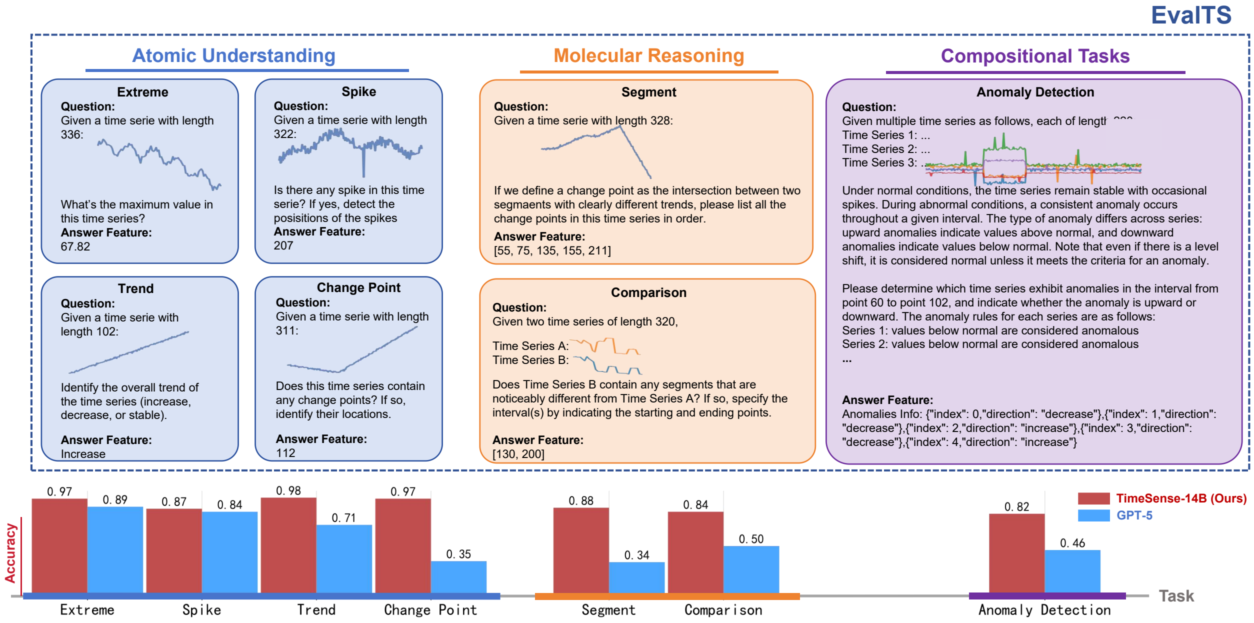
TIMESENSE: MAKING LARGE LANGUAGE MODELS PROFICIENT IN TIME-SERIES ANALYSIS

Anonymous authors
Paper under double-blind review



Extreme
Question:
Given a time serie with length 336:
What's the maximum value in this time series?
Answer Feature:
67.82

Spike
Question:
Given a time serie with length 322:
Is there any spike in this time serie? If yes, detect the posisitions of the spikes
Answer Feature:
207

Trend
Question:
Given a time serie with length 102:
Identify the overall trend of the time series (increase, decrease, or stable).
Answer Feature:
Increase

Change Point
Question:
Given a time serie with length 311:
Does this time series contain any change points? If so, identify their locations.
Answer Feature:
112

Segment
Question:
Given a time serie with length 328:
If we define a change point as the intersection between two segmaents with clearly different trends, please list all the change points in this time series in order.
Answer Feature:
[55, 75, 135, 155, 211]

Comparison
Question:
Given two time series of length 320,
Time Series A:
Time Series B:
Does Time Series B contain any segments that are noticeably different from Time Series A? If so, specify the interval(s) by indicating the starting and ending points.
Answer Feature:
[130, 200]

Anomaly Detection
Question:
Given multiple time series as follows, each of lengtl
Time Series 1: ...
Time Series 2: ...
Time Series 3: ...

Under normal conditions, the time series remain stable with occasional spikes. During abnormal conditions, a consistent anomaly occurs throughout a given interval. The type of anomaly differs across series: upward anomalies indicate values above normal, and downward anomalies indicate values below normal. Note that even if there is a level shift, it is considered normal unless it meets the criteria for an anomaly.

Please determine which time series exhibit anomalies in the interval from point 60 to point 102, and indicate whether the anomaly is upward or downward. The anomaly rules for each series are as follows:
Series 1: values below normal are considered anomalous
Series 2: values below normal are considered anomalous
...

Answer Feature:
Anomalies Info: {"index": 0,"direction": "decrease"},{"index": 1,"direction": "decrease"},{"index": 2,"direction": "increase"},{"index": 3,"direction": "decrease"},{"index": 4,"direction": "increase"}

Figure 1: Examples from the EvalTS benchmark and performance comparison between our model and GPT-5.

ABSTRACT

In the time-series domain, an increasing number of works combine text with temporal data to leverage the reasoning capabilities of large language models(LLMs) for various downstream time-series understanding tasks. This enables a single model to flexibly perform tasks that previously required specialized models for each domain. However, these methods typically rely on text labels for supervision during training, biasing the model toward textual cues while potentially neglecting the full temporal features. Such a bias can lead to outputs that contradict the underlying time-series context. To address this issue, firstly, we construct the EvalTS benchmark, comprising 10 tasks across three difficulty levels, from fundamental temporal pattern recognition to complex real-world reasoning, to evaluate models under more challenging and realistic scenarios. We also propose TimeSense, a multimodal framework that makes LLMs proficient in time-series analysis by balancing textual reasoning with a preserved temporal sense. TimeSense incorporates a Temporal Sense module that reconstructs the input time-series within the model's context, ensuring that textual reasoning is grounded in the time-series dynamics. Moreover, to enhance spatial understanding of time-series data, we explicitly incorporate coordinate-based positional embeddings, which provide each time point with spatial context and enable the model to capture structural dependencies more effectively. Experimental results demonstrate that TimeSense achieves state-of-the-art performance across multiple tasks, and it particularly outperforms existing methods on complex multi-dimensional time-series reasoning tasks. Our code and data are released at https://anonymous.4open.science/r/timesense-984F.

1 INTRODUCTION

Time series data lie at the core of domains such as electricity, healthcare, traffic, weather, and finance (Nogales et al., 2002; Morid et al., 2023; Lippi et al., 2013; McGovern et al., 2011; Yu et al., 2023). Recent advances in multimodal learning show that combining language with temporal signals can unlock stronger reasoning, echoing the progress seen in vision-language modeling. (Hu et al., 2024; Jin et al., 2023b; Wang et al., 2023a) Time-series multimodal models extend this idea by using large-scale pretraining to perform diverse tasks in a zero-shot setting. (Xie et al., 2024; Wang et al., 2025; Xu et al., 2025b; Kong et al., 2025)

Yet, in many of these models, text serves only as an auxiliary signal to boost performance on predefined temporal tasks, rather than enabling flexible adaptation through context (Jin et al., 2023a; Yang et al., 2025). Natural language, however, is more than structured labels. It offers rich descriptions of temporal patterns and a human-interpretable reasoning process. This motivates the design of multimodal time-series models that not only solve tasks but also explain them in natural language, leading to richer reasoning and more intuitive interaction.

Despite this promise, current approaches face two key challenges:

(i) Datasets remain narrow. Classical time series tasks focus on low-level tasks such as forecasting or classification. While useful, these tasks capture only basic dynamics and cannot test whether a model generalizes to reasoning that links temporal and textual information. Existing reasoning datasets emphasize surface-level alignment between text and time series, but do not test deeper temporal sense or cross-feature reasoning.

(ii) Most models are trained with only textual labels, which biases optimization toward language and weakens temporal modeling. This often produces outputs that contradict the underlying sequence, especially for long series, multi-dimensional dynamics, or local anomalies. A more balanced method is needed—one that preserves temporal sense while still using textual supervision.

