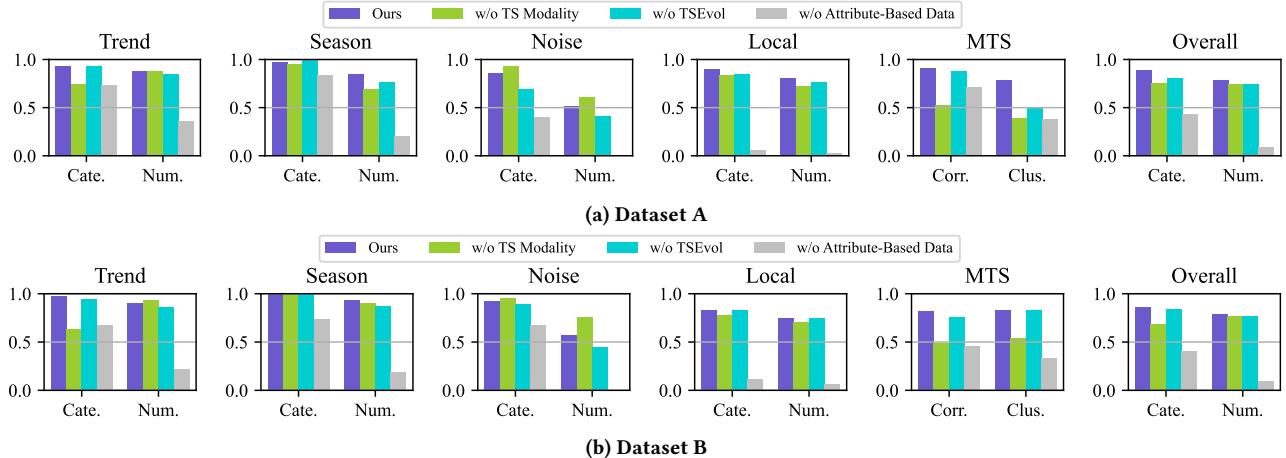


Figure 9: Ablation studies on alignment tasks.

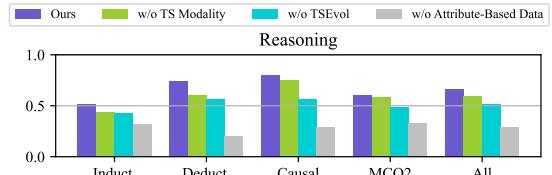


Figure 10: Ablation studies on reasoning tasks.

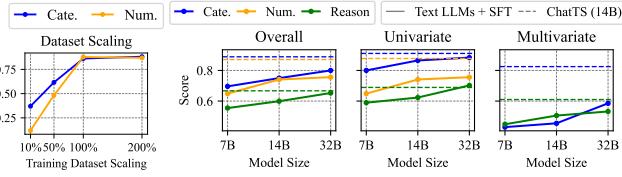


Figure 11: Scaling of training dataset and text-based LLMs.

training dataset size from 10% to 100% of the current size significantly improves performance, but further expansion yields minimal gains. Thus, our chosen training set size is well-balanced, ensuring sufficient data for effective alignment while avoiding too much resource consumption during training.

4.6 RQ5. Study of Time Series Modality

To investigate the effectiveness of the time series multimodality in ChatTS, we performed an ablation study based on a text-only version of ChatTS (w/o TS Modality). We remove the time series encoder in ChatTS (*i.e.* using the original QWen-2.5 model) and use the same training data with ChatTS (the time series arrays are encoded into text) in model training. The experimental results are shown in Figure 9 and Figure 10. Overall, the model using only the text modality performs significantly worse than the original ChatTS model. This indicates that encoding multimodal information is crucial for accurately capturing both shape and numerical information. However, in certain sub-evaluation metrics (e.g., noise), the text-only model outperforms the multimodal ChatTS, suggesting that text modality models still have strong capabilities for identifying small fluctuations. In MTS tasks, the text-only model is nearly incapable of answering any questions. This implies that even with

extensive multivariate training data, text-only LLMs still struggle to handle multivariate problems due to excessively long context lengths because of severe hallucinations and inaccurate responses. Additionally, to compare the performance gains between text-based LLMs and TS-MLLMs, we fine-tuned various sizes of the Qwen2.5 series text-based LLMs using the text version of the ChatTS training dataset (as shown in Figure 11b). Experimental results indicate that even when fine-tuning the larger Qwen2.5-32B text model, the results still do not outperform those of ChatTS (14B), which has native multimodal capabilities. This further validates the importance of native multimodal capabilities in ChatTS, whether in the accuracy for MTS analysis or cost efficiency (see Table 3).

