

TimeSense: Decoupling Modality Learning for Robust Time Series Reasoning

Zhirui Zhang^{1,2}, Changhua Pei³, Tianyi Gao⁴, Zhe Xie¹, Yibo Hao¹, Zhaoyang Yu¹, Longlong Xu¹, Tong Xiao¹, Jing Han⁴, Gaogang Xie³, Dan Pei^{1,2}

¹Department of Computer Science and Technology, Tsinghua University,

²Beijing National Research Center for Information Science and Technology, Tsinghua University,

³Computer Network Information Center, Chinese Academy of Sciences,

⁴ZTE Corporation

Correspondence: chpei@cnic.cn

Abstract

In the time series domain, increasing efforts extend large language models (LLMs) to multimodal large language models (MLLMs) by incorporating temporal data for downstream temporal understanding tasks. In conventional MLLM training, models primarily rely on language signals, while temporal information provides substantially weaker supervision. This imbalance leads to insufficient learning of temporal patterns. To address this issue, we propose TimeSense, a framework with a disentangled reconstruction structure that introduces explicit temporal supervision and encourages the learning of both global trends and local temporal variations that are often underrepresented in text-dominated multimodal training. Existing QA benchmarks provide limited coverage of temporal variations and compositional temporal dependencies, making it difficult to systematically diagnose these MLLMs capabilities across diverse temporal understanding tasks. To facilitate a more fine-grained evaluation, we introduce EvalTS, a comprehensive benchmark spanning three difficulty levels (Atomic, Molecular, and Compositional) with diverse high-quality queries for evaluating temporal understanding abilities in large language and multimodal language models. Experiments on EvalTS and existing benchmarks demonstrate that TimeSense consistently outperforms prior methods and substantially improves temporal understanding performance.

direction for enabling more expressive, explainable, and human-aligned temporal understanding. Inspired by the success of vision-language models, time series multimodal systems aim to extend such capabilities by combining temporal inputs with language, offering zero-shot reasoning over complex sequential patterns (Xie et al., 2025; Wang et al., 2025a,b; Kong et al., 2025).

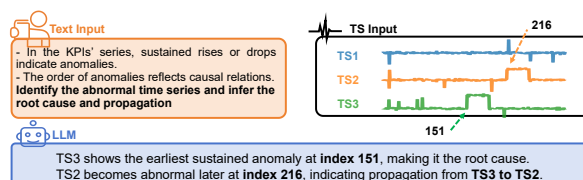


Figure 1: Example of time series reasoning task with multimodal large language model (MLLM).

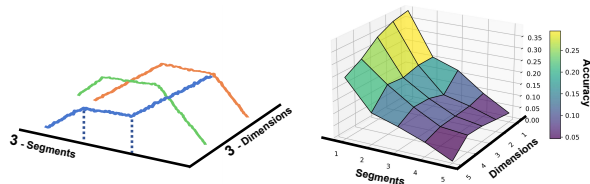


Figure 2: **Left:** An illustration of time series reasoning on the different segment and dimensions. **Right:** Performance comparison of baseline model under varying numbers of time series segments and dimensions.

1 Introduction

Time series data are foundational to numerous real-world domains, including energy, healthcare, transportation, weather, and finance (Nogales et al., 2002; Morid et al., 2023; Lippi et al., 2013; McGovern et al., 2011; Yu et al., 2023). With the recent progress of large language models (LLMs) and multimodal learning, integrating natural language with time series signals presents a promising

Current models often treat language merely as a wrapper or annotation for temporal tasks, rather than a means for interaction or compositional reasoning (Jin et al., 2024; Yang et al., 2025; Su et al., 2024; Ye et al., 2024; Zhang et al., 2024b). In contrast, natural language can provide rich, contextualized supervision for temporal events, causal chains, and anomaly descriptions. Unlocking this potential calls for time series LLMs that not only solve numerical tasks, but also understand, explain, and reason about time series data in natural language. Drawing parallels to the revolution in Vision-Language Models (VLMs) (Radford et al.,

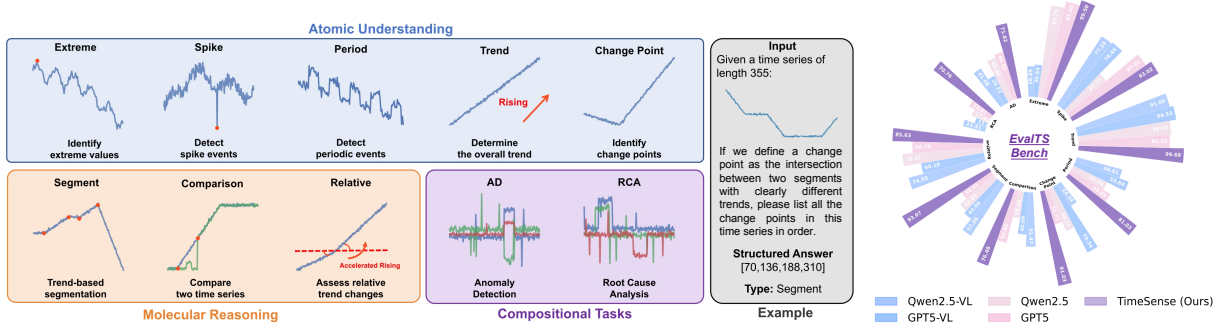


Figure 3: Examples from the EvalTS benchmark and performance comparison between our model and other widely-used LLMs/MLLMs.

2021; Zhang et al., 2024a; Liu et al., 2023; Bordes et al., 2024), a pivotal question arises: *Can we replicate this multimodal synergy in the temporal domain, empowering LLMs to perceive and reason over time series data?* However, the realization of such text-time series multimodal models currently faces the following challenges:

First, existing multimodal time series benchmarks provide limited evaluation of complex temporal reasoning scenarios. As the number of temporal segments and dimensions increases, existing multimodal time series models exhibit substantial performance degradation (Figure 2), suggesting limited capability in complex temporal reasoning tasks. However, current evaluations are typically restricted to specific downstream tasks or simplified multiple-choice settings, preventing controlled and fine-grained analysis of model behavior across varying levels of temporal complexity, ranging from atomic pattern recognition to compositional analysis. This limitation highlights the need for a comprehensive diagnostic benchmark for multimodal time series reasoning.

Second, existing time series MLLMs suffer from **text-dominated optimization**. Most models are trained using text-generation objectives in which supervision is applied primarily in the language space. As a result, optimization is disproportionately driven by textual tokens, while temporal representations receive comparatively weak supervision. This imbalance causes models to focus on sparse local patterns while underrepresenting intermediate temporal transitions and long-range dependencies.

In this work, we introduce **EvalTS**, shown in Figure 3, a comprehensive benchmark for multimodal time series reasoning. EvalTS systematically evaluates reasoning capabilities across multiple levels of temporal complexity, including *Atomic Understanding*, *Molecular Reasoning*, and *Compo-*

sitional Tasks.

We further propose **TimeSense**, a fusion-then-decoupling framework that models time series as a dedicated modality within the LLM architecture. By decoupling temporal hidden states after multimodal fusion and applying explicit reconstruction-based supervision, TimeSense mitigates text-dominated optimization and improves the model’s ability to analyze and reason over temporal patterns.

Comprehensive experiments demonstrate that TimeSense consistently outperforms prior methods on both EvalTS and four external benchmarks, validating its robustness and generalizability. Compared to leading baselines like GPT5, our approach demonstrates superior robustness, delivering a 121% improvement in Trend-based segmentation on GPT5 (Figure 3). We release our code and related EvalTS resources¹ to facilitate further research on multimodal time series analysis.

Our contributions are summarized as follows:

- We introduce EvalTS, a comprehensive benchmark for multimodal time series reasoning with controlled evaluation across multiple levels of temporal complexity.
- We propose TimeSense, a fusion-then-decoupling framework that preserves cross-modal interaction while explicitly supervising decoupled temporal representations for improved temporal modeling.
- We demonstrate that TimeSense achieves strong performance and robustness across EvalTS and multiple external benchmarks, substantially outperforming existing multimodal baselines.

¹<https://github.com/NetManAIOps/TimeSense>

2 Related Works

2.1 Classical Time Series Tasks and Methods

Research on time series has long focused on tasks such as 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 relied on statistical models, including ARIMA, exponential smoothing, and state-space models (Zhang, 2003), while recent approaches increasingly adopt deep neural architectures such as RNNs, LSTMs, and temporal convolutional networks to model complex temporal dependencies and non-linear dynamics (Liu et al., 2020; Siami-Namini et al., 2018; He and Zhao, 2019). Despite methodological differences, most existing approaches remain task-specific and pattern-oriented, emphasizing local temporal structures in classification and motif discovery (Ye and Keogh, 2009; Zakaria et al., 2012) or deviations from learned temporal signatures in anomaly detection and change-point detection.