To address challenge of datasets, we introduce EvalTS, a benchmark designed to comprehensively assess multimodal time-series models in a dialog-style manner, capturing both their perceptual and reasoning abilities. EvalTS draws inspiration from human temporal understanding, rather than being constrained to fixed tasks or simple alignment-based evaluations common in existing benchmarks, starting from atomic units of temporal cognition and progressively combining them to tackle more complex tasks. To support this evaluation, we propose ChronGen, a controllable rule-based generator that systematically creates multimodal time-series data, filling the gap of data scarcity.

To enable models to better tackle such complex tasks, we propose TimeSense, an architecture that mimics human temporal analysis by encoding each time point's positional information and integrating a Time Sensor module, allowing the model to retain, reason over, and fully exploit temporal information. Together, these contributions enable models to achieve holistic temporal awareness and effectively leverage both temporal and contextual cues for complex time-series tasks.

Together, EvalTS and TimeSense form a unified framework that not only addresses the limitations of prior work but also lays the foundation for systematic evaluation and improvement of temporal reasoning models. In summary, our contributions are threefold:

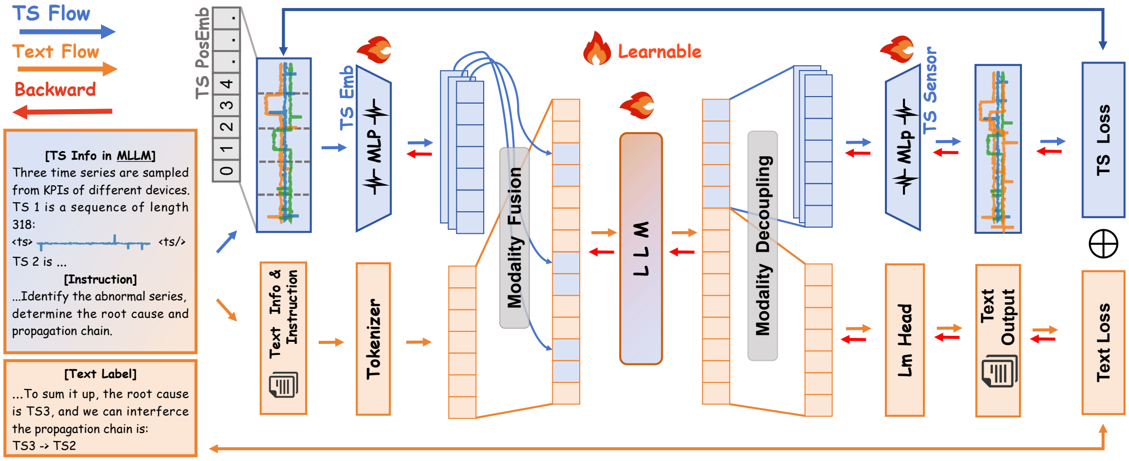
1. We design EvalTS, a benchmark for systematic evaluation across diverse temporal reasoning tasks, going beyond simple alignment tasks and classical temporal tasks to capture a broader spectrum of multimodal time-series analysis challenges.

2. We propose TimeSense, which integrates temporal and textual information through a Temporal Sense module, enabling the model to process contextual text and temporal signals in a balanced, human-like manner and mitigating language bias in multimodal training.

3. We show that TimeSense consistently outperforms existing models across all 10 tasks in EvalTS and remains highly competitive on four additional multiple-choice evaluation datasets, demonstrating its robust and generalizable time-series analysis capabilities.

2 RELATED WORKS

Time-series analysis has progressed from classical statistical and deep learning models to emerging multimodal approaches that integrate language with temporal signals. We first review foundational tasks and methods in time-series analysis, showing how traditional models capture temporal dependencies and patterns. We then discuss recent advances in multimodal time-series modeling, focusing on methods that use large language models (LLMs) to reason over numerical sequences and textual annotations. This review sets the stage for our proposed framework.



[TS Info in MLLM]
Three time series are sampled from KPIs of different devices. TS 1 is a sequence of length 318:
<ts> <ts/>
TS 2 is ...
[Instruction]
...Identify the abnormal series, determine the root cause and propagation chain.

[Text Label]
...To sum it up, the root cause is TS3, and we can interferce the propagation chain is:
TS3 -> TS2

Figure 2: **Workflow of TimeSense.** The time-series related modules are marked with blue lines, and the text-related modules with orange lines.

2.1 Classical Time-Series Tasks and Methods

Research on time series has centered on forecasting, classification, clustering, change-point detection, anomaly detection, and causal inference Zhou et al. (2021); Dempster et al. (2020); Tiwari (2023); Gong et al. (2024). Early methods used statistical models such as ARIMA, exponential smoothing, and state-space models (Zhang, 2003; Holmes et al., 2012; Kalekar et al., 2004), which explicitly encode temporal dependencies. Later, deep learning models such as RNNs, LSTMs, and temporal convolutional networks (Liu et al., 2020; Siami-Namini et al., 2018; He & Zhao, 2019) showed stronger ability to capture non-linear dynamics.

Many of these methods address task-specific temporal patterns. Pattern-based approaches have been common in classification and motif discovery, where recognizing local temporal structures improves accuracy (Ye & Keogh, 2009; Zakaria et al., 2012). Similarly, anomaly detection and change-point detection often depend on identifying recurring or distinctive temporal signatures. Overall, classical methods focus on patterns in temporal data itself, with little use of textual or multimodal signals.

2.2 Time-Series Multimodal Models

In recent years, multimodal large language models (MLLMs) have achieved rapid progress in images, videos, and audio (Zhang et al., 2005; Wang et al., 2024b; Mroueh et al., 2015). These models use natural language understanding to support reasoning, question answering, and decision making.

By contrast, multimodal modeling for time series is still in its early stages. Work in this area is limited by the lack of datasets that pair sequences with text. Emerging time-series MLLMs attempt to connect numerical signals with language, often producing textual rationales or answers. For example, ChatTime Wang et al. (2025) treats time series as a "foreign language," enabling bi-modal inputs and zero-shot forecasting. ChatTS Chow et al. (2024) aligns time-series attributes with LLMs using synthetic descriptions to improve reasoning. ITFormer links a time-series encoder with a frozen LLM and introduces a domain-specific multi-task benchmark, EngineMT-QA, to evaluate temporal-textual reasoning (Xu et al., 2025b;a). Survey papers summarize this fast-moving field and categorize approaches by encoders, prompting strategies, and training objectives (Zhu et al., 2024).

3 Methodology

3.1 Problem Definition

We first present the motivation and the challenges that TimeSense aims to solve, followed by the design details. As shown in Figure 2, the multi-modal architecture consists of two main modules: modality embedding and modality sensor. After receiving multimodal inputs, the textual and temporal modalities are processed along separate flows. The time series is transformed within the time series(TS) flow (blue path) into TS tokens that can be embedded into a language model. These tokens are aligned with tokenized text according to their original positions. The mixed tokens are then fed into an LLM for reasoning.

At the output stage, the tokens are split into TS tokens and text tokens. The temporal component is reconstructed into a multivariate time series, which serves as a supervision signal to ensure faithful modeling of temporal dynamics. The textual component is decoded into natural language via a tokenizer, producing human-interpretable reasoning outputs for the target tasks (Wei et al., 2022; Wang et al., 2023b; OpenAI, 2023; Touvron et al., 2023a).

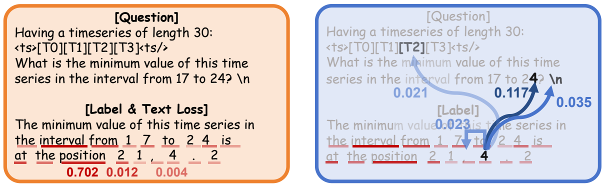
The goal of a Time Series Multimodal Large Language Model (TS-MLLM) is to jointly leverage multivariate time series and contextual text for question answering and reasoning. Formally, the multivariate time series is

$$\mathbf{X} = \{x_0, x_1, \ldots, x_{L-1}\} \in \mathbb{R}^{D \times L},$$

where D is the dimensionality and L the sequence length. The textual information is a natural language instruction $\mathbf{I}$, which specifies contextual knowledge and the task. We abstract the TS-MLLM as a mapping function ϕ, and the task can be formulated as $A = \phi(\mathbf{I}, \mathbf{X})$, where A is the predicted answer.