4.7 RQ6. Study of Agent-Based Methods

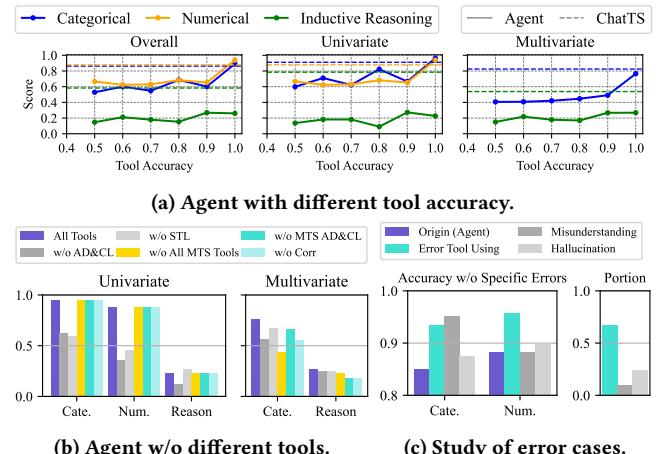


Figure 12: Agent with “Perfect Tools”. Even with perfect tools, Agent still makes errors (e.g., Error Tool Using)

Agent-based methods are widely applied but showed suboptimal performance in our evaluations (RQ1, RQ2) due to several main issues: (1) tool inaccuracy (2) error tool use, and (3) response formatting that caused parsing failures. To explore their performance upper bound, we conducted detailed analyses:

- (1) **Parsing Failures:** We exclude responses that failed to parse, ensuring all outputs were valid.
- (2) **Perfect Tools:** We design “perfect tools” with controlled accuracy through time series labels in the synthetic dataset (Figure 12a). The accuracy of the perfect tools can be strictly controlled.
- (3) **Tool Ablation:** We perform ablation studies (Figure 12b) to evaluate the impact of individual tools on accuracy.
- (4) **Error Analysis:** We categorize agent errors into three types: Error Tool Using, Misunderstanding, and Hallucination. We analyze their impact on performance (Figure 12c).

Our sensitivity analysis (Figure 12a) shows that agent performance is highly sensitive to tool accuracy, especially in the [0.9, 1.0] range. For categorical and numerical tasks, the agent with perfect tools slightly outperforms ChatTS on UTS tasks but lags behind in MTS tasks. For agents, MTS tasks typically require more tool calls and reasoning, which places higher demands on LLMs’ tool-using and summarization capabilities. In contrast, ChatTS processes multiple time series natively, reducing complexity and improving accuracy for MTS tasks. The tool ablation study (Figure 12b) shows that Agent depends heavily on both tool precision and completeness, particularly for MTS tasks. Even with all tools available, agents frequently fail to invoke the correct tool at the right time (e.g., using the classification tool rather than the anomaly detection tool to identify the position of a spike), limiting their effectiveness. Error analysis (Figure 12c) reveals “Error Tool Using” as the largest source of errors. When these cases are excluded, agent accuracy exceeds 95%, surpassing ChatTS. This validates the correctness of the implementation of perfect tools and the model, which also shows their limitations: Agents may struggle with tool selection and reasoning.

In summary, while perfect tools improve agent performance, challenges such as tool selection errors, misunderstandings, and hallucinations persist, leaving agents less effective than ChatTS for complex time series tasks.

5 CASE STUDIES AND APPLICATIONS

5.1 Case Studies on Real-World Data

To investigate the performance of ChatTS on *real-world* time series, we perform several case studies with challenging questions, the results are shown in Figure 13.

5.1.1 Shape and Statistical Analysis. The “Basic Shape Analysis” case demonstrates ChatTS’s capability to analyze an NYC taxi passenger time series with complex periodic fluctuations and local anomalies. ChatTS accurately identifies multiple trend segments, the periodicity, and the upward spike along with their amplitudes. This shows ChatTS’s capability to capture both global patterns and localized features. In the “Statistic Analysis” case, ChatTS analyzes advertisement CPC data with misleading scaling. Despite potential confusion in the minimum value, ChatTS correctly identifies the max/min values and their positions. These cases show ChatTS’s robustness in statistical analysis in complex real-world time series.