2.2 Time Series Multimodal Models

Multimodal large language models (MLLMs) combine perceptual representations with language reasoning and have achieved strong performance in vision, video, and audio domains (Zhang et al., 2005; Wang et al., 2024; Mroueh et al., 2015). In contrast, multimodal modeling for time series remains relatively underexplored, partly due to the scarcity of paired temporal-text datasets. Existing time series MLLMs primarily align numerical sequences with language for tasks such as question answering, explanation generation, and zero-shot analysis. Representative approaches include ChatTime (Wang et al., 2025a), which treats time series as a “foreign language” for bimodal modeling and forecasting, ChatTS (Xie et al., 2025), which uses synthetic textual descriptions to align temporal attributes with LLMs, and ITFormer (Wang et al., 2025b), which connects a time series encoder to a frozen LLM for temporal-textual reasoning. Despite architectural and training differences, existing methods often exhibit unimodal bias, where language representations dominate temporal representations during multimodal training (Zhang et al., 2024c; Wang et al., 2020; Peng et al., 2022), leading models to over-rely on linguistic cues and underutilize fine-grained temporal information. Existing remedies for multimodal imbalance or repre-

sentation learning do not fully address this issue in text-time-series generative models. Temporal representations are often learned separately and only later integrated with the language pathway, limiting their participation in instruction-conditioned reasoning. Moreover, after multimodal fusion, training is typically governed primarily by the language-generation objective, without direct supervision on the temporal hidden states. As a result, optimization may favor a small subset of task-salient temporal features while inadequately preserving intermediate transitions and long-range temporal dependencies required for complex reasoning.

3 Preliminaries

3.1 Unified Token-Based Text-Time Series Task

We study a basic text-time series task, where a model takes a multivariate time series $\mathbf{X} \in \mathbb{R}^{D \times L}$ together with a textual condition $\mathbf{C} = [c_1, \dots, c_M]$ as input, and generates a textual output $\mathbf{Y} = [y_1, \dots, y_T]$. This can be formulated as a conditional generation problem $p(\mathbf{Y} | \mathbf{X}, \mathbf{C})$. The time series $\mathbf{X}$ provides structured numerical observations with temporal dependencies, while the textual condition $\mathbf{C}$ specifies how these observations should be interpreted and what information should be extracted. By introducing text as a flexible task interface, the model is no longer restricted to a fixed, task-specific output space, but can generalize across diverse time series reasoning tasks solely through different textual conditions, without modifying the model architecture, thereby enabling practical systems that help users better understand time series data ubiquitously arising across real-world domains. (Xie et al., 2025)

3.2 Challenges: Text-Dominated Optimization in Time Series MLLMs

We identify a fundamental optimization issue when training multimodal large language models (MLLMs) with time series inputs.

Multimodal large models for time series reasoning are typically trained using a text-generation objective,

$$\mathcal{L}_{\text{txt}} = - \sum_t y_t^\top \log p_\theta(y_t | T_{\text{txt}}, T_{\text{ts}}), \quad (1)$$

where supervision is applied entirely in the language space. Although the model receives both

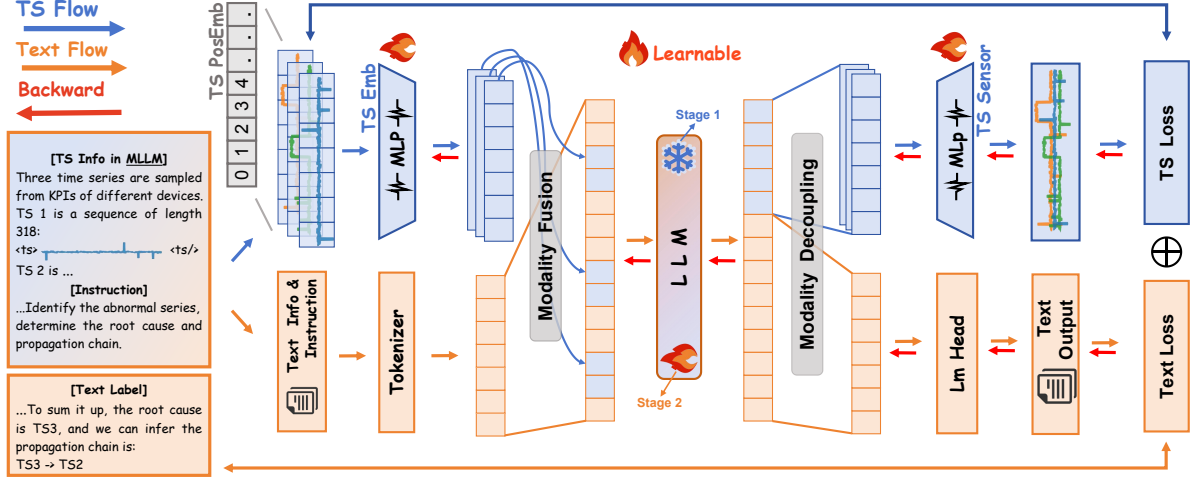


Figure 4: **Workflow of TimeSense.** The time series related modules are marked with blue lines, and the text related modules with orange lines. In Stage 1, only the TS Flow is trained, while in Stage 2, the TS Flow and Text Flow are jointly trained.

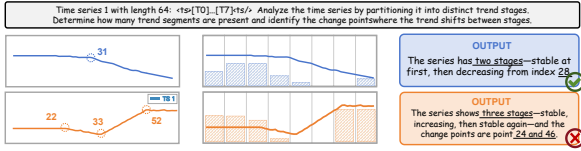


Figure 5: **Time series token embedding strength under increasing temporal regimes.** Gray grids denote different time series tokens. The embedding strength of each token is computed as the L2 norm of its embedding vector, with higher background bars indicating higher magnitude.

textual and temporal inputs, optimization is primarily driven by predicting textual tokens rather than preserving temporally grounded representations.

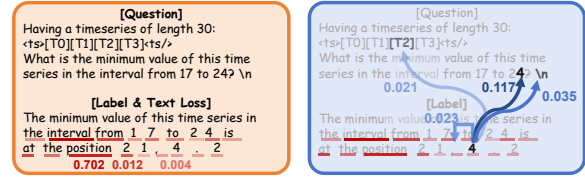
This creates a strong optimization imbalance across modalities:

$$\mathbb{E}[\|\nabla_{T_{ts}} \mathcal{L}_{txt}\|] \ll \mathbb{E}[\|\nabla_{T_{txt}} \mathcal{L}_{txt}\|]. \quad (2)$$

The imbalance is particularly severe for time series reasoning tasks. Numerically grounded outputs often occupy only a small fraction of the generated sequence, and small temporal inaccuracies typically induce only minor changes in token-level cross-entropy loss. As training progresses, gradients associated with temporal reasoning rapidly diminish,

$$\mathbb{E}[\|\nabla_{T_{ts}} \mathcal{L}_{txt}\|] \rightarrow 0, \quad (3)$$

once the model reaches a text-sufficient solution. Subsequent optimization mainly reinforces linguistic answer patterns instead of improving temporal understanding.



(a) Loss distribution.

(b) Attention distribution.

Figure 6: **Token-level loss and attention under text-dominated supervision.** (a) Token-wise training loss for a representative QA instance, where larger red values indicate higher loss and stronger optimization pressure. (b) Attention weights used when generating time series-related numerical outputs, with darker blue indicating stronger reliance on the attended tokens.

As a consequence, the model learns highly selective temporal representations. Let

$$H^{(ts)}(X) = [h_1, \dots, h_{LD}]$$

denote the token-level temporal embeddings. Under text-dominated supervision, optimization mainly updates temporal regions that are directly correlated with the language loss, while weakly supervising the remaining tokens:

$$\mathbb{E}[\|\nabla_{h_i} \mathcal{L}_{txt}\|] \approx 0, \quad \forall i \in S^c. \quad (4)$$

As a result, the learned representation increasingly depends on only a small subset of temporal tokens,

$$H^{(ts)}(X) \approx H^{(ts)}(\Pi_S X), \quad (5)$$

where S denotes a sparse task-relevant subset.

This selective optimization weakens intermediate temporal transitions and long-range dependencies, causing progressive loss of global temporal structure as temporal regimes become more complex. In more complex temporal regimes, some fine-grained transition information can be substantially underrepresented in the learned embeddings; for example, in the three-change-point case shown in Figure 5, the embedding strength around the intermediate change point is nearly zero, potentially hindering subsequent layers from accurately localizing this transition. Figure 6 further shows that optimization pressure and inference attention concentrate on only a small subset of tokens, while most temporal regions receive limited supervision.