3.2 Time Series Embedding

Motivation. A central challenge in multimodal large models lies in how to transform and integrate heterogeneous modalities into a unified token space. For time-series signals, prior work Nie et al. (2023) widely adopts a patching strategy, which segments sequences into local chunks to reduce length and capture short-range dependencies. While efficient, this compression introduces two challenges: (i) absolute temporal positions are lost once consecutive points are merged, making it difficult for the model to recover precise ordering across the sequence; and (ii) patching only preserves local structures, whereas downstream reasoning often demands both local and global temporal relations. As illustrated in Figure 4b, neglecting absolute indices significantly degrades temporal reconstruction and reasoning. Hence, a robust temporal embedding scheme should jointly encode local patterns and absolute positions along the sequence.


[Question]
Having a timeseries of length 30:
<ts>[T0][T1][T2][T3]<ts/>
What is the minimum value of this time series in the interval from 17 to 24? \n
[Label & Text Loss]
The minimum value of this time series in the interval from 1 7 to 2 4 is at the position 2 1 , 4 . 2
0.702
0.012
0.004
[Question]
Having a timeseries of length 30:
<ts>[T0][T1][T2][T3]<ts/>
What is the minimum value of this time series in the interval from 17 to 24? \n
0.021
0.117
0.035
[Label]
0.023
The minimum value of this time series in the interval from 1 7 to 2 4 is at the position 2 1 , 4 . 2

(a) Loss distribution.

(b) Attention distribution.

Figure 3: Token-level loss and attention of label tokens, showing how temporal information is overshadowed by textual content during model optimization.

Solution. We design a temporal embedding pipeline to address the above challenges. First, to retain absolute positions, we explicitly inject the index of each time step as an additional dimension alongside its value. Second, to preserve local relative dependencies while ensuring computational efficiency, we follow the patching strategy of Xie et al. (2024); Nie et al. (2023) with non-overlapping patches. This fusion ensures that each token encodes not only local temporal context but also the absolute coordinate of the underlying segment, providing the model with richer information for downstream tasks.

Details. Let the input be $\mathbf{X} = \{x_t^{(m)} \mid t = 0, \ldots, L-1;\ m = 1, \ldots, D\} \in \mathbb{R}^{L \times D}$, where D is the number of channels and L is the sequence length. In the following, we describe how the raw time series is transformed through positional encoding and tokenization, and eventually fused with other modalities.

A) *Time Series Position Embedding.* We process each channel independently. For channel d, the sequence is $\mathbf{x}^{(d)} = [x_0^{(d)}, \ldots, x_{L-1}^{(d)}] \in \mathbb{R}^L$, and the absolute index vector is $\text{Index} = [0, 1, \ldots, L-1] \in \mathbb{R}^L$. We then augment values with indices by concatenation: $\tilde{\mathbf{X}}^{(d)} = \left[\text{Index}; \mathbf{x}^{(d)}\right] \in \mathbb{R}^{2 \times L}$. Stacking across channels yields $\tilde{\mathbf{X}} \in \mathbb{R}^{D \times 2 \times L}$, which retains both absolute positions and values.

B) *Patching.* Each augmented channel is split into non-overlapping patches of length P. For a sequence of length L, the number of patches is $N^{(L)} = \lceil L/P \rceil$, where zero-padding is applied to the last patch if needed. For channel d and patch index $n \in \{0, \ldots, N^{(L)} - 1\}$, the flattened segment is defined as $\hat{\mathbf{x}}_n^{(d)} = \text{reshape}(\tilde{\mathbf{X}}^{(d)}_{:,\, nP:(n+1)P-1}) \in \mathbb{R}^{2P}$. Collecting all patches yields $\hat{\mathbf{X}}^{(L)} \in \mathbb{R}^{D \times N^{(L)} \times 2P}$. When constructing a batch containing multiple sequences with different lengths $\{L_i\}$, the corresponding patch numbers $\{N^{(L_i)}\}$ may vary. To enable efficient batch training, we set $N = \max_i N^{(L_i)}$ and apply zero-padding along the patch dimension for all sequences with $N^{(L_i)} < N$. Thus, the final embedded representation for the whole batch is consistently aligned as $\hat{\mathbf{X}} \in \mathbb{R}^{B \times D \times N \times 2P}$, where B is the batch size.

C) *MLP Encoder.* To map each patch into the hidden dimension H, we apply a shared MLP $f_\phi : \mathbb{R}^{2P} \to \mathbb{R}^H$. Each patch token is $\mathbf{T}_{ts}^{(d,n)} = f_\phi(\hat{\mathbf{x}}_n^{(d)}) \in \mathbb{R}^H$, forming $\mathbf{T}_{ts} \in \mathbb{R}^{D \times N \times H}$. For transformer-style consumption, we flatten channel and patch axes to $\mathbf{T}_{ts}^{\text{flat}} \in \mathbb{R}^{(D \cdot N) \times H}$, producing flat tokens suitable for modality joint with text tokens.

D) *Modality Fusion.* For each extracted time-series segment $\hat{\mathbf{X}}$, the MLP encoder yields flat tokens

$$\mathbf{T}_{ts}^{\text{flat}} = \left[\mathbf{T}_{ts}^{(1)}, \mathbf{T}_{ts}^{(2)}, \ldots, \mathbf{T}_{ts}^{(M)}\right] \in \mathbb{R}^{(D \cdot N) \times H},$$

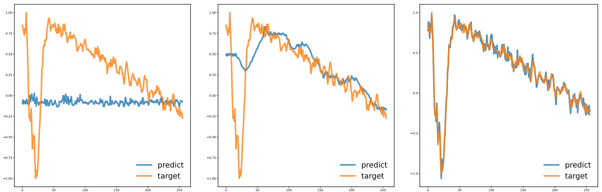
where each sub-sequence $\mathbf{T}_{ts}^{(m)} \in \mathbb{R}^{N_m \times H}$ corresponds to the m-th original series segment, and N_m matches the number of patches derived from that segment. Preserving temporal order, these sub-sequences are inserted into the textual sequence at designated positions, enclosed by special markers `<ts>` and `<ts/>`. This design ensures seamless alignment between temporal and textual tokens for joint modeling.

3.3 Time Series Sensing

Motivation. A common paradigm in multimodal reasoning is to encode non-text modalities (e.g., time series) and feed them into an LLM, while using text as the sole output channel. This design has a clear advantage: it naturally leverages the strong reasoning and generation capabilities of LLMs, enabling seamless interaction with humans in natural language. However, this approach also comes with a critical drawback. Since the training objective is dominated by text-based supervision, the model tends to gradually ignore the non-text modalities. As illustrated in Figure 3, text-related components in the label contribute significantly more to the loss than numerical tokens that reflect temporal information. Similarly, the attention visualization in Figure 3b shows that when generating time-series-related tokens, the model primarily attends to the textual instructions rather than temporal features. Consequently, the model learns only shallow text matching and fails to capture the structural patterns of time series. In short, relying solely on text labels prevents the model from forming a holistic sense of multi-modal time series.

Solution.
The key challenge is the lack of explicit supervision for the time-series modality. To address this, we require the model to decode its hidden time-series tokens into multi-dimensional sequences that reconstruct the original input, ensuring that temporal information is explicitly preserved during training.

Details. Similar to standard LLMs reports Bai et al. (2025); Touvron et al. (2023b), the model generates a hidden representation $\mathbf{h}_t \in \mathbb{R}^d$ for each token. We extend this process by introducing the following time-series-aware components:



(a) w/o FFT loss (b) w/o PosEmb (c) Full Conf

Figure 4: Temporal reconstruction results under different configurations.

A) Modality Decoupling. The hidden output $\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_T\}$ contains mixed information from both modalities. To avoid interference, we designate the first $D \cdot N$ tokens as time series tokens $\mathbf{H}^{(ts)} = \{\mathbf{h}_1, \ldots, \mathbf{h}_{D \cdot N}\}$, while the remaining tokens $\mathbf{H}^{(txt)} = \{\mathbf{h}_{D \cdot N+1}, \ldots, \mathbf{h}_T\}$ are used for conventional text modeling.