5.1.2 OOD Fluctuation Recognition. The “OOD Fluctuation” case presents ChatTS with a traffic occupancy time series containing an OOD fluctuation pattern *absent from its training data*. However, ChatTS accurately describes it as a “Convex-Shaped Elevation”, characterized by a gradual rise followed by a sharper decline along

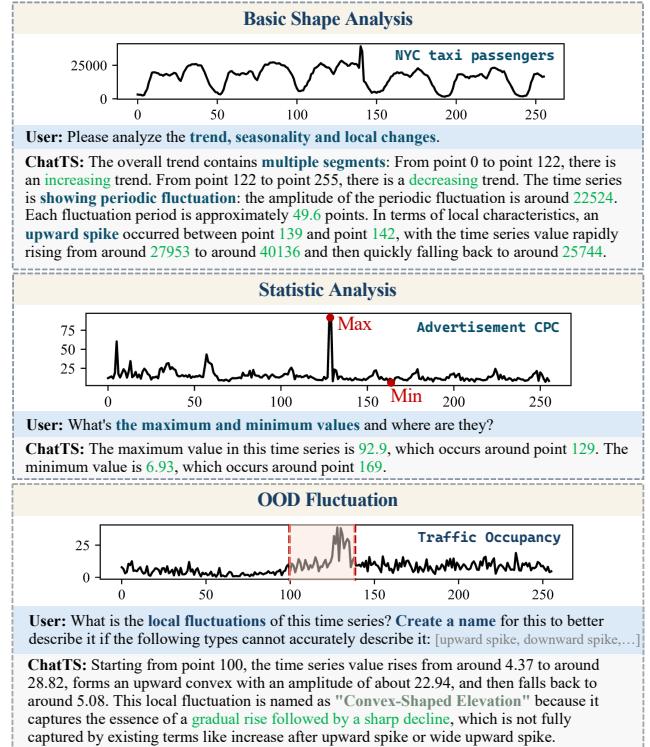


Figure 13: Case studies on real-world time series data.

with the overall shape of a convex. This demonstrates ChatTS’s inherent understanding of time series patterns themselves, rather than simply repeating representations from the training set. This indicates that ChatTS has a certain capacity to generalize to real-world data despite being trained exclusively on synthetic data.

5.2 Real-World Application: DB Operation

To illustrate the performance of ChatTS with its native time series multimodal capability in real-world applications, we present a typical Oracle DB operation application through an MTS-related multi-turn dialogue with ChatTS. In this case study, an Oracle DB operator has identified a recent anomaly and retrieves several time series metrics from the monitoring system, inputting them into ChatTS for analysis (as shown in Figure 14)². By querying ChatTS, the operator obtains the names of all metrics with anomalies. Then, to accurately pinpoint the root cause, the operator provides ChatTS with a textual document titled “Oracle Database Troubleshooting Rulebook” and requests ChatTS to analyze the root cause and propagation of the system failure step-by-step, combining insights from the rulebook and the time series anomalies. Notably, the rulebook is entirely in *text* form, without a strictly structured format, which is helpful for the operators to share their expert experience effectively. The responses of ChatTS show that it can accurately identify anomalies and amplitudes in multivariate time series. By leveraging “the metric with the largest fluctuation” in the rulebook, ChatTS can further reason about the root cause and failure propagation path.

²DB Metrics can be input into ChatTS through API. The system implementation details are out of the scope of this paper.