4 Method

4.1 Overall Design

To mitigate text-dominated optimization, we propose a fusion-then-decoupling design for text-time series multimodal learning. The model first integrates textual and temporal tokens within a shared language-model backbone, enabling cross-modal interaction in a unified representation space. After fusion, we decouple the hidden states corresponding to the time series modality and provide them with explicit temporal supervision. This design preserves multimodal interaction while preventing temporal representations from being optimized solely through the language-generation objective.

Within this fusion-then-decoupling framework, we use temporal reconstruction as the supervision mechanism for the decoupled time series representations.

Given an input time series $X \in \mathbb{R}^{D \times L}$ and its temporal representation $H^{(\text{ts})}(X)$, the model reconstructs the original signal via

$$\mathcal{L}_{\text{ts}} = \|\text{Dec}(H^{(\text{ts})}(X)) - X\|_2^2. \quad (6)$$

Unlike pure text-generation objectives, which primarily optimize language outputs, the reconstruction objective provides dense supervision directly on temporal representations.

Rather than constituting a standalone contribution, the reconstruction objective serves as a modality-specific training signal that encourages these representations to retain both global trends and local temporal variations.

This encourages the model to preserve intermediate temporal transitions and long-range dependencies

rather than relying only on sparse task-relevant patterns.

We combine the reconstruction objective with the textual objective:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{txt}} + \lambda \mathcal{L}_{\text{ts}}, \quad (7)$$

where $\lambda > 0$ controls the strength of temporal supervision. Training is performed in two stages: we first optimize $\mathcal{L}_{\text{ts}}$ to obtain temporally grounded representations, and then jointly fine-tune the model with Eq. (7).

4.2 Model Architecture

Position-Sensitive Time Series Embedding.

Given an input multivariate time series $\mathbf{X} \in \mathbb{R}^{D \times L}$, we associate each timestep $t \in \{0, \dots, L-1\}$ with an explicit temporal coordinate $p_t \in \mathbb{R}$, e.g., $p_t = \frac{t}{L-1}$ (or any fixed positional code). For each channel d , we form a position-aware point representation

$$\tilde{\mathbf{x}}_t^{(d)} = [x_t^{(d)}; p_t] \in \mathbb{R}^2, \quad (8)$$

and obtain an augmented sequence $\tilde{\mathbf{X}}^{(d)} = [\tilde{\mathbf{x}}_0^{(d)}, \dots, \tilde{\mathbf{x}}_{L-1}^{(d)}] \in \mathbb{R}^{L \times 2}$. This explicit position attachment ensures that the representation cannot be invariant to index permutations, enabling the model to localize events (e.g., spikes, regime shifts) at the correct coordinates.

We then divide each $\tilde{\mathbf{X}}^{(d)}$ into non-overlapping patches of length P , yielding $N = \lceil L/P \rceil$ patches. The n -th patch of channel d is flattened as

$$\hat{\mathbf{x}}_n^{(d)} = \text{vec}(\tilde{\mathbf{X}}_n^{(d)}) \in \mathbb{R}^{2P}, \hat{\mathbf{X}} \in \mathbb{R}^{D \times N \times 2P}. \quad (9)$$

A shared MLP encoder $f_\phi : \mathbb{R}^{2P} \rightarrow \mathbb{R}^H$ maps each patch to a time series token:

$$\mathbf{T}_{\text{ts}}^{(d,n)} = f_\phi(\hat{\mathbf{x}}_n^{(d)}) \in \mathbb{R}^H, \mathbf{T}_{\text{ts}} \in \mathbb{R}^{D \times N \times H}. \quad (10)$$

Flattening channel and patch axes gives $\mathbf{T}_{\text{ts}}^{\text{flat}} \in \mathbb{R}^{(DN) \times H}$, which is inserted into the LLM token stream between special markers $\langle \text{ts} \rangle$ and $\langle \text{ts}/ \rangle$.

Modality Fusion and Decoupling. Let the fused sequence be processed by a shared Transformer backbone (the LLM):

$$\mathbf{H} = \text{LLM}([\langle \text{ts} \rangle, \mathbf{T}_{\text{ts}}^{\text{flat}}, \langle \text{ts}/ \rangle, \mathbf{T}_{\text{txt}}]) \in \mathbb{R}^{T \times H}. \quad (11)$$

We then decouple the hidden states into time series and text parts:

$$\mathbf{H}^{(\text{ts})} = \mathbf{H}_{1:DN} \in \mathbb{R}^{(DN) \times H}, \mathbf{H}^{(\text{txt})} = \mathbf{H}_{DN+1:T}. \quad (12)$$

Disentangled Time Series Decoder. To prevent temporal representations from being implicitly shaped only by text loss (unimodal bias) and to stabilize encoding against local perturbations, we impose an explicit reconstruction objective *through a decoder that is disentangled from language generation*. Concretely, a dedicated MLP decoder $g_\psi : \mathbb{R}^H \rightarrow \mathbb{R}^P$ maps each time series hidden token back to a length- P patch:

$$\hat{\mathbf{X}}_{\text{rec}}^{(d,n)} = g_\psi(\mathbf{H}_{(d,n)}^{(\text{ts})}) \in \mathbb{R}^P, \hat{\mathbf{X}}_{\text{rec}} \in \mathbb{R}^{D \times N \times P}. \quad (13)$$

Inverse patching merges reconstructed patches into the recovered sequence $\mathbf{X}_{\text{rec}} \in \mathbb{R}^{D \times L}$. We supervise it with an MSE reconstruction loss:

$$\mathcal{L}_{\text{ts}} = \|\mathbf{X}_{\text{rec}} - \mathbf{X}\|_2^2. \quad (14)$$

Intuitively, minimizing $\mathcal{L}_{\text{ts}}$ forces $\mathbf{H}^{(\text{ts})}$ to preserve globally sufficient temporal information rather than overfitting to a few locally salient fragments; later, in [Appendix D](#), we formalize this as a non-vanishing reconstruction lower bound for subset-only embeddings.

Time Series Warm-up and Joint Finetuning. We adopt a two-stage training strategy. First, the time series encoder and decoder are optimized using the reconstruction objective while freezing the LLM:

$$\min_{\phi, \psi} \mathcal{L}_{\text{ts}}. \quad (15)$$

We then jointly finetune the multimodal model with both textual supervision and temporal reconstruction:

$$\min_{\Theta} \mathcal{L}_{\text{txt}} + \lambda \mathcal{L}_{\text{ts}}, \quad (16)$$

where λ controls the strength of temporal supervision.

5 Benchmark

To systematically evaluate a model’s ability to understand and reason over temporal data in multimodal settings, we construct **EvalTS**, a diagnostic benchmark covering a wide range of time series reasoning tasks. As illustrated in [Figure 3](#), EvalTS decomposes temporal understanding into three progressive levels: *Atomic Understanding*, *Molecular Reasoning*, and *Compositional Tasks*. This structure is inspired by human cognitive processes for perceiving, interpreting, and integrating temporal information.

Table 1: Overview of existing benchmark datasets for evaluating time series understanding in LLMs. **Types** denote the question format: MCQ refers to multiple-choice questions, T/F denotes true-or-false questions, and OG indicates open-generation tasks that require free-form textual responses.

Dataset	Types	Max.S	Max.D
TSQA-A(ChatTime)	MCQ	2	1-D
TSQA-B(TimeMQA)	MCQ, T/F	2	1-D
BEDTime	MCQ, T/F	2	1-D
TimeSeriesExam	MCQ, T/F	2	2-D
EvalTS	OG	5	5-D

Atomic Understanding targets the most fundamental temporal units, such as identifying extreme, detecting change points, recognizing global trends, and retrieving values at specific indices. Tasks are designed for both single-series and multi-series settings, with unconstrained sequence lengths to reflect real-world variability.

Molecular Reasoning builds on atomic cognition by requiring the model to reason about relationships between multiple segments or series. Tasks include local trend segmentation, inter-series comparison, and analysis of relative changes. These tasks test a model’s ability to combine trend-level structure with fine-grained value understanding.

Compositional Tasks assess high-level reasoning over complex or multivariate time series. Tasks include rule-based anomaly detection, anomaly ranking, and causal reasoning. These scenarios simulate real-world challenges where temporal understanding is neither local nor static, but integrative and context-dependent.