B) Time-Series Reconstruction. The latent representation is $\mathbf{H}^{(ts)} \in \mathbb{R}^{(D \cdot N) \times H}$, which can be viewed as $D \cdot N$ tokens of dimension H. Applying the MLP decoding $f_{\phi_{\text{r}}} : \mathbb{R}^H \to \mathbb{R}^P$ independently to each token yields $\hat{\mathbf{X}}_{\text{rec}}^{(d)} \in \mathbb{R}^{(D \cdot N) \times P}$. Re-arranging this tensor into channel token form gives $\hat{\mathbf{X}}_{\text{rec}}^{(d)} \in \mathbb{R}^{D \times N \times P}$, and finally the inverse patching merges (N, P) into the original sequence length L, producing $\mathbf{X}_{\text{rec}} \in \mathbb{R}^{D \times L}$. Formally, the above process can be expressed as follows:

$$(D \cdot N) \times H \xrightarrow{f_{\phi_r}} (D \cdot N) \times P \xrightarrow{\text{reshape}} D \times N \times P \xrightarrow{\text{inv-patch}} D \times L$$

C) Time-Series Loss. Unlike most time series foundation models trained solely on time-series targets (Figure 4a), using only MSE is insufficient when optimizing both time-series and textual objectives. Small point-wise variations contribute little to the MSE, especially compared to the textual cross-entropy loss, making it difficult for the time-series module to capture temporal patterns. To address this, we introduce a frequency-domain loss that captures high-frequency variations, following Wang et al. (2024a), which shows that frequency-domain targets can enhance model performance. To preserve both value fidelity and temporal dynamics, we combine time-domain and frequency-domain constraints.

$$\mathcal{L}_{ts} = \|\mathbf{X_{rec}} - \mathbf{X}\|_2^2 + \|\mathcal{F}(\mathbf{X_{rec}}) - \mathcal{F}(\mathbf{X})\|_2^2,$$

where $\mathcal{F}(\mathbf{X}) = \sum_{t=0}^{L-1} x_t e^{-j2\pi kt/L}$ denotes the discrete Fourier transform (DFT).

D) Loss Integration and Training. For the textual part, we only consider the hidden states corresponding to the text tokens, i.e., $\mathbf{H}^{(\text{txt})} = \{\mathbf{h}_{D\cdot N+1}, \ldots, \mathbf{h}_T\}$. Each hidden state $\mathbf{h}_i$ is projected into the vocabulary space by the language modeling head: $\mathbf{z}_i = lm\_head(\mathbf{h}_i) \quad i = D \cdot N + 1, \ldots, T$. where $\mathbf{z}_i \in \mathbb{R}^{|\mathcal{V}|}$ denotes the logits.

The predicted distribution is obtained via softmax: $\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{z}_i) \in [0, 1]^{|\mathcal{V}|}$. Finally, the textual cross-entropy loss is computed as $\mathcal{L}_{\text{txt}} = -\frac{1}{T-D\cdot N} \sum_{i=D\cdot N+1}^{T} \mathbf{y}_i^\top \log \hat{\mathbf{y}}_i$, where $\mathbf{y}_i \in \{0, 1\}^{|\mathcal{V}|}$ is the one-hot label for the i-th token.

The final training objective combines both constraints: $\mathcal{L} = \mathcal{L}_{\text{txt}} + \mathcal{L}_{\text{ts}}$, jointly updating the LLM backbone, the time-series embedding, and the reconstruction MLP during back propagation.

4 ChronGen

To train and evaluate multimodal time-series models on their ability to recognize and utilize various temporal features, we propose ChronGen, a data generator that incrementally incorporates local features into a base trend line and annotates these features using natural language. ChronGen produces comprehensive representations of time series, from which we construct question–answer pairs by selecting and combining local features, thereby enabling the training of models in temporal cognition and reasoning. To thoroughly assess this capability, we design three difficulty levels covering a total of eleven types of temporal reasoning tasks, as detailed in § 4.3.

4.1 Implementation

To systematically construct multimodal datasets that align temporal dynamics with textual annotations, we propose **ChronGen**, a *CHange-awaRe rules-OrieNtd time series GENerator*. As in Figure 5, ChronGen is designed to synthesize sequences that progressively incorporate diverse trends while maintaining explicit textual labels for each segment. This capability provides a controlled environment for studying multimodal reasoning, where both temporal evolution and natural language descriptions are essential. Given a target length L and the number of segments K, ChronGen first samples $K - 1$ change points along the timeline, which divide the sequence into contiguous segments. For the initial segment, a base trend (e.g., linear, constant, or oscillatory) is instantiated. Each subsequent segment is generated by conditioning on the previous trend and injecting additional variations to gradually enrich the dynamics. This process yields a piecewise-evolving time series that captures both continuity and variability. For every generated segment, ChronGen produces a corresponding textual annotation, which describes the dominant trend (e.g., "increasing steadily," "plateau rising," "decreasing sharply"). These annotations serve as aligned labels, ensuring that each temporal unit has a semantic description that can be leveraged in downstream multimodal learning. As a result, ChronGen provides a flexible and interpretable way to generate synthetic data that balances temporal complexity with linguistic clarity.

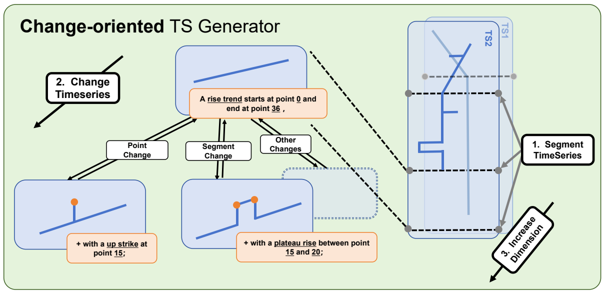

Change-oriented TS Generator
2. Change Timeseries
A rise trend starts at point 0 and end at point 36 ,
Point Change
Segment Change
Other Changes
+ with a up strike at point 15;
+ with a plateau rise between point 15 and 20;
TS2
TS1
1. Segment TimeSeries
3. Increase Dimension

Figure 5: **Change-oriented Time Series Generator** pipeline.

4.2 Model training

To endow models with the ability to reason about and analyze time series that exhibit rich and evolving dynamics, we construct large-scale training data using the proposed ChronGen framework. Specifically, ChronGen generates multivariate time series characterized by diverse change patterns and temporal complexities. Each sequence is paired with textual annotations that describe the segment-wise dynamics, thereby aligning temporal signals with natural-language semantics.

To further enhance the reasoning dimension, we design task-oriented question–answer templates, which transform the generated time series and annotations into training samples for multimodal reasoning. These templates encourage the model to connect raw temporal evolution with high-level textual inference, reflecting real-world analytical demands. The detailed configurations of the templates are provided in § A.1.

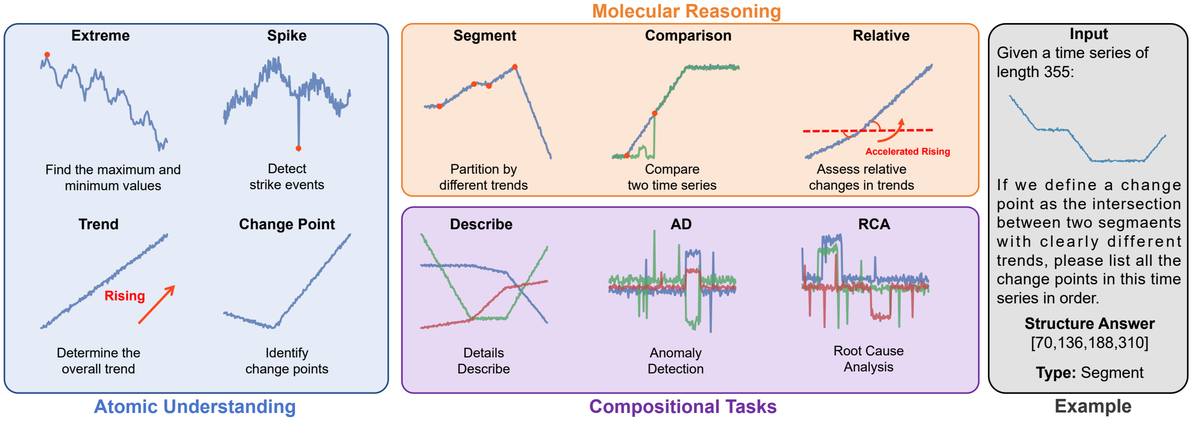


Figure 6: Overview of task categories in EvalTS.