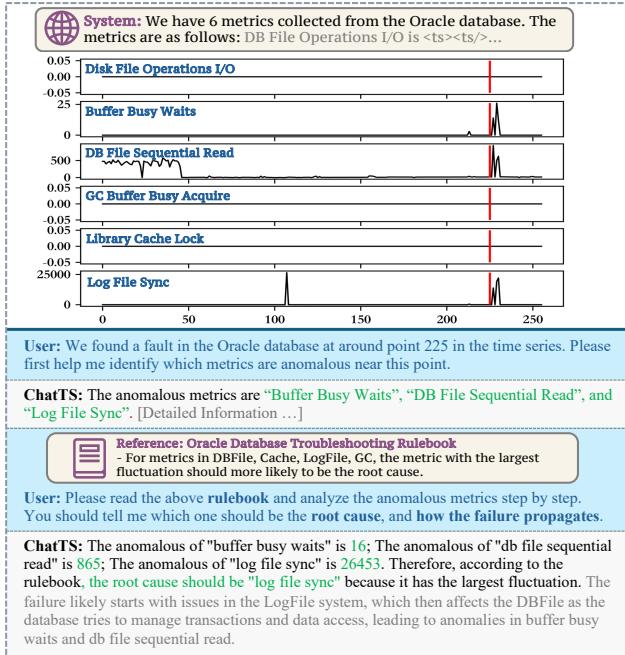


Figure 14: An application case of ChatTS in a failure diagnosis with an Oracle database system.

This further shows that ChatTS can effectively utilize its *alignment capability* to analyze time series and perform complex analysis in real-world applications with its robust *reasoning ability*.

5.3 Real-World Application: Detailed Analysis

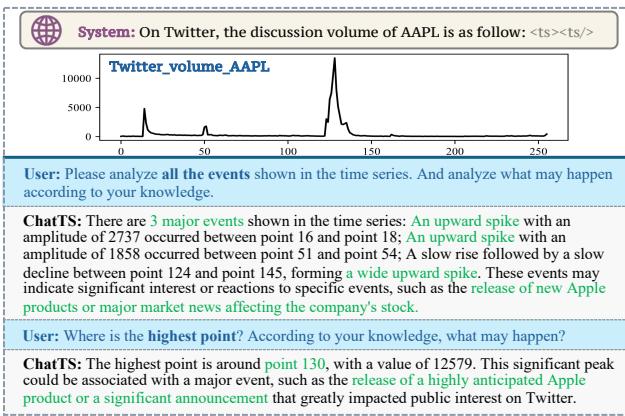
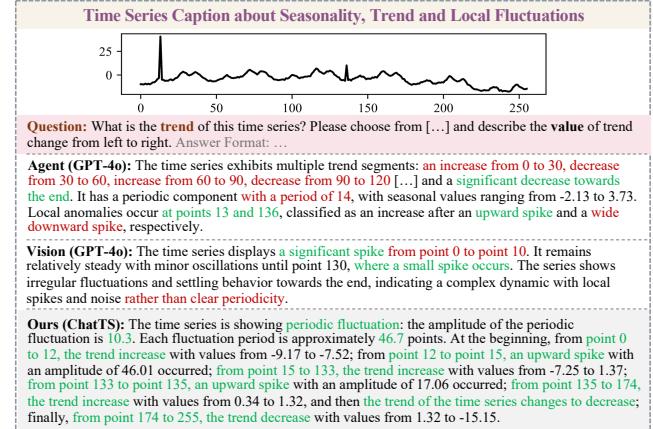


Figure 15: An ChatTS application case in detailed time series.

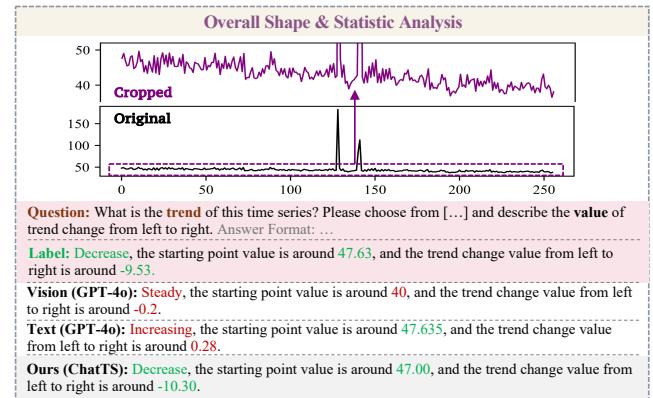
Another typical application of ChatTS is conducting a detailed analysis of time series features, combined with LLMs' knowledge and reasoning capabilities to perform simple reasoning and question answering. In Figure 15, we present a case study of time series analysis on the discussion intensity of AAPL-related topics on Twitter, using data from NAB [7]. Notably, even without explicit instructions from the user to identify local fluctuations, ChatTS can accurately infer the user's intent and determine the timestamps of

all three "hot events" from the time series. Furthermore, ChatTS can precisely identify the highest point and its position in the time series based on the numerical values of the local peaks and perform event analysis according to the physical meaning of the series. This demonstrates that ChatTS can accurately recognize both shape and numerical characteristics of time series and perform reasoning and analysis based on vague user input.