Compared to existing benchmarks ([Table 1](#)), EvalTS introduces two key advances. First, it supports up to 5-dimensional inputs, enabling evaluation of a model’s capacity to reason over multivariate dynamics, while prior datasets are mostly limited to 1D settings. Second, EvalTS adopts an open-ended question-answering format, where models generate free-form answers directly from the time series and textual instructions, without being provided with predefined answer choices or binary true/false statements. This differs from existing benchmarks whose questions remain structurally constrained as multiple-choice or true/false tasks, even when model responses are generated in an open-ended manner. By removing such predefined answer spaces, EvalTS encourages models to demonstrate genuine temporal understanding and

natural language reasoning rather than selecting or reproducing answers from a fixed set of candidates.

To enable reliable evaluation of free-form responses, we employ an LLM-based parser to extract task-relevant structured information from model outputs. The LLM is used solely for semantic extraction rather than scoring; the extracted fields are subsequently compared with the reference features using deterministic task-specific scripts. This evaluation pipeline accommodates diverse natural-language response formats without enforcing a fixed answer template, while preserving strict and reproducible scoring of the underlying temporal semantics.

Full task definitions and examples are provided in [Appendix E](#). The prompt used for LLM-based parsing is provided in [Figure 16](#), while the corresponding reference answer cases used for deterministic scoring are provided in [Figure 15](#).

6 Experiments

6.1 Setup

Datasets. We evaluate all models on the EvalTS introduced in [section 5](#), where task definitions and example queries are described in [Appendix E](#). Our main experiments are conducted on four time series question answering benchmarks: TimeMQA ([Kong et al., 2025](#)), BEDTime ([Sen et al., 2025](#)), TimeSeriesExam ([Cai et al., 2024](#)), and ChatTime ([Wang et al., 2025a](#)). These datasets differ in question format and input representation. TimeMQA and BEDTime contain multiple-choice and true/false questions with one-dimensional time series inputs. TimeSeriesExam supports the same question formats but additionally includes two-dimensional inputs. ChatTime focuses on multiple-choice questions with one-dimensional inputs. To expand the size of the evaluation set, we adopt the open-source dataset released with ChatTime and follow the original evaluation protocol without modification.

Models. As mentioned in [Appendix F](#), we train TimeSense and compare against GPT5, ChatTS ([Xie et al., 2025](#)), 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. To investigate temporal recognition capabilities across different modalities, we also include multimodal versions of these two models, namely GPT5-VL and Qwen2.5-VL. Since Qwen2.5-VL does not have a 14B version, we uti-

lize the 7B version for evaluation. For other temporal modality models, we select ChatTS, TimeMQA, ChatTime, and ITFormer for comparative evaluation.

Metrics. For multiple-choice and true/false questions, we directly use accuracy as the evaluation metric. For generative questions in EvalTS, we employ Qwen2.5-32B-Instruct to extract key numerical content from model responses. For single-feature questions, accuracy is used as the metric, while for tasks involving multiple points or segments, we utilize the F1 score for evaluation. To account for minor annotation imprecision, we allow a small tolerance when evaluating questions involving temporal coordinates.

Results. The experiment results are summarized in [Table 2](#). Across most categories, TimeSense achieves the best performance, substantially outperforming other baselines. For a controlled comparison, all SFT baselines are fine-tuned on the same training data and for the same number of optimization steps as TIMESENSE. Under these conditions, TIMESENSE achieves the strongest performance across most reported metrics. This controlled comparison indicates that the observed gains are not merely attributable to differences in task-specific data exposure or training duration. TimeSense exhibits even more pronounced capabilities in TimeSeriesExam and EvalTS, where the time series dimensions are greater and temporal variations are more complex. The performance of current T+TS models varies significantly across different tasks, reflecting the poor compatibility of existing baselines with various temporal tasks. This further highlights TimeSense’s adaptability to diverse temporal tasks.

6.2 Further Analysis

Results on EvalTS. We report results for all three difficulty levels in EvalTS, including Atomic Understanding, Molecular Reasoning, and Compositional Tasks, as summarized in [Table 3](#). Model performance differs substantially across these levels.

Atomic tasks evaluate basic temporal reasoning abilities. In particular, the Extreme tasks mainly require simple numerical comparisons to identify extreme, while the Trend tasks focus on coarse global trend recognition, such as determining whether a sequence increases or decreases. Most models therefore achieve near-perfect performance on these tasks. The lower scores observed for several mod-

Table 2: Main Results (higher is better). Best results are highlighted in **red**, and second-best results are in **blue**.

Model	Modality	TSQA-A	TSQA-B		TimeSeriesExam		BEDTime		EvalTS		
		MCQ	T/F	MCQ	T/F	MCQ	T/F	MCQ	A	M	C
GPT5	T	0.587	0.838	0.683	0.691	0.534	0.614	0.538	0.695	0.506	0.351
GPT5-VL	T+V	0.781	0.775	0.667	0.108	0.164	0.746	0.706	0.655	0.484	0.202
Qwen2.5-Instruct	T	0.413	0.675	0.350	0.179	0.284	0.000	0.122	0.606	0.481	0.305
Qwen2.5-VL	T+V	0.521	0.750	0.575	0.559	0.491	0.695	0.594	0.550	0.613	0.313
ChatTS	T+TS	0.657	0.013	0.017	0.429	0.379	0.091	0.482	0.547	0.402	0.371
TimeMQA	T+TS	0.371	0.838	0.692	0.274	0.310	0.284	0.375	0.450	0.220	0.227
ChatTime	T+TS	0.904	–	0.017	–	0.172	–	0.102	–	–	–
ITFormer	T+TS	0.371	0.225	0.289	0.087	0.048	0.203	0.130	0.218	0.169	0.128
Qwen2.5+SFT	T	0.308	0.713	0.450	0.560	0.362	0.558	0.315	0.548	0.403	0.491
ChatTS+SFT	T+TS	0.748	0.688	0.517	0.607	0.543	0.716	0.508	0.728	0.521	0.567
TimeMQA+SFT	T+TS	0.377	0.775	0.625	0.000	0.000	0.279	0.173	0.409	0.286	0.200
ChatTime+SFT	T+TS	0.287	0.000	0.242	0.012	0.000	0.076	0.041	0.151	0.077	0.082
ITFormer+SFT	T+TS	0.392	0.650	0.433	0.512	0.388	0.426	0.305	0.282	0.356	0.154
TimeSense	T+TS	0.781	0.825	0.692	0.726	0.716	0.843	0.817	0.930	0.854	0.713

Note: T: Text-only; V: Visual; TS: time series; MCQ: Multiple Choice Questions; T/F: True/False; A: Atomic Understanding Task; M: Molecular Reasoning Task; C: Compositional Tasks. ChatTime’s outputs only include options for multiple-choice questions and are solely used for MCQ evaluation.

Table 3: Results of EvalTS tasks(higher is better). Best results are highlighted in **red**, and second-best results are in **blue**.

Model	Atomic Understanding										Molecular Reasoning			Compositional Tasks	
	Change Point		Extreme		Spike		Trend		Period		Segment	Comparison	Relative	RCA	AD
	Uni	Multi	Uni	Multi	Uni	Multi	Uni	Multi	Uni	Multi					
GPT5	0.320	0.355	0.995	0.891	0.900	0.841	0.913	0.811	0.467	0.454	0.345	0.507	0.667	0.236	0.466
GPT5-VL	0.800	0.607	0.425	0.188	0.790	0.779	0.970	0.917	0.580	0.497	0.450	0.401	0.601	0.119	0.285
Qwen2.5-Instruct	0.593	0.269	0.937	0.818	0.467	0.498	0.930	0.831	0.337	0.375	0.424	0.254	0.764	0.174	0.436
Qwen2.5-VL	0.487	0.174	0.297	0.281	0.787	0.758	0.943	0.877	0.423	0.469	0.580	0.515	0.745	0.215	0.411
ChatTS	0.383	0.485	0.310	0.174	0.887	0.275	0.960	0.886	0.550	0.556	0.315	0.272	0.620	0.157	0.586
TimeMQA	0.447	0.256	0.667	0.537	0.233	0.363	0.742	0.600	0.273	0.378	0.295	0.183	0.183	0.201	0.252
ITFormer	0.303	0.405	0.028	0.025	0.32	0.286	0	0	0.393	0.421	0.066	0.303	0.135	0.039	0.215
TimeSense (Ours)	0.967	0.854	1.000	1.000	0.960	0.900	1.000	0.994	0.800	0.821	0.940	0.765	0.856	0.708	0.718

els are mainly caused by difficulties in generating outputs in the required textual or numerical format.

As task difficulty increases, most models exhibit a clear performance drop relative to the Atomic tasks. Molecular tasks place greater emphasis on analyzing and comparing overall temporal trends and patterns. Models with vision-language capabilities generally perform better on these tasks.