Table 1: Model Performance on Fundamental and Specialized Tasks

Model	Atomic Understanding									Molecular Reasoning				Compositional Tasks			
	Change Point		Extreme		Spike		Trend		Avg.	Segment	Comparison	Relative	Avg.	Describe	RCA	AD	Avg.
	Uni	Multi	Uni	Multi	Uni	Multi	Uni	Multi									
ChatTime	–	–	–	–	–	–	0.85	0.31	–	–	–	–	–	–	–	–	–
Time-MQA	0.06	0.19	0.56	0.27	0.21	0.24	0.41	0.37	0.21	0.13	0.14	0.02	0.09	0.05	0.01	0.02	0.02
Qwen2.5	0.18	0.05	0.86	0.56	0.30	0.37	0.95	0.68	0.42	0.15	0.13	0.58	0.29	0.05	0.01	0.00	0.02
ChatTS	0.03	0.06	0.30	0.013	0.79	0.56	**0.99**	**0.77**	0.33	0.00	0.32	0.54	0.28	**0.14**	**0.36**	0.46	**0.32**
GPT-5	**0.32**	**0.35**	**0.98**	**0.89**	**0.90**	**0.84**	0.95	0.71	**0.70**	**0.34**	**0.50**	**0.66**	0.50	0.13	0.23	0.46	0.27
TimeSenses-7B	0.05	0.15	0.93	0.44	0.23	0.24	0.98	0.94	0.43	0.19	0.33	0.53	0.35	0.10	0.33	**0.49**	0.31
TimeSense-14B	**0.98**	**0.97**	**0.99**	**0.97**	**0.95**	**0.87**	**0.99**	**0.98**	**0.79**	**0.88**	**0.84**	**0.84**	**0.85**	**0.39**	**0.49**	**0.82**	**0.57**

Through this process, our dataset integrates synthetic yet interpretable temporal trajectories with semantically grounded textual descriptions, offering a controlled but expressive environment for training multimodal time-series reasoning models.

During model training, we first adopted the time-series data generation strategy proposed in Xie et al. (2024) to construct an initial alignment dataset, comprising 100K question–answer pairs sampled from a mixture of the original datasets. This step was designed to endow the model with fundamental alignment capabilities. Building upon this foundation, we further performed supervised fine-tuning (SFT) on another 100K training instances generated using the same method, thereby enabling deeper enhancement of the model's reasoning and task-solving abilities. To further strengthen the model's instruction-following capacity, we additionally incorporated the Tulu instruction dataset from Lambert et al. (2024). The full training configurations are detailed in § A.3.

4.3 EvalTS

As shown in Figure 6, the EvalTS benchmark decomposes multimodal time-series evaluation into three categories: Atomic Understanding, Molecular Reasoning, and Compositional Tasks. Illustrative examples are provided in § A.4.

Atomic Understanding. This category targets the most fundamental units of time-series cognition: individual points and global trends. Tasks include extremum identification, overall trend recognition, change-point detection, strike detection, and value retrieval at a specific index. Tasks are divided into single-series and multi-series settings, and sequence length is unconstrained, reflecting real-world scenarios with variable-duration series.

Molecular Reasoning. Building on atomic cognitions, this category requires establishing relationships between multiple fundamental units for dynamic temporal understanding. Tasks include segmenting a series into local trends, comparing trends across two series, and assessing relative changes between consecutive trends. These tasks demand integration of trend and value information to evaluate temporal patterns and relative dynamics.

Compositional Tasks. This category assesses the application of atomic and molecular reasoning in complex scenarios. Tasks include analyzing multivariate or complex series, detecting anomalies based on user-defined rules, and ranking anomalies while identifying root causes. These tasks require combining multiple basic abilities and are designed to closely reflect real-world temporal reasoning challenges.

5 Experiments

5.1 Experiment Setup

Evaluation Datasets. We evaluate all models on the EvalTS introduced in § 4.3, where task definitions and example queries are described in § A.4. To expand the size of the evaluation set, we choose an open-source dataset used by ChatTime Wang et al. (2025). We selected four types of multiple-choice questions that reflect temporal reasoning capabilities and categorized them into cross-domain and out-of-domain tasks. Specifically, two datasets focusing on temporal trends and outliers were used as cross-domain evaluation sets, denoted as MCQA D1 and MCQA D2. Datasets containing seasonality and volatility features, which are not present in our training data, were used as out-of-domain evaluation sets and displayed as MCQA D3 and MCQA D4.

Evaluation Models. In particular, we train TimeSense-14B and compare against GPT-5, ChatTS (Xie et al., 2024), and Qwen2.5 (Bai et al., 2025). Specifically, Qwen2.5 refers to Qwen2.5-14B-Instruct, while ChatTS is based on this model as its backbone. For parameter-scale comparison, we also train and evaluate TimeSense-7B alongside Time-MQA (Kong et al., 2025) and ChatTime (Wang et al., 2025), where both ChatTime and Time-MQA are 7B-scale models. Since ChatTime only supports multiple-choice outputs, we adapt the evaluation format for the *Trend* task into a multiple-choice style to make the evaluation feasible.

5.2 Experiment Results

The EvalTS results are summarized in Table 5. Across nearly all categories, TimeSense-14B achieves the best performance, substantially outperforming other baselines. This demonstrates the effectiveness of explicitly incorporating temporal reasoning modules into multimodal LLMs. We make the following observations:

Comparison to LLM. Mainstream text-based models such as GPT-5 show strong results on fundamental tasks (e.g., Extreme, Index, and Spike), likely due to their superior language understanding and pretraining scale. However, their advantage diminishes in compound and complex temporal reasoning tasks, where temporal dynamics are more critical. Compared with text-only Qwen-14B, both ChatTS-14B and TimeSense-14B consistently perform better, especially on complex tasks (e.g., RCA, AD). Interestingly, on several datasets, they even surpass GPT-5, highlighting that multimodal temporal modules better capture sequential dependencies for reasoning.

Comparison to TS-MLLM. When scaling down to 7B parameters, TimeSense-7B still surpasses Time-MQA-7B and ChatTime-7B. This suggests that the design of TimeSense provides stronger generalization, though smaller models naturally show degraded performance compared to their 14B counterparts.

Table 2: Model performance across cross-domain tasks. (Red bold = best, Blue bold = second)

Model	**MCQ D1**				**MCQ D2**			
	64	128	256	512	64	128	256	512
ChatTime	**0.90**	**0.90**	0.88	0.82	0.79	0.70	0.62	0.57
Time-MQA	0.41	0.42	0.32	0.41	0.33	0.50	0.27	0.35
Qwen2.5	0.25	0.36	0.21	0.48	0.32	0.36	0.32	0.28
ChatTS	0.54	0.88	**0.91**	**0.92**	0.67	**0.96**	**0.99**	**0.99**
GPT5	0.81	0.54	0.46	0.67	**0.84**	0.65	0.79	0.68
TimeSense-7B	0.86	0.85	0.85	0.86	0.72	0.78	0.72	0.73
TimeSense-14B	**0.94**	**0.92**	**0.95**	**0.93**	**0.99**	**0.98**	**0.99**	**0.99**