5.4 Baseline Comparison



(a) Case Study on Complex Time Series Caption



(b) Case Study on Trend

Figure 16: Case studies on different types of LLMs in time series alignment tasks.

5.4.1 Seasonality, Trend and Fluctuations. As shown in Figure 16a, due to tool inaccuracy, the Agent fails to identify the periodicity. This further led the Agent to misinterpret periodic patterns as different trend changes, resulting in errors in trend analysis. Moreover, the LLM does not realize the problem or attempt to correct it. Similarly, the Vision-based model also exhibited errors in analyzing local fluctuations and periodicity. In contrast, ChatTS, with its time-series modality awareness, accurately captures the periodicity and trend transitions. This case shows a key limitation of agent-based tools: the precision of tools alone cannot overcome the cascading errors caused by the initial error of time series patterns.

5.4.2 Detailed Trend Analysis. Figure 16b presents a misleading case where the original image suggests a steady trend. However, in the cropped plot that ignores the two spikes, there is a significant decreasing trend. Due to the subtle nature of trends displayed in the time series images, the Vision-based model incorrectly classified them as steady. Similarly, text-based models, while identifying starting values, fail to identify the global shape of the time series. In contrast, ChatTS captures both the overall trend and numerical details accurately due to its native time series encoding capabilities.

6 RELATED WORK

Multimodal LLMs (MLLMs). MLLMs have developed rapidly in recent years and found extensive applications [58, 62]. A significant body of research integrates different types of data to achieve multimodal fusion, including images [9, 31, 37], videos [32, 41, 63], audio [16, 48], and graphs [45, 64]. These models have been applied across diverse domains, with image-based question answering and reasoning representing an important research direction. Many studies leverage vision-based LLMs for image reasoning tasks [25, 40], fully utilizing the natural language understanding and reasoning capabilities of large language models. However, in the field of time series, despite the existence of numerous works (as discussed below) that combine time series data with LLMs, research on aligning LLMs with time-series modalities with time-series modalities remains limited. This limitation is primarily due to the scarcity of high-quality multimodal datasets that combine time series with textual information [15, 27, 43]. As a result, the development of time series-specific MLLMs for question-answering and reasoning tasks has lagged behind other modalities.

Time Series Question Answering (TSQA). With the rapid development of LLMs, TSQA systems have combined the reasoning capabilities of LLMs with time series analysis to enable more efficient cross-domain decision-making and complex task handling [27]. Time series question-answering systems have been explored in various fields, such as AIOps [52, 68], IoT [22, 54], healthcare [44, 60], finance [28, 42], and traffic [17, 29]. However, these methods are often limited to agent-based [57] and retrieval-augmented generation (RAG) [30] approaches, lacking a comprehensive understanding of time series and sufficient reasoning capabilities. Although some recent studies [15] have attempted to leverage temporal multimodal approaches for time series reasoning tasks, they typically rely on task-specific corpora. They are trained and evaluated on specific tasks (e.g., classification tasks or forecasting tasks), lacking multivariate analysis capabilities. Compared to the research on multimodal question answering in fields like images and videos, time series question answering still lacks robust multimodal alignment methods and evaluation frameworks [10, 43]. Therefore, in contrast to existing studies, this paper is the first to propose a comprehensive time series modality alignment and fine-tuning process, evaluated using multiple alignment and reasoning tasks.

LLM + Time Series. In addition to the research above, many studies have combined LLMs with time series for various downstream tasks, leveraging the powerful capabilities of LLMs [11–13, 23, 26, 38, 51, 67]. However, while these models are using LLMs as backbones, they are designed for specific downstream tasks and lack language alignment capabilities, making them unsuitable for

question answering and reasoning applications. Moreover, some studies employ vision-based multimodal LLMs for time series prediction [14] and anomaly detection [69]. This approach aligns with the vision-based LLM methods discussed in this paper but is significantly constrained in its ability to analyze time series.