Compositional tasks involve more complex temporal structures and question formulations, requiring the integration of multiple reasoning abilities. The relatively low performance on these tasks indicates that such compositional temporal reasoning remains challenging for current models. More results can be found in [Appendix B](#), where we provide detailed performance of each model on the sub-tasks of EvalTS and analyze the strengths of different models.

Modality Comparison. Different modality combinations exhibit distinct strengths and limitations in temporal understanding. Pure text models, including large closed-source models such as GPT5, capture global trends and isolated point features but perform poorly on tasks requiring relational comparison and complex temporal reasoning. Vision modalities can model internal temporal structures but struggle with precise point-level recognition. By directly incorporating temporal signals, temporal modalities address the limitations of text and vision modalities and achieve more balanced performance across tasks.

Impact of Trend Segments of Time Series. To assess how the number of trend segments in time series data affects model performance, we conduct experiments on the Segment task by varying the number of segments from 1 to 5. As shown in Fig-

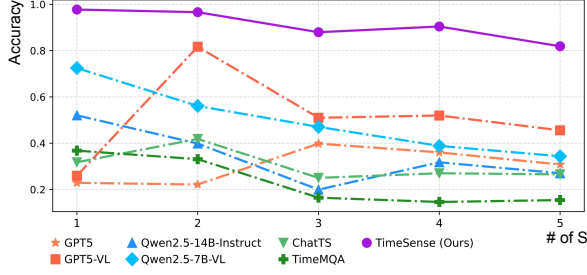


Figure 7: Performance comparison on the Segment task with varying numbers of time series segments.

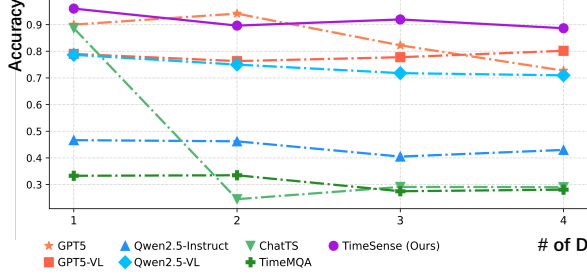


Figure 8: Performance comparison on the Spike task with varying numbers of time series dimensions.

Figure 7, TimeSense consistently outperforms other baselines across all configurations. Notably, as the number of segments increases, the performance gap between TimeSense and other models widens, indicating its superior capability in handling complex temporal structures.

Impact of Time Series Dimensions. We further investigate the effect of increasing the number of time series dimensions on the Spike task. By varying the number of dimensions from 1 to 4, Figure 8 shows that TIMESENSE maintains robust performance across all evaluated dimensional settings and achieves the best performance in three of the four settings.

Although most baseline models exhibit performance drops at one or more dimensional settings, TIMESENSE maintains consistently high accuracy across all four settings, reflecting the limitations in handling high-dimensional time series data. Additionally, we observe slight improvements in some general models with increased dimensions, possibly due to the higher proportion of temporal input guiding the model to better localize spike points.

Ablation Studies on Architecture and Training To examine the contribution of different structural designs, we construct three model variants: w/o Stage 1, w/o Rec(onstruction), and w/o All, which respectively remove stage 1 separate temporal-module training, the temporal reconstruc-

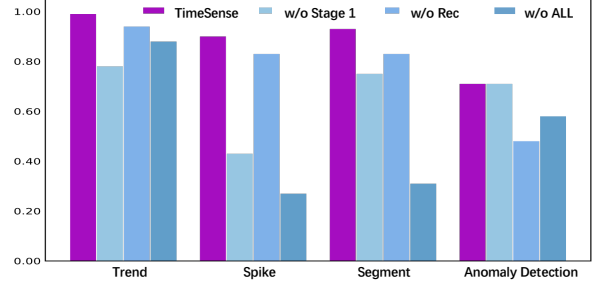


Figure 9: **Ablation study of different model variants.** Performance comparison of TimeSense and its ablated variants on selected EvalTS tasks.

Table 4: Ablation on EvalTS.

Variant	Atomic-Uni	Atomic-Multi	Molecular	Compositional
TimeSense	0.945	0.914	0.854	0.713
w/o Stage 1	0.835	0.584	0.495	0.615
w/o Rec	0.921	0.799	0.439	0.566
w/o All	0.618	0.475	0.402	0.372

tion mechanism, and all temporal training. Most variant results fall in regions where the full TimeSense structure performs better, indicating that the complete design achieves the strongest overall performance. Although some variants outperform the full model in limited cases, different variants exhibit distinct performance patterns, and none consistently surpass the full TimeSense structure.

7 Conclusion

We presented EvalTS, a diagnostic benchmark for open-generation text–time series reasoning, and showed that existing multimodal LLMs struggle with fine-grained temporal understanding and compositional reasoning. To address the text-dominated optimization underlying these limitations, we proposed TimeSense, a fusion-then-decoupling framework that first integrates temporal and textual information within a shared language-model backbone and then decouples temporal hidden states for direct temporal supervision. This design enables reconstruction-based supervision to preserve temporal information without treating reconstruction as a standalone modeling objective. Experiments on EvalTS and external benchmarks demonstrate consistent improvements in temporal reasoning. Overall, our results suggest that explicitly structuring how temporal information is fused, supervised, and retained provides an effective direction for text–time series multimodal learning.

8 Limitation

EvalTS is a controlled synthetic benchmark designed to systematically evaluate temporal reasoning in multimodal LLMs. Although this design enables fine-grained control over task complexity and temporal structure, it does not fully capture the variability and noise of real-world time series data, such as irregular sampling, missing values, and domain-specific semantics.

In addition, TimeSense is primarily evaluated on the Qwen2.5 model family. Its effectiveness on other LLM architectures and larger-scale settings remains to be further studied.

Finally, while TimeSense improves text-time series multimodal reasoning, the current study does not evaluate high-stakes real-world applications such as medical or industrial decision-making. Additional validation on real-world datasets and expert-defined tasks is necessary before deployment in such settings.

9 Ethics Statement

This work studies text-time series multimodal reasoning under controlled research settings. EvalTS is intended solely for research evaluation and should not be interpreted as evidence that a model is suitable for clinical, financial, industrial, or other high-stakes applications without additional domain-specific validation and human oversight.

The benchmark and experiments are constructed using synthetic or publicly available research data and open-source software under their respective licenses and access conditions. We do not collect or release personally identifiable information. Evaluation samples and generated task instances were manually reviewed to reduce annotation errors and inappropriate content.

Because this work relies on large language models, model outputs may still reflect biases, inaccuracies, or misleading temporal correlations inherited from pretraining data. AI-assisted tools were used for limited writing and editing support, and all technical content and experimental claims were reviewed by the authors.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No.2024YFB4505903), and the Beijing National Research Center for Information Science and Technology (BNRist) key projects.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.
- Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yanyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, and 22 others. 2024. [An introduction to vision-language modeling](#). *Preprint*, arXiv:2405.17247.
- Yifu Cai, Arjun Choudhry, Mononito Goswami, and Artur Dubrawski. 2024. Timeseriesexam: A time series understanding exam. *arXiv preprint arXiv:2410.14752*.
- Angus Dempster, François Petitjean, and Geoffrey I Webb. 2020. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495.
- Chang Gong, Chuzhe Zhang, Di Yao, Jingping Bi, Wenbin Li, and Yongjun Xu. 2024. Causal discovery from temporal data: An overview and new perspectives. *ACM Computing Surveys*, 57(4):1–38.
- Yangdong He and Jiabao Zhao. 2019. [Temporal convolutional networks for anomaly detection in time series](#). *Journal of Physics: Conference Series*, 1213(4):042050.
- Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. [Time-llm: Time series forecasting by reprogramming large language models](#). In *International Conference on Learning Representations*, volume 2024, pages 23857–23880.
- Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. 2025. [Time-MQA: Time series multi-task question answering with context enhancement](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29736–29753, Vienna, Austria. Association for Computational Linguistics.
- Marco Lippi, Matteo Bertini, and Paolo Frasconi. 2013. 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.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in*