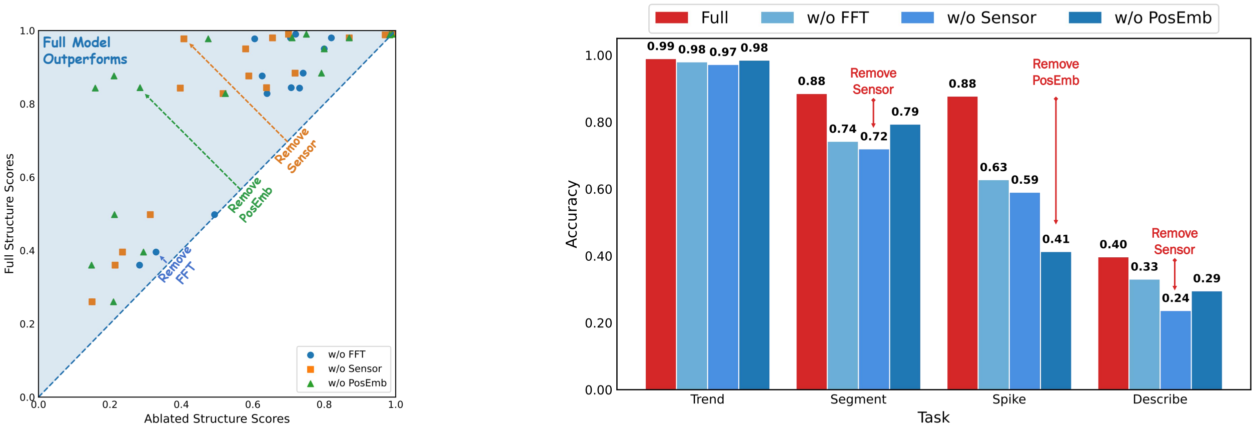
Table 3: Model performance across out-of-domain tasks. (Red bold = best, Blue bold = second)

Model	**MCQ D3**				**MCQ D4**			
	64	128	256	512	64	128	256	512
Time-MQA	**0.73**	0.52	0.40	0.42	0.45	0.35	0.18	0.35
ChatTime	**0.66**	**0.65**	**0.65**	**0.62**	**0.88**	**0.90**	**0.86**	**0.75**
Qwen2.5	0.54	0.62	0.52	0.44	0.34	0.50	0.46	0.47
ChatTS	0.33	0.38	0.36	0.25	0.39	0.61	0.61	0.61
GPT5	0.63	0.50	0.63	0.54	0.42	0.50	0.32	0.43
TimeSense-7B	0.18	0.27	0.25	0.27	0.25	0.32	0.26	0.17
TimeSense-14B	0.62	**0.65**	**0.67**	**0.65**	**0.72**	**0.68**	**0.67**	**0.69**

Overall, these results confirm that TimeSense not only scales effectively but also consistently outperforms both text-only and multimodal baselines across different task categories, especially in complex temporal reasoning.

5.3 Generalization of TimeSense

As exhibited in Table 2 and Table 3, this evaluation consists of two parts, namely Cross-Domain tasks and Out-of-Domain tasks, depending on their relevance to TimeSense's training data. To better assess TimeSense's generalization ability, we intentionally did not use the released training corpus from Wang et al. (2025). Under this setting, TimeSense-14B achieves consistent superiority on Cross-Domain tasks and remains competitive with ChatTime on Out-of-Domain tasks, despite the latter being trained with domain data. When scaling down to TimeSense-7B, the model performs on par with ChatTime in Cross-Domain tasks and even shows advantages on longer series. However,



(a) Comparison across all tasks

(b) Comparison across representative tasks

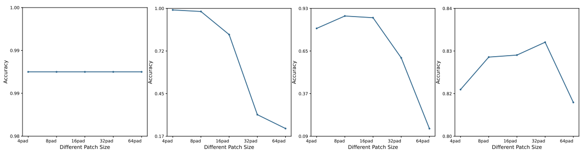
Figure 7: Ablation studies of different model structures

it exhibits a notable performance drop in Out-of-Domain scenarios, highlighting the dependency of generalization ability on model size.

5.4 Ablation Studies

Effectiveness of Different Model Structures. To evaluate the contributions of time-series position embedding (PosEmb), sensor(Sensor), and FFT loss(FFT) in training, we conduct ablation studies by removing each component. The results are shown in Figure 7a, from which we can see that the performance drops when removing each module. Detailed results are provided in Section A.5, and four representative cases are shown in Figure 7b. For the basic Trend task, all variants remain strong, showing robustness to minor changes. However, Segment exhibits sharp declines, especially when removing Sensor and FFT loss (up to 15%), highlighting the Sensor's role in capturing global temporal shifts and the FFT loss in refining periodic sensitivity. For Spike task, ablating Time-Series PosEmb causes the largest drop, confirming its importance for precise temporal localization. Finally, Describe shows broader degradation under Sensor ablation, reflecting its reliance on multimodal integration. Overall, TS PosEmb ensures temporal precision, Sensor provides global awareness, and FFT loss enhances periodic pattern modeling, together driving superior temporal reasoning.

Effectiveness of Different Patch Sizes. Enlarging the patch size strengthens the model's ability to encode compressed temporal patterns, but reduces point-wise precision. This trade-off varies across tasks. For uni-variable trend recognition (Figure 8a), accuracy remains stable since coarse temporal representations are sufficient. In contrast, uni-variable change-point detection (Figure 8b) is sensitive to local fluctuations, so larger patches degrade performance. Temporal segmentation (Figure 8c) shows a non-monotonic trend: moderate patches improve context and accuracy, but overly large ones blur segment boundaries. Anomaly detection (Figure 8d) follows a similar pattern, peaking at patch size 32, with further enlargement impairing fine-grained anomaly detection. Detailed results are given in § A.5. Overall, setting the patch size to 8 provides a balanced and near-optimal performance across tasks.



(a) Trend (b) CP (c) Segment (d) AD

Figure 8: Ablation study results for patch size across different tasks.

6 Conclusion

In this work, we presented TimeSense, a multimodal time series model that integrates a Temporal Sense module to balance textual and temporal information, addressing the bias toward language in prior approaches. To characterize complex temporal reasoning, we introduced ChronGen, a controllable generator for multidimensional time series with trend variations, anomalies, and cross-channel interactions, and built the EvalTS benchmark spanning ten tasks across three difficulty levels. Experiments demonstrate that TimeSense achieves state-of-the-art performance, particularly excelling in complex reasoning scenarios where temporal dynamics and textual understanding must be jointly considered.

7 ETHICS STATEMENT

This paper does not involve human participants, confidential or sensitive data, or any activities that could pose ethical concerns. We affirm adherence to the ICLR Code of Ethics.

8 REPRODUCIBILITY STATEMENT

For reproducibility purpose, we release our core code at: https://anonymous.4open.science/r/timesense-984F

REFERENCES

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

Winnie Chow, Lauren Gardiner, Haraldur Thor Hallgrímsson, Maxwell A. Xu, and Shirley You Ren. Chatts: Aligning time series with llms via synthetic data and attribute descriptions. *arXiv:2412.03104*, 2024. URL `https://arxiv.org/abs/2412.03104`.

Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

Chang Gong, Chuzhe Zhang, Di Yao, Jingping Bi, Wenbin Li, and Yongjun Xu. Causal discovery from temporal data: An overview and new perspectives. *ACM Computing Surveys*, 57(4):1–38, 2024.

Yangdong He and Jiabao Zhao. Temporal convolutional networks for anomaly detection in time series. In *Journal of Physics: Conference Series*, volume 1213, pp. 042050. IOP Publishing, 2019.

Elizabeth E Holmes, Eric J Ward, and Kellie Wills. Marss: Multivariate autoregressive state-space models for analyzing time-series data. 2012.

Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. Bliva: A simple multimodal llm for better handling of text-rich visual questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2256–2264, 2024.

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

Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin Chen, Chenyi Lei, An Liu, Chengru Song, et al. Unified language-vision pretraining in llm with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023b.

Prajakta S Kalekar et al. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi school of information Technology*, 4329008(13):1–13, 2004.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

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

Yeqi Liu, Chuanyang Gong, Ling Yang, and Yingyi Chen. Dstp-rnn: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with Applications*, 143:113082, 2020.

Amy McGovern, Derek H Rosendahl, Rodger A Brown, and Kelvin K Droegemeier. Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction. *Data Mining and Knowledge Discovery*, 22(1):232–258, 2011.

Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29, 2023.

Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pp. 2130–2134. IEEE, 2015.

Y. Nie, N. Nguyen, et al. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Francisco Javier Nogales, Javier Contreras, Antonio J Conejo, and Rosario Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17(2):342–348, 2002.