7 LIMITATION AND FUTURE WORK

Due to the limited existing research on time series understanding and reasoning, although ChatTS has explored an effective approach, we believe it still has a number of limitations. First, while our experiments demonstrate that synthetic data can achieve satisfactory alignment and reasoning performance, we believe that real-world data is essential for further enhancing the capabilities of TS-MLLMs. We hope more relevant datasets will emerge in the future. Second, although we found that a simple MLP encoder performs well due to the relatively simple structure of time series data, exploring more effective methods for multimodal encoding and integration remains a valuable research direction. Third, despite labeling hundreds of real-world time series and using 14 evaluation metrics for evaluation, we believe that this is still insufficient for a comprehensive evaluation of TS-MLLMs. More labeled real-world data is needed for a more comprehensive evaluation. Finally, while this work focuses on *understanding tasks* like language alignment and reasoning, MLLM-based time series *generation* is also worth exploring. Thus, developing a multimodal model that can generate time series based on textual input is an important area for future research.

8 CONCLUSION

Understanding and reasoning are important for real-world time series applications, but research is limited due to the lack of time series-text data. In this paper, we propose ChatTS, the first TS-MLLM with multivariate time series as input for complex time series QA and reasoning, which is fine-tuned on synthetic data. We introduce an attribute-based time series generation method, which not only generates diverse time series but also provides complete and precise attribute descriptions. Building on this, we further propose TSEvol, which leverages rich attribute combinations from the attribute pool and Evol-Instruct to generate diverse and accurate QAs, enhancing the model’s capabilities in complex question answering and reasoning. To comprehensively evaluate the capabilities of our model, we collect datasets that include real-world time series data, covering the evaluation of both alignment tasks and reasoning tasks. Evaluation results show that our model achieves significant improvements, outperforming baselines by 46.0% in alignment tasks and 25.8% in reasoning tasks. These findings demonstrate the effectiveness of our approach in bridging the gap between time series data and natural language understanding. We have open-sourced the source code, trained model weights, and the evaluation datasets for reproduction and future research: <https://github.com/NetManAIOps/ChatTS>.

REFERENCES

- [1] 2019. Anomaly Detection Toolkit. <https://github.com/arundo/adtk>
- [2] 2020. DeepSpeed. <https://www.deepspeed.ai/>
- [3] 2023. Weather Dataset. <https://www.bgc-jena.mpg.de/wetter/>
- [4] 2024. MCQ2 Dataset. <https://github.com/behavioral-data/TSandLanguage>
- [5] 2024. OpenAI GPT-4o. <https://openai.com/index/hello-gpt-4o/>