- neural information processing systems*, 36:34892–34916.
- Yeqi Liu, Chuanyang Gong, Ling Yang, and Yingyi Chen. 2020. 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.
- Amy McGovern, Derek H Rosendahl, Rodger A Brown, and Kelvin K Droegemeier. 2011. Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction. *Data Mining and Knowledge Discovery*, 22(1):232–258.
- Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. 2023. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29.
- Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. 2015. Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 2130–2134. IEEE.
- Francisco Javier Nogales, Javier Contreras, Antonio J Conejo, and Rosario Espínola. 2002. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17(2):342–348.
- Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. 2022. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8238–8247.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Medhasweta Sen, Zachary Gottesman, Jiaxing Qiu, C Bayan Bruss, Nam Nguyen, and Tom Hartvigsen. 2025. Bedtime: A unified benchmark for automatically describing time series. *arXiv preprint arXiv:2509.05215*.
- Sima Siامي-Namini, Neda Tavakoli, and Akbar Siامي Namin. 2018. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1394–1401. Ieee.
- 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*.
- Seemant Tiwari. 2023. Segmentation and clustering of time series data. In *2023 International Conference for Advancement in Technology (ICONAT)*, pages 1–6. IEEE.
- Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. 2025a. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12694–12702.
- Weiyao Wang, Du Tran, and Matt Feiszli. 2020. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. 2024. Internvideo2: Scaling foundation models for multimodal video understanding. In *Computer Vision – ECCV 2024*, pages 396–416, Cham. Springer Nature Switzerland.
- Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang Li, and Zhongyu Wei. 2025b. Itformer: bridging time series and natural language for multi-modal qa with large-scale multitask dataset. In *Proceedings of the 42nd International Conference on Machine Learning, ICML’25*. JMLR.org.
- Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. 2025. [Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning](#). *Proceedings of the VLDB Endowment*, 18(8):2385–2398.
- Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. 2025. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.
- Jiexia Ye, Weiqi Zhang, Ke Yi, Yongzi Yu, Ziyue Li, Jia Li, and Fugee Tsung. 2024. A survey of time series foundation models: Generalizing time series representation with large language model. *arXiv preprint arXiv:2405.02358*.
- Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956.
- Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. 2023. Temporal data meets llm—explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*.

- Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. 2012. Clustering time series using unsupervised-shapelets. In *2012 IEEE 12th international conference on data mining*, pages 785–794. IEEE.
- G Peter Zhang. 2003. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175.
- Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024a. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644.
- Ruofei Zhang, Zhongfei Zhang, Mingjing Li, Wei-Ying Ma, and Hong-Jiang Zhang. 2005. 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, pages 846–851. IEEE.
- Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. 2024b. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*.
- Yedi Zhang, Peter E. Latham, and Andrew M Saxe. 2024c. [Understanding unimodal bias in multimodal deep linear networks](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 59100–59125. PMLR.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyang Luo. 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)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.
- Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

A Responsible Research Details

A.1 Artifacts, Licenses, and Intended Use

This work uses and creates scientific artifacts. We create EVALTS, the ChronGen generation procedure, and the TIMESENSE model/training recipe. We also use external benchmarks and models, including TimeMQA (Kong et al., 2025), BED-Time (Sen et al., 2025), TimeSeriesExam (Cai et al., 2024), ChatTime (Wang et al., 2025a), ChatTS (Xie et al., 2025), ITFormer (Wang et al., 2025b), Qwen2.5-family models, and LLAMAFACTORY (Zheng et al., 2024). Their creators are cited

in Sections 2, 5, and 6 where the artifacts are introduced or used.

EVALTS is intended solely for research evaluation of text-time series reasoning under controlled synthetic settings. Performance on EVALTS should not be interpreted as evidence of suitability for clinical, financial, industrial, or other high-stakes applications without additional domain-specific evaluation and validation. We do not redistribute third-party benchmarks, model checkpoints, or software artifacts beyond their original licensing and access conditions.

A.2 Data Content, Privacy, and Documentation

EVALTS consists of synthetic numerical time series and templated English task descriptions generated by ChronGen. To the best of our knowledge, it does not contain names, contact information, user identifiers, demographic attributes, social-media content, or other personal data. The text templates describe temporal phenomena such as trends, spikes, change points, and anomaly rules, and do not contain intentionally offensive content. We manually inspected the templates and example outputs used in the paper for personally identifying information and offensive content. For third-party benchmarks, we rely on the original artifact creators’ documentation and use them only for research evaluation.

The benchmark documentation in [section 5](#) describes the task taxonomy, dimensionality, answer formats, and train/evaluation separation. Additional train/evaluation split controls include different ChronGen parameters and seeds, Pearson-similarity filtering at a threshold of 0.9, a sampled train/evaluation mean Pearson correlation of 0.4999, and a Wasserstein distance of 0.706. Additional examples and implementation details are provided in [Appendix E](#).

A.3 Computational Experiment Details

The computational experiments use multimodal LLMs, text-only LLMs, vision-language models, and time series baselines. Model sizes, training-time parameters, inference-time parameters, wall-clock training time, and 300-sample inference time are reported in Section 6. Training uses 8 A800 GPUs. The reconstruction decoder is used only during training and is removed at inference.

The main hyperparameters are reported in Section 6: per-device batch size 1, gradient accumulation 32, learning rate 10^{-5} , warmup ratio 0.02,

300 warm-up finetuning steps, and 1200 joint finetuning steps. Patch length and reconstruction-weight sensitivity are also reported in Section 6. Open-generation EvalTS answers are parsed with Qwen2.5-32B-Instruct only to extract structured fields; deterministic rule scripts compute the final score. The parser/scorer was manually verified on 500 examples.

A.4 Human Validation Protocol

Human validation is limited to checking the EvalTS answer-extraction and rule-scoring pipeline on synthetic examples. Annotators were shown the generated question, the reference answer or reference feature fields, the model output, and the parser-extracted fields. They were instructed to mark whether the extracted fields preserved the answer needed by the deterministic scorer and to correct the fields when extraction failed. The task did not collect personal or demographic information from annotators.

No crowdworker-generated text is used as training or evaluation data. The validation was conducted by members of the research team familiar with the tasks. Because the task involved validation of synthetic examples and model outputs rather than collection of personal or sensitive data, no consented personal dataset is created by this step.

A.5 Potential Risks and AI-Assistant Use

Potential risks are discussed in Section 8. The main risk is over-interpreting performance on a controlled synthetic benchmark as evidence of robustness in real deployments. Time series decisions in healthcare, finance, transportation, or industrial monitoring may require calibrated uncertainty, domain expertise, and safety constraints that are outside the scope of this paper. The system can also produce fluent but incorrect explanations if deployed without task-specific validation.

AI assistants were used during writing and revision for language editing, LaTeX editing, checklist organization, and summarizing reviewer concerns.

B EvalTS Results Table

B.1 Comparative Performance Overview

As illustrated in Table 5, our proposed model achieves better performance across evaluated tasks and dimensions in our experiments. It maintains a significant performance margin over both specialized time series baselines and general-purpose

large language models (LLMs). Specifically, in tasks requiring high-precision localization such as *extreme* and *trend*, TIMESENSE reaches near-optimal accuracy, demonstrating its superior capability in capturing both local anomalies and long-term temporal dependencies.

B.2 Task-Specific Observations

Robustness in Extremum and Trend Detection:

TIMESENSE exhibits exceptional stability in the *extreme* task, where it outperforms all baseline models by a substantial margin. While competitive models like GPT5 and QWEN2.5-INSTRUCT show varying degrees of success, they fail to match the consistency of TIMESENSE across multiple dimensions.

Localization Tasks: In the *change_point* and *spike* tasks, vision-language variants such as GPT5-VL demonstrate competitive results, often emerging as the second-best performers. However, TIMESENSE still retains the lead, particularly in higher-dimensional settings, highlighting the efficacy of its architectural design in handling increased temporal complexity.

B.3 Scalability Across Dimensions

A critical observation from the experimental results is the performance trend as the input *Dimension* increases. While several baseline models experience performance degradation or significant fluctuations when moving from lower to higher dimensions, TIMESENSE exhibits remarkable invariance. Its performance remains consistently high across all four dimensions, indicating that the model successfully learns dimension-agnostic temporal features, which is essential for scaling to real-world multivariate time series data.

B.4 Comparison with Vision-Language Models

The results underscore a performance gap between specialized temporal models and visual-modality-based models. Although GPT5-VL and QWEN2.5-VL leverage visual representations to interpret time series data, they struggle with granular tasks like *extreme* detection compared to the direct temporal feature extraction approach of TIMESENSE. This validates our hypothesis that dedicated temporal modeling is more effective than translating sequences into the visual domain for high-precision analytical tasks.

C Additionally Visualization of Temporal Embeddings

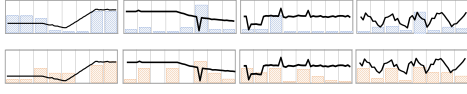


Figure 10: Visualization examples of temporal embeddings from variants models on different EvalTS tasks. The upper row shows embeddings from the model trained without joint training, while the lower row shows embeddings from the model trained with joint training.