OpenAI. Gpt-4 technical report. Technical report, OpenAI, 2023. URL https://arxiv.org/abs/2303.08774.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pp. 1394–1401. Ieee, 2018.

Seemant Tiwari. Segmentation and clustering of time series data. In *2023 International Conference for Advancement in Technology (ICONAT)*, pp. 1–6. IEEE, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Aneesh Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Hao Wang, Licheng Pan, Zhichao Chen, Degui Yang, Sen Zhang, Yifei Yang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. Fredf: Learning to forecast in the frequency domain. *arXiv preprint arXiv:2402.02399*, 2024a.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36:61501–61513, 2023a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pp. 396–416. Springer, 2024b.

Z. Wang et al. Chattime: A unified multimodal time series foundation model. *AAAI*, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/33384/35539.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

Z. Xu et al. Enginemt-qa: A large-scale temporal-textual qa dataset. OpenReview preprint, 2025a. URL https://openreview.net/forum?id=GByP03IitA.

Z. Xu et al. Itformer: Bridging time series and natural language for multi-task temporal-textual qa. *arXiv:2506.20093*, 2025b. URL `https://arxiv.org/abs/2506.20093`.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 947–956, 2009.

Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm–explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.

Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. Clustering time series using unsupervised-shapelets. In *2012 IEEE 12th international conference on data mining*, pp. 785–794. IEEE, 2012.

G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

Ruofei Zhang, Zhongfei Zhang, Mingjing Li, Wei-Ying Ma, and Hong-Jiang Zhang. A probabilistic semantic model for image annotation and multimodal image retrieval. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pp. 846–851. IEEE, 2005.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 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)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL `http://arxiv.org/abs/2403.13372`.

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

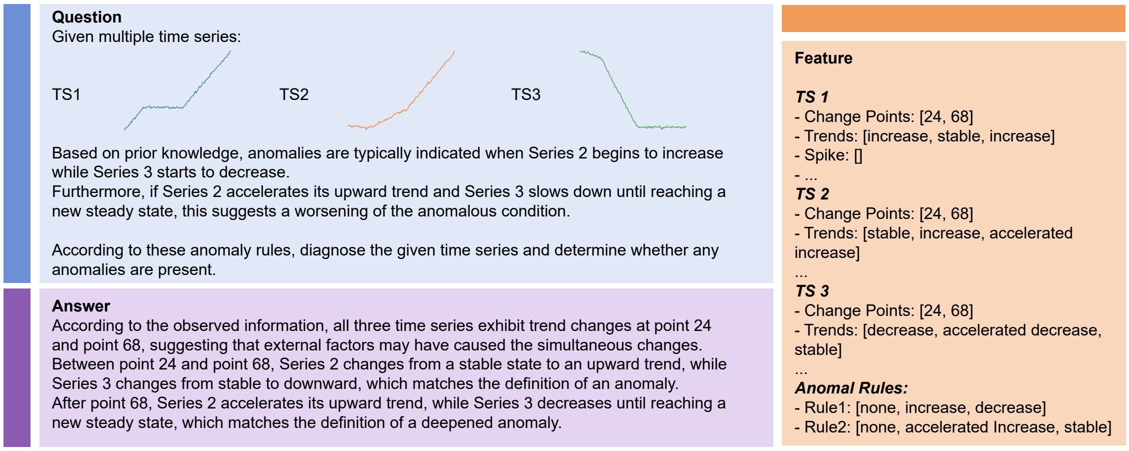
Z. Zhu et al. Large language models for time series: A survey. *arXiv:2402.01801*, 2024. URL `https://arxiv.org/abs/2402.01801`.

A APPENDIX

You may include other additional sections here.

A.1 TRAINING QA TEMPLATE

In this section, we provide illustrative examples of the data generation procedure described in § 4.2. The Question and Answer components are used for SFT training, where the temporal information is extracted into a dedicated time-series modality that serves both as input and as the reconstruction target. The Feature component highlights the internal design of ChronGen, including the step-by-step generation of change points, the trend patterns associated with the segments defined by these change points, and optional spike values that capture abrupt local variations.



Question
Given multiple time series:

Based on prior knowledge, anomalies are typically indicated when Series 2 begins to increase while Series 3 starts to decrease.
Furthermore, if Series 2 accelerates its upward trend and Series 3 slows down until reaching a new steady state, this suggests a worsening of the anomalous condition.

According to these anomaly rules, diagnose the given time series and determine whether any anomalies are present.

Answer
According to the observed information, all three time series exhibit trend changes at point 24 and point 68, suggesting that external factors may have caused the simultaneous changes. Between point 24 and point 68, Series 2 changes from a stable state to an upward trend, while Series 3 changes from stable to downward, which matches the definition of an anomaly.
After point 68, Series 2 accelerates its upward trend, while Series 3 decreases until reaching a new steady state, which matches the definition of a deepened anomaly.

Feature

TS 1
- Change Points: [24, 68]
- Trends: [increase, stable, increase]
- Spike: []
- ...

TS 2
- Change Points: [24, 68]
- Trends: [stable, increase, accelerated increase]

...

TS 3
- Change Points: [24, 68]
- Trends: [decrease, accelerated decrease, stable]

...

Anomal Rules:
- Rule1: [none, increase, decrease]
- Rule2: [none, accelerated Increase, stable]

Figure 9: Examples of training datasets

A.2 CHRONGEN CODE

The generator produces a piecewise time series by first splitting the timeline into K segments. Each segment starts with a base trend T_i, and richer dynamics are introduced by iteratively applying change functions (e.g., `point_change`, `segment_change`). Each change is immediately recorded in a feature array $\mathbf{F}$ and simultaneously converted into a textual annotation via a text template function. This ensures that every incremental modification to the time series has a corresponding annotation, providing fine-grained supervision for both numerical trends and textual descriptions.

A.3 TRAINING DETAILS

In the initial stage of time-series–text alignment, we integrated the Stage 1 and Stage 2 training procedure of Xie et al. (2024). Specifically, their released datasets include four instruction-tuning corpora: ChatTS-IFT for instruction following, ChatTS-SFT for question answering, and two alignment-focused datasets, ChatTS-Align_256 and ChatTS-Align_random. Different from Xie et al. (2024), we constructed a mixed dataset by combining ChatTS-Align_random, ChatTS-SFT, and ChatTS-IFT with a ratio of 5:3:2, and randomly sampled 100K pairs as the training corpus. This enabled us to accomplish model alignment training within a single stage.

In the second stage, we adopted the data generators introduced in § 4.2 to produce 100K time-series–text pairs. These raw pairs were then converted into task-specific question–answer formats through a set of designed QA rules, resulting in the TS-Enhance dataset. To further equip the model with capabilities for domain-specific tasks, we additionally incorporated a portion of time-series reasoning data derived from proprietary corpora, enhancing the model's ability to handle practical, real-world scenarios.

Both stages employed identical training hyperparameters. We performed full-parameter supervised fine-tuning (SFT) using DeepSpeed and LlamaFactory (Zheng et al., 2024) on a single machine with 8×A800 GPUs. The detailed hyperparameters are summarized in Table 4.

Algorithm 1 Change-oriented Time Series Generator with Incremental Annotation

```
Require: time length $L$, number of segments $K$
Ensure: time series $\mathbf{X}$ with text annotations $\mathbf{Y}$ and feature array $\mathbf{F}$
Initialize empty sequence $\mathbf{X} \leftarrow []$, feature array $\mathbf{F} \leftarrow []$, annotation set $\mathbf{Y} \leftarrow []$
Randomly sample $K - 1$ change points within $[1, L]$
Split timeline into segments $\{S_1, S_2, \ldots, S_K\}$
for $i = 1$ to $K$ do
    Generate base trend $T_i$
    Initialize segment sequence $S_i \leftarrow []$
    Initialize segment annotation $Y_i \leftarrow []$
    Iteratively construct variation $\Delta T_i$:
    while more changes to apply do
        Sample a change operation change $\in$ {point_change, segment_change, ...}
        Update segment: $S_i \leftarrow S_i + \text{change}(S_i, T_{i-1})$
        Record feature: $\mathbf{F} \leftarrow \mathbf{F} \cup \{\text{change}\}$
        Generate textual annotation: $y \leftarrow$ text_template(change)
        Append $y$ to $Y_i$
    end while
    Append segment $S_i$ to $\mathbf{X}$
    Append segment annotations $Y_i$ to $\mathbf{Y}$
end for
return $\mathbf{X}$, $\mathbf{Y}$, and feature array $\mathbf{F}$
```

Table 4: Training hyperparameters used in both Stage 1 and Stage 2.