- [6] 2024. Qwen2.5-14B-Instruct Model. <https://huggingface.co/Qwen/Qwen2.5-14B-Instruct>
- [7] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuhu Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147.
- [8] Sarah Alnegheimish, Linh Nguyen, Laure Berti-Equille, and Kalyan Veeramachaneni. 2024. Large language models can be zero-shot anomaly detectors for time series? *arXiv preprint arXiv:2405.14755* (2024).
- [9] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966* (2023).
- [10] Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. 2024. Time-SeriesExam: A time series understanding exam. *arXiv preprint arXiv:2410.14752* (2024).
- [11] Yifu Cai, Mononito Goswami, Arjun Choudhry, Arvind Srinivasan, and Artur Dubrawski. 2023. Jolt: Jointly learned representations of language and time-series. In *Deep Generative Models for Health Workshop NeurIPS 2023*.
- [12] Defu Cao, Furong Jia, Sercan 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* (2023).
- [13] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469* (2023).
- [14] Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. 2024. VisionTS: Visual Masked Autoencoders are Free-Lunch Zero-Shot Time Series Forecasters. *arXiv preprint arXiv:2408.17253* (2024).
- [15] 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* (2024).
- [16] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919* (2023).
- [17] Longchao Da, Kuanru Liou, Tiejin Chen, Xuesong Zhou, Xiangyong Luo, Yezhou Yang, and Hua Wei. 2024. Open-ti: Open traffic intelligence with augmented language model. *International Journal of Machine Learning and Cybernetics* (2024), 1–26.
- [18] Angus Dempster, François Fleuret, and Geoffrey I Webb. 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* 34, 5 (2020), 1454–1495.
- [19] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ra-gas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217* (2023).
- [20] Elizabeth Fons, Rachneet Kaur, Soham Palande, Zhen Zeng, Tucker Balch, Manuela Veloso, and Svitlana Vyetrenko. 2024. Evaluating Large Language Models on Time Series Feature Understanding: A Comprehensive Taxonomy and Benchmark. *arXiv preprint arXiv:2404.16563* (2024).
- [21] Fanzhe Fu, Junru Chen, Jing Zhang, Carl Yang, Lvbin Ma, and Yang Yang. 2024. Are Synthetic Time-series Data Really not as Good as Real Data? *arXiv preprint arXiv:2402.00607* (2024).
- [22] Simone Gallo, Fabio Paterno, and Alessio Malizia. 2023. Conversational interfaces in iot ecosystems: where we are, what is still missing. In *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*. 279–293.
- [23] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems* 36 (2024).
- [24] Xiao He, Ye Li, Jian Tan, Bin Wu, and Feifei Li. 2023. OneShotSTL: One-Shot Seasonal-Trend Decomposition For Online Time Series Anomaly Detection And Forecasting. *Proc. VLDB Endow.* 16, 6 (2023), 1399–1412. <https://doi.org/10.14778/3583140.3583155>
- [25] Ding Jiang and Mang Ye. 2023. Cross-modal implicit relation reasoning and aligning for text-to-image person retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2787–2797.
- [26] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-lm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728* (2023).
- [27] Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. 2024. Position paper: What can large language models tell us about time series analysis. *arXiv preprint arXiv:2402.02713* (2024).
- [28] Litton Jose Kurisinkel, Pruthwik Mishra, and Yue Zhang. 2024. Text2timeseries: Enhancing financial forecasting through time series prediction updates with event-driven insights from large language models. *arXiv preprint arXiv:2407.03689* (2024).
- [29] Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. 2023. Large language models as traffic signal control agents: Capacity and opportunity. *arXiv preprint arXiv:2312.16044* (2023).
- [30] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.
- [32] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhui Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355* (2023).
- [33] Li Li, Xiaonan Su, Yi Zhang, Yuetong Lin, and Zhiheng Li. 2015. Trend modeling for traffic time series analysis: An integrated study. *IEEE Transactions on Intelligent Transportation Systems* 16, 6 (2015), 3430–3439.
- [34] Zeyan Li, Nengwen Zhao, Mingjie Li, Xianglin Lu, Lixin Wang, Dongdong Chang, Xiaohui Nie, Li Cao, Wenchang Zhang, Kaixin Sui, et al. 2022. Actionable and interpretable fault localization for recurring failures in online service systems. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 996–1008.
- [35] Zeyan Li, Nengwen Zhao, Shenglin Zhang, Yongqian Sun, Pengfei Chen, Xidao Wen, Minghua Ma, and Dan Pei. 2022. Constructing large-scale real-world benchmark datasets for aiops. *arXiv preprint arXiv:2208.03938* (2022).
- [36] Bryan Lim and Stefan Zohren. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* 379, 2194 (2021), 20200209.
- [37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems* 36 (2024).
- [38] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. 2024. Time-MMD: A New Multi-Domain Multimodal Dataset for Time Series Analysis. *arXiv preprint arXiv:2406.08627* (2024).
- [39] Dongsheng Luo, Wei Cheng, Yingheng Wang, Dongkuan Xu, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Yanchi Liu, Yuncong Chen, Haifeng Chen, et al. 2023. Time series contrastive learning with information-aware augmentations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4534–4542.
- [40] Run Luo, Haonan Zhang, Longze Chen, Ting-En Lin, Xiong Liu, Yuchuan Wu, Min Yang, Minzheng Wang, Pengpeng Zeng, Lianli Gao, et al. 2024. Mmevol: Empowering multimodal large language models with evol-instruct. *arXiv preprint arXiv:2409.05840* (2024).
- [41] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424* (2023).
- [42] Elliot Maître, Zakaria Chemli, Max Chevalier, Bernard Douillet, Jean-Philippe Gitté, and Olivier Teste. 2020. Event detection and time series alignment to improve stock market forecasting. In *Joint conference of the information retrieval communities in europe (circle 2020)*, Vol. 2621. CEUR-WS. org, 1–5.
- [43] Mike A Merrill, Mingfai Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. 2024. Language Models Still Struggle to Zero-shot Reason about Time Series. *arXiv preprint arXiv:2404.11757* (2024).
- [44] Jungwoo Oh, Gyubok Lee, Seongsu Bae, Joon-myung Kwon, and Edward Choi. 2024. Ecg-q: A comprehensive question answering dataset combined with electrocardiogram. *Advances in Neural Information Processing Systems* 36 (2024).
- [45] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [46] Robert B Penfold and Fang Zhang. 2013. Use of interrupted time series analysis in evaluating health care quality improvements. *Academic pediatrics* 13, 6 (2013), S38–S44.
- [47] CLEVELAND RB. 1990. STL: A seasonal-trend decomposition procedure based on loess. *J Off Stat* 6 (1990), 3–73.
- [48] Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quirye, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925* (2023).
- [49] Neil Savage. 2023. Synthetic data could be better than real data. *Nature* (2023).
- [50] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing* 90 (2020), 106181.
- [51] Jing Su, Chufeng Jiang, Xin Jin, Yuxin Qiao, Tingsong Xiao, Hongda Ma, Rong Wei, Zhi Jing, Jiajun Xu, and Junhong Lin. 2024. Large language models for forecasting and anomaly detection: A systematic literature review. *arXiv preprint arXiv:2402.10350* (2024).
- [52] Zefan Wang, Zichuan Liu, Yingying Zhang, Aoxiao Zhong, Jihong Wang, Fengbin Yin, Lunting Fan, Lingfei Wu, and Qingsong Wen. 2024. Ragent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 4966–4974.
- [53] Yemane Wolde-Rufael. 2006. Electricity consumption and economic growth: a time series experience for 17 African countries. *Energy policy* 34, 10 (2006),