As illustrated in Figure 10, we present a comparative visualization of temporal embeddings generated by models trained with and without joint training across various EvalTS tasks. The results indicate that models trained with joint training produce temporal embeddings that more finely capture the nuances of different segments within the time series. This suggests that joint training facilitates the learning of more comprehensive temporal features, thereby enhancing the model’s performance in multimodal time series understanding and reasoning tasks.

D Theoretical Analysis

D.1 Incomplete Temporal Embedding: Subset-only Representations Imply Irreducible Reconstruction Error

Setup and notation. Let $X \in \mathbb{R}^{D \times L}$ and write its vectorization as $x := \text{vec}(X) \in \mathcal{X} \subset \mathbb{R}^m$ with $m := DL$, where $\mathcal{X}$ is a bounded domain representing the data support. For any index subset $S \subseteq \{1, \dots, m\}$, let $\Pi_S : \mathbb{R}^m \rightarrow \mathbb{R}^m$ denote the coordinate projection that keeps entries in S and zeros out entries in S^c . Let $H^{(\text{ts})} : \mathbb{R}^m \rightarrow \mathbb{R}^q$ denote the hidden time series representation. A *subset-only* representation means there exists a measurable function g such that

$$H^{(\text{ts})}(x) = g(\Pi_S x), \quad \forall x \in \mathcal{X}. \quad (17)$$

Table 5: Results across different dimensions

Task	Dimension	TimeSense	TimeMQA	CharTS	GPT5	GPT5-VL	Qwen2.5-Instruct	Qwen2.5-VL
change_point	1	0.967	0.447	0.383	0.320	0.800	0.593	0.487
	2	0.848	0.227	0.490	0.250	0.714	0.237	0.259
	3	0.855	0.259	0.529	0.405	0.552	0.276	0.227
	4	0.858	0.282	0.439	0.300	0.575	0.294	0.250
extreme	1	1.000	0.667	0.310	0.990	0.425	0.937	0.297
	2	1.000	0.569	0.172	1.000	0.271	0.860	0.262
	3	1.000	0.535	0.188	0.955	0.167	0.832	0.299
	4	1.000	0.505	0.161	0.933	0.117	0.760	0.282
spike	1	0.960	0.233	0.887	0.900	0.790	0.467	0.787
	2	0.896	0.335	0.245	0.941	0.763	0.462	0.750
	3	0.919	0.375	0.291	0.822	0.778	0.505	0.818
	4	0.886	0.381	0.290	0.727	0.802	0.530	0.710
trend	1	1.000	0.742	0.960	0.913	0.970	0.930	0.943
	2	0.989	0.661	0.911	0.844	1.000	0.839	0.933
	3	0.993	0.620	0.906	0.761	0.929	0.822	0.909
	4	0.998	0.534	0.847	0.829	0.857	0.833	0.802

Let $\text{Dec} : \mathbb{R}^q \rightarrow \mathbb{R}^m$ be an arbitrary decoder.

[Subset-only representation yields a non-vanishing minimax reconstruction lower bound] Assume $H^{(\text{ts})}$ is subset-only in the sense of Eq. (17). Define the *subset diameter* relative to the domain $\mathcal{X}$:

$$c_S := \frac{1}{2} \sup \left\{ \|x - x'\|_2 : x, x' \in \mathcal{X}, \Pi_S x = \Pi_S x' \right\}. \quad (18)$$

Then for any decoder Dec ,

$$\sup_{x \in \mathcal{X}} \|\text{Dec}(H^{(\text{ts})}(x)) - x\|_2 \geq c_S. \quad (19)$$

In particular, if there exist $x, x' \in \mathcal{X}$ with $\Pi_S x = \Pi_S x'$ but $x \neq x'$, then $c_S > 0$ and the reconstruction error cannot vanish.

Fix any decoder Dec . Consider any pair $x, x' \in \mathcal{X}$ with $\Pi_S x = \Pi_S x'$. By the subset-only property, $H^{(\text{ts})}(x) = g(\Pi_S x) = g(\Pi_S x') = H^{(\text{ts})}(x')$. Consequently, the decoder outputs coincide: $\hat{x} := \text{Dec}(H^{(\text{ts})}(x)) = \text{Dec}(H^{(\text{ts})}(x'))$. By the triangle inequality,

$$\begin{aligned} \|x - x'\|_2 &\leq \|x - \hat{x}\|_2 + \|\hat{x} - x'\|_2 \\ &\leq 2 \max \{ \|x - \hat{x}\|_2, \|x' - \hat{x}\|_2 \}. \end{aligned}$$

Thus, $\sup_{u \in \{x, x'\}} \|\text{Dec}(H^{(\text{ts})}(u)) - u\|_2 \geq \frac{1}{2} \|x - x'\|_2$. Taking the supremum over all such pairs (x, x') yields Eq. (19).

D.2 Unimodal Bias: Joint Objective Restores Gradient Visibility

Setup. Let $T_{\text{ts}} = \text{Tok}_\phi(X)$ be the time series token sequence. Consider the joint finetuning objective:

$$\begin{aligned} \mathcal{L}_{\text{joint}}(\theta, \phi, \psi) &= \mathcal{L}_{\text{txt}}(\theta, \phi) \\ &\quad + \lambda \mathcal{L}_{\text{ts}}(\theta, \phi, \psi), \lambda > 0. \end{aligned} \quad (20)$$

We analyze gradients with respect to ϕ . We treat gradients ∇ as row vectors to align with the Jacobian notation $J = \partial(\cdot)/\partial(\cdot)$.

Assumption (Non-degenerate reconstruction head). We assume the reconstruction head is locally non-degenerate on the reachable manifold: there exists $\mu > 0$ such that for all reachable embeddings u , $\sigma_{\min}(J_{\text{Dec}_\psi}(u)) \geq \mu$.

[Joint objective eliminates TS-ignored stationary points] Fix (θ, ψ) . Assume there exists ϕ^* such that $\nabla_{T_{\text{ts}}} \mathcal{L}_{\text{txt}}(\theta, \phi^*) = 0$ (locally insensitive to TS

tokens). If $\mathcal{L}_{ts}(\theta, \phi^*, \psi) > 0$ and the parameterization $\phi \mapsto T_{ts}$ has a Jacobian of full row rank at ϕ^* , then ϕ^* is not a stationary point of $\mathcal{L}_{joint}$.

By the chain rule, the gradient of the joint objective is:

$$\nabla_{\phi} \mathcal{L}_{joint} = \left(\nabla_{T_{ts}} \mathcal{L}_{txt} + \lambda \nabla_{T_{ts}} \mathcal{L}_{ts} \right) \frac{\partial T_{ts}}{\partial \phi}. \quad (21)$$

At ϕ^* , since $\nabla_{T_{ts}} \mathcal{L}_{txt} = 0$, this simplifies to:

$$\nabla_{\phi} \mathcal{L}_{joint} = \lambda \nabla_{T_{ts}} \mathcal{L}_{ts} \frac{\partial T_{ts}}{\partial \phi}.$$

First, we show $\nabla_{T_{ts}} \mathcal{L}_{ts} \neq 0$. Let $u := H_{\theta}^{(ts)}(T_{ts})$ and $r := \text{Dec}_{\psi}(u) - X$. Differentiating $\mathcal{L}_{ts} = \|r\|_2^2$ w.r.t. u :

$$\begin{aligned} \|\nabla_u \mathcal{L}_{ts}\|_2 &= 2\|r\|_2 J_{\text{Dec}_{\psi}}(u) \\ &\geq 2\|r\|_2 \sigma_{\min}(J_{\text{Dec}_{\psi}}(u)) \\ &\geq 2\|r\|_2 \mu. \end{aligned}$$

Since $\mathcal{L}_{ts} > 0$, we have $\|r\|_2 > 0$, implying $\nabla_u \mathcal{L}_{ts} \neq 0$. Assuming the encoder $H_{\theta}^{(ts)}$ does not collapse gradients (i.e., its Jacobian is invertible or the gradient is not in its kernel), we have $v := \nabla_{T_{ts}} \mathcal{L}_{ts} \neq 0$.

Finally, since we assume $\frac{\partial T_{ts}}{\partial \phi}$ has full row rank (surjective locally), there is no non-zero vector v such that $v \frac{\partial T_{ts}}{\partial \phi} = 0$. Thus, $\nabla_{\phi} \mathcal{L}_{joint} \neq 0$, completing the proof.

[Connection to Eq. (3)] Eq. (3) states that under pure textual supervision, the time series gradient diminishes once the model reaches a text-sufficient regime. Theorem D.2 shows that adding $\lambda \mathcal{L}_{ts}$ prevents the time series module from becoming gradient-invisible in such regimes unless reconstruction is already perfect, thereby mitigating unimodal bias.