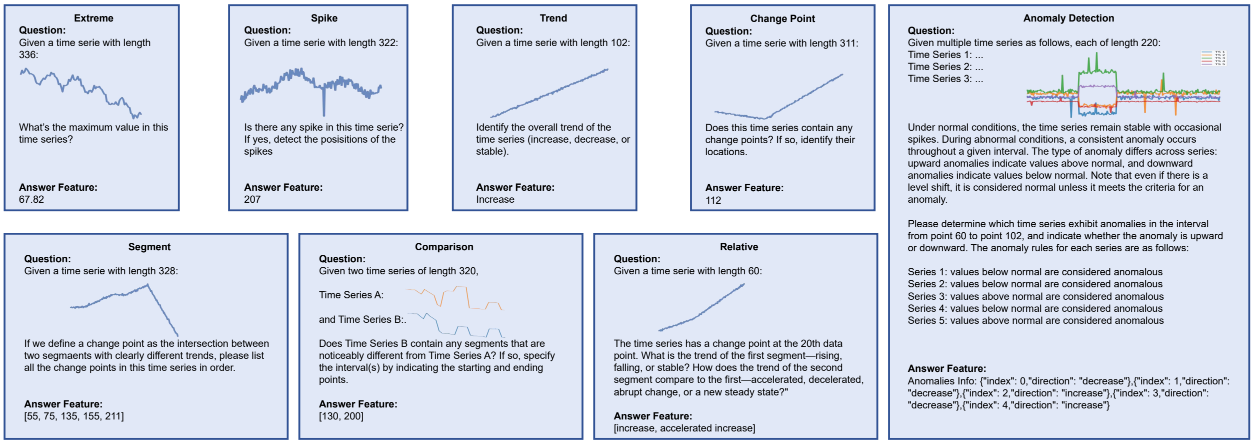
Hyperparameter	**Value**
Per-device train batch size	1
Gradient accumulation steps	32
Maximum training steps	1200
Learning rate	1e-5
Warmup ratio	0.02
GPUs	8 × A800

A.4 EVALTS EXAMPLES

In this section, we present examples for the majority of the tasks. The question represents the multimodal input directly fed into the model, while the answer feature serves as the reference for evaluation. We first extract the corresponding predicted feature from the model's output using a large language model (LLM), and then perform a one-to-one comparison between the predicted and reference features to compute the evaluation accuracy.

A.5 ABLATION STUDY DETAILS

In Table 5 and Table 6, we present the complete results of the ablation studies on both the model architecture and patch size. Overall, the findings demonstrate that each component contributes distinctly to the model's capabilities: time series PosEmb modules enhance temporal precision, time series sensor provide global contextual awareness, and certain loss functions improve pattern modeling. The effects of patch size are generally task-dependent, reflecting a trade-off between capturing broader temporal context and maintaining fine-grained resolution. Across the board, the results highlight how architectural design choices and temporal representation granularity jointly shape the model's overall performance and robustness.



Extreme

Question:
Given a time serie with length 336:

What's the maximum value in this time series?

Answer Feature:
67.82

Spike

Question:
Given a time serie with length 322:

Is there any spike in this time serie? If yes, detect the posisitions of the spikes

Answer Feature:
207

Trend

Question:
Given a time serie with length 102:

Identify the overall trend of the time series (increase, decrease, or stable).

Answer Feature:
Increase

Change Point

Question:
Given a time serie with length 311:

Does this time series contain any change points? If so, identify their locations.

Answer Feature:
112

Anomaly Detection

Question:
Given multiple time series as follows, each of length 220:
Time Series 1: ...
Time Series 2: ...
Time Series 3: ...

Under normal conditions, the time series remain stable with occasional spikes. During abnormal conditions, a consistent anomaly occurs throughout a given interval. The type of anomaly differs across series: upward anomalies indicate values above normal, and downward anomalies indicate values below normal. Note that even if there is a level shift, it is considered normal unless it meets the criteria for an anomaly.

Please determine which time series exhibit anomalies in the interval from point 60 to point 102, and indicate whether the anomaly is upward or downward. The anomaly rules for each series are as follows:

Series 1: values below normal are considered anomalous
Series 2: values below normal are considered anomalous
Series 3: values above normal are considered anomalous
Series 4: values below normal are considered anomalous
Series 5: values above normal are considered anomalous

Answer Feature:
Anomalies Info: {"index": 0,"direction": "decrease"},{"index": 1,"direction": "decrease"},{"index": 2,"direction": "increase"},{"index": 3,"direction": "decrease"},{"index": 4,"direction": "increase"}

Segment

Question:
Given a time serie with length 328:

If we define a change point as the intersection between two segmaents with clearly different trends, please list all the change points in this time series in order.

Answer Feature:
[55, 75, 135, 155, 211]

Comparison

Question:
Given two time series of length 320,

Time Series A:

and Time Series B:.

Does Time Series B contain any segments that are noticeably different from Time Series A? If so, specify the interval(s) by indicating the starting and ending points.

Answer Feature:
[130, 200]

Relative

Question:
Given a time serie with length 60:

The time series has a change point at the 20th data point. What is the trend of the first segment—rising, falling, or stable? How does the trend of the second segment compare to the first—accelerated, decelerated, abrupt change, or a new steady state?"

Answer Feature:
[increase, accelerated increase]

Figure 10: EvalTS examples

Table 5: Model Performance on different patch size

Model	Fundamental Tasks									Compound Tasks				Complex Tasks			
	Change Point		Extreme		Spike		Trend		Avg.	Segment	Comparison	Relative	Avg.	Describe	RCA	AD	Avg.
	Uni	Multi	Uni	Multi	Uni	Multi	Uni	Multi									
4padded	0.99	0.99	0.99	0.98	0.97	0.89	0.99	0.98	0.90	0.80	0.85	0.83	0.83	0.11	0.44	0.82	0.46
8padded	0.98	0.98	0.99	0.97	0.99	0.99	0.94	0.88	0.84	0.84	0.85	0.40	0.50	0.83	0.58		
16padded	0.83	0.77	0.99	0.96	0.87	0.84	0.99	0.99	0.88	0.87	0.77	0.89	0.84	0.25	0.50	0.83	0.52
32padded	0.31	0.22	0.99	0.96	0.87	0.78	0.99	0.97	0.74	0.61	0.42	0.37	0.46	0.04	0.49	0.83	0.45
64padded	0.22	0.19	0.99	0.90	0.72	0.29	0.99	0.97	0.65	0.14	0.36	0.19	0.23	0.05	0.48	0.81	0.45

Table 6: Ablation Study on Benchmark Tasks

Model	Fundamental Tasks									Compound Tasks				Complex Tasks			
	Change Point		Extreme		Spike		Trend		Avg.	Segment	Comparison	Relative	Avg.	Describe	RCA	AD	Avg.
	Uni	Multi	Uni	Multi	Uni	Multi	Uni	Multi									
TimeSense-Full	0.98	0.98	0.99	0.97	0.95	0.88	0.99	0.99	0.84	0.88	0.84	0.84	0.85	0.40	0.50	0.83	0.58
TimeSense-w/o-fft_loss	0.52	0.61	0.98	0.97	0.80	0.63	0.99	0.98	0.59	0.44	0.31	0.73	0.49	0.23	0.49	0.64	0.45
TimeSense-w/o-pos	0.57	0.28	0.99	0.96	0.80	0.43	0.99	0.91	0.50	0.17	0.28	0.16	0.20	0.39	0.21	0.52	0.37
TimeSense-w/o-sensor	0.37	0.28	0.97	0.97	0.26	0.59	0.99	0.97	0.43	0.24	0.19	0.40	0.31	0.14	0.31	0.52	0.33