- 1106–1114.
- [54] Tianwei Xing, Luis Garcia, Federico Cerutti, Lance Kaplan, Alun Preece, and Mani Srivastava. 2021. DeepSQA: Understanding sensor data via question answering. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 106–118.
 - [55] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhuan Feng, Chongyang Tao, and Dixin Jiang. 2023. WizardLM: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244* (2023).
 - [56] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Cheng-peng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024).
 - [57] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).
 - [58] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review* (2024), nwae403.
 - [59] Luke Yoffe, Alfonso Amayuelas, and William Yang Wang. 2024. DebUnc: mitigating hallucinations in large language model agent communication with uncertainty estimations. *arXiv preprint arXiv:2407.06426* (2024).
 - [60] Han Yu, Peikun Guo, and Akane Sano. 2023. Zero-shot ECG diagnosis with large language models and retrieval-augmented generation. In *Machine Learning for Health (ML4H)*. PMLR, 650–663.
 - [61] Chi Zhang, Sammukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2018. Generative adversarial network for synthetic time series data generation in smart grids. In *2018 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm)*. IEEE, 1–6.
 - [62] Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. 2024. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601* (2024).
 - [63] Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858* (2023).
 - [64] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. In *Proceedings of the ACM on Web Conference 2024*. 1003–1014.
 - [65] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyen Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Bangkok, Thailand. <http://arxiv.org/abs/2403.13372>
 - [66] Zhenyu Zhong, Qiliang Fan, Jiacheng Zhang, Minghua Ma, Shenglin Zhang, Yongqian Sun, Qingwei Lin, Yuzhi Zhang, and Dan Pei. 2023. A Survey of Time Series Anomaly Detection Methods in the AIOps Domain. *arXiv preprint arXiv:2308.00393* (2023).
 - [67] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.
 - [68] Xuanhe Zhou, Guoliang Li, Zhaoyan Sun, Zhiyuan Liu, Weize Chen, Jianming Wu, Jiesi Liu, Ruohang Feng, and Guoyang Zeng. 2023. D-bot: Database diagnosis system using large language models. *arXiv preprint arXiv:2312.01454* (2023).
 - [69] Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, Ruiqi Wang, Jiawei Zhang, and Yuan-tao Gu. 2024. See it, Think it, Sorted: Large Multimodal Models are Few-shot Time Series Anomaly Analyzers. *arXiv preprint arXiv:2411.02465* (2024).