E EvalTS Implementation Details

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 ten types of temporal reasoning tasks, as detailed in section 5.

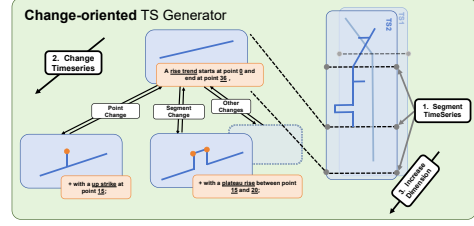


Figure 11: Time Series Generator pipeline.

E.1 Implementation

To systematically construct multimodal datasets that align temporal dynamics with textual annotations, we propose **ChronGen**, a *CHange-awaRe rules-OrieNted time series GENerator*. As in Figure 11, 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.

ChronGen-generated training examples cover the Atomic, Molecular, and Compositional levels in a 5:3:2 ratio.

E.2 QA Template

In this section, we provide illustrative examples of the data generation procedure described in subsection E.1. 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.

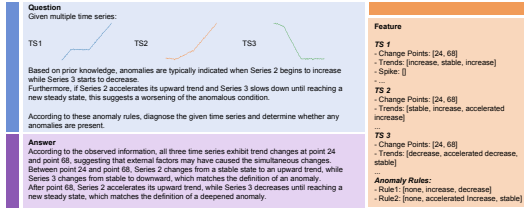


Figure 12: Examples of training datasets

E.3 Data Generation 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.

E.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.

In a manual audit of 500 randomly sampled responses, the extracted fields agreed with human extraction in 499 cases. In the sole disagreement, “point 50–55” was extracted as point 50, while the reference answer was point 18. Both interpretations were therefore incorrect, and the final score was unchanged.

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 $\mathbf{F}$

- 1: Initialize empty sequence $\mathbf{X} \leftarrow []$, feature array $\mathbf{F} \leftarrow []$, annotation set $\mathbf{Y} \leftarrow []$
- 2: Randomly sample $K - 1$ change points within $[1, L]$
- 3: Split timeline into segments $\{S_1, S_2, \dots, S_K\}$

- 4: **for** $i = 1$ to K **do**
- 5: Generate base trend T_i
- 6: Initialize segment sequence $S_i \leftarrow []$
- 7: Initialize segment annotation $Y_i \leftarrow []$
- 8: Iteratively construct variation ΔT_i :
- 9: **while** more changes to apply **do**
- 10: Sample a change operation $\text{change} \in \{\text{point_change}, \text{segment_change}, \dots\}$
- 11: Update segment: $S_i \leftarrow S_i + \text{change}(S_i, T_{i-1})$
- 12: Record feature: $\mathbf{F} \leftarrow \mathbf{F} \cup \{\text{change}\}$
- 13: Generate textual annotation: $y \leftarrow \text{text_template}(\text{change})$
- 14: Append y to Y_i
- 15: **end while**
- 16: Append segment S_i to $\mathbf{X}$
- 17: Append segment annotations Y_i to $\mathbf{Y}$
- 18: **end for**
- 19: **return** $\mathbf{X}$, $\mathbf{Y}$, and feature array $\mathbf{F}$

F Training Details

Both stages employed identical training hyperparameters except training steps. 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 6. The training loss curves and reconstruction visualizations are presented in Figure 14, which demonstrates that the two-stage training enables the

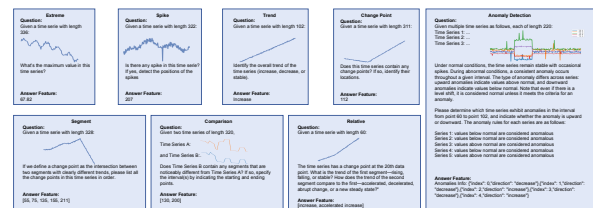


Figure 13: EvalTS examples

model to better incorporate the time series modality during training and successfully reconstruct the input time series, respectively.

Table 8 reports measured cost on 300 samples. TimeSense is slower than ChatTS because it carries reconstruction supervision during training, but it remains far faster than text-only numerical serialization with Qwen2.5 at inference.

Table 6: Training hyperparameters used in warm-up stage and joint finetuning stage.

Hyperparameter	Warm-Up Finetuning	Joint Finetuning
Per-device train batch size	1	1
Gradient accumulation steps	32	32
Maximum training steps	300	1200
Learning rate	1e-5	1e-5
Warmup ratio	0.02	0.02
GPUs	8 × A800	8 × A800

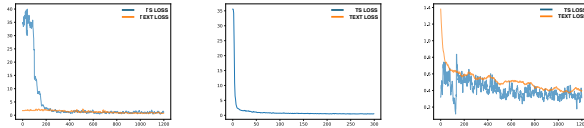


Figure 14: Training loss curves for single-stage training (left) and two-stage training (right).

G Model Output Examples

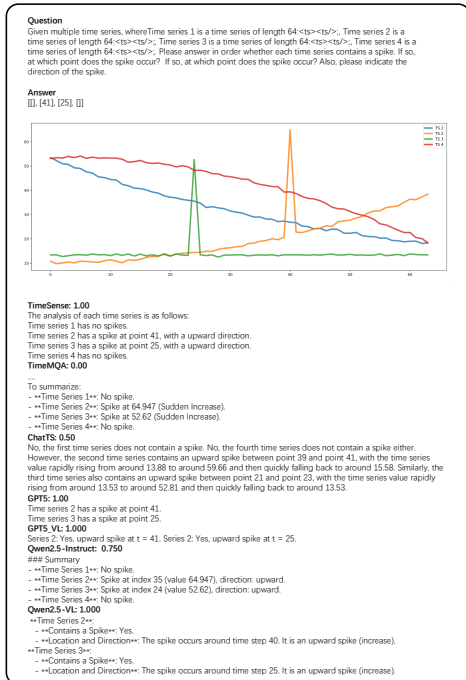


Figure 15: Model output examples on EvalTS benchmark: Spike Task.

Table 7: Statistics of the training data used for TimeSense.

Source	ChronGen	ChatTS Align	ChatTS SFT	ChatTS IFT	ChatTime	Total
# Samples	100,000	5,000	5,000	5,000	5,000	120,000

Table 8: Training and inference overhead.

Model	Train Params	Train Time	Infer Params	Infer Time
TimeSense	14.981B	1d 6h 38m	14.876B	16m 34s
ChatTS	14.864B	1d 2h 51m	14.864B	14m 56s
Qwen2.5	14.2B	—	14.167B	261m 40s
GPT5	—	—	—	134m 50s

H More Ablation Studies

Table 9 reports the robustness of TimeSense under increasing sequence lengths. Performance remains stable within the training regime up to length 512, but deteriorates noticeably beyond that range. In particular, segment accuracy drops from 0.821 at length 512 to 0.674 and 0.533 at lengths 1024 and 2048, respectively.

Table 10 illustrates the influence of patch length on different temporal reasoning tasks. Smaller patches preserve fine-grained local dynamics and therefore achieve strong performance on spike and segment localization. However, excessively small patches weaken relational understanding, as reflected by the poor Relative score at patch length 4. In contrast, larger patches provide broader temporal context but significantly impair local event localization, with spike and segment scores collapsing at patch length 32. These results reveal a clear precision–context trade-off governed by the patch granularity.

Table 11 studies the effect of the reconstruction weight λ during Stage-II training. Introducing reconstruction supervision consistently improves temporal preservation compared with very small weights, but the relationship is not monotonic. Moderate values around $\lambda = 1.0$ achieve the best balance between spike and segment reasoning. When the weight is too small, temporal structure is insufficiently preserved, whereas overly large values begin to interfere with instruction tuning and reduce downstream robustness.

Table 9: Length stress evaluation. Segment accuracy declines as sequence length exceeds the training range.

Length	24	64	128	256	512	1024	2048
Segment Acc.	0.959	0.987	0.904	0.899	0.821	0.674	0.533

LLM Parser Prompt
Model response: {model response}
Convert the model response into one JSON object that answers the time-series task.
Preserve every reported index, value, direction, trend label, and list order.
Return JSON only. Do not add an explanation.
Expected answer schema: {schema}

Figure 16: Parser prompt.

Table 10: Effect of patch length on downstream tasks.

Patch	Spike	Segment	Relative	AD
4	0.970	0.904	0.437	0.653
8	0.960	0.940	0.856	0.718
16	0.823	0.890	0.554	0.733
32	0.320	0.199	0.567	0.703

Table 11: Sensitivity to the Stage-II reconstruction weight λ .

λ	Spike	Segment
0.1	0.733	0.615
0.5	0.803	0.908
1.0	0.960	0.940
2.0	0.983	0.866
5.0	0.910	0.843