

From Consistency to Complementarity: Aligned and Disentangled Multi-modal Learning for Time Series Understanding and Reasoning

Hang Ni¹, Weijia Zhang¹, Fei Wang², Zezhi Shao², Hao Liu¹✉

¹AI Thrust, The Hong Kong University of Science and Technology (Guangzhou)

²State Key Laboratory of AI Safety, Institute of Computing Technology,
Chinese Academy of Sciences

{hni017, wzhang411}@connect.hkust-gz.edu.cn

{wangfei, shaozezhi}@ict.ac.cn, liuh@ust.hk

Abstract

Advances in multi-modal large language models (MLLMs) have inspired time series understanding and reasoning tasks. Recent attempts hybridize numerical time series with their visualized plots, facilitating precise value reasoning and visual structure comprehension for comprehensive time series understanding of MLLMs. However, effective numerical-visual modality integration remains challenging due to fine-grained temporal misalignment across modalities and severe entanglement between shared and modality-specific semantics, which hinder localized interpretation and complementary reasoning. To address these issues, we propose **MADI**, a **m**ulti-modal LLM enhanced with fine-grained **a**lignment and **d**isentangled **i**nteraction, featuring (1) Patch-level Alignment, which enforces physically grounded fine-grained correspondence across heterogeneous modalities, (2) Discrete Disentangled Interaction, which separates modality-common semantics into compact discrete latents and adaptively synergizes the purified modality-unique information, and (3) Critical-token Highlighting, which emphasizes informative, query-relevant signals for robust reasoning. Experiments on synthetic and real-world benchmarks show that MADI consistently outperforms general-purpose LLMs and time-series-specialized MLLMs. The code is available at <https://github.com/usail-hkust/MADI>.

1 Introduction

Time series understanding and reasoning (TSUR) aim to interpret natural language queries alongside time series inputs, producing textual explanations or analyses of temporal dynamics (Xie et al., 2025). Unlike classical time series tasks (e.g., forecasting, classification, anomaly detection) (Wang et al., 2026) which operate on unimodal numerical inputs and yield task-specific outputs with limited flexibility and interpretability,

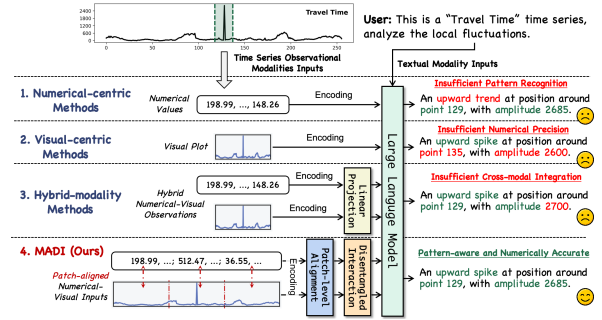


Figure 1: Comparison of multi-modal architectures.

TSUR supports interpretable, language-driven interactions between time series data and users via open-ended natural language dialogue. Such capabilities are crucial for human decision-making in real-world scenarios, including healthcare (Lan et al., 2025), finance (Chen et al., 2025), and manufacturing (Wang et al., 2025b).

With the rapid advancement of large language models (LLMs), TSUR tasks have become dominated by LLM-based paradigms, enabling flexible user queries and textual outputs. Beyond linguistic inputs, existing works primarily focus on *how to formulate time series into effective observational modalities* that can be jointly processed with language by LLMs. *Numerical-centric methods* directly feed raw time series values into LLMs, either by encoding them as plain text via the LLM tokenizer and embeddings (Liu et al., 2025b; Merrill et al., 2024), or by introducing specialized time series encoders as an additional modality (Wang et al., 2025a,b; Xie et al., 2025; Li et al., 2025b). While preserving precision, studies indicate that LLMs struggle to recognize high-level temporal structures like trends and periodicity (Liu et al., 2025b), reflecting a fundamental numerical-language modality gap induced by text-only LLM pre-training (Tan et al., 2024). In contrast, *visual-centric methods* convert time series into visual plots and leverage the strong visual processing abilities of multi-

modal LLMs (MLLMs) (Liu et al., 2025b; Merrill et al., 2024; Zhang et al., 2025a). Although effective at capturing high-level temporal structures, these approaches often sacrifice fine-grained numerical details that are critical for precise temporal analysis (Xu et al., 2026; Zhao et al., 2025). Recent hybrid approaches like GEM (Lan et al., 2025) attempt to combine numerical and visual modalities within MLLMs via token concatenation for comprehensive time series understanding. Despite these advances, effectively integrating these two observational modalities for MLLMs reveals two critical challenges:

(1) Fine-grained Cross-modal Alignment. Numerical values and visual plots describe the same physical signal, but they are usually encoded independently, leading to representational misalignment that hinders cross-modal reasoning. This problem is further amplified by the vision-language pre-training gap in MLLMs, making numerical interpretation less reliable than visual understanding. Existing approaches either rely on implicit alignment through linear projections (Lan et al., 2025), which lacks explicit physical correspondence, or perform series-level contrastive alignment (Dong et al., 2025), which overlooks local temporal variations such as fluctuations and noise. Therefore, a key challenge is to establish physically grounded, fine-grained alignment between numerical and visual modalities, so as to provide a consistent semantic reference for cross-modal understanding.

(2) Disentangled Cross-modal Interaction. While alignment improves cross-modal consistency, it alone cannot fully exploit the complementary strengths of visual patterns and numerical details. Existing fusion strategies, including concatenation (Lan et al., 2025), additive fusion (Zhong et al., 2025), and cross-attention (Liu et al., 2025a), directly operate on full-modal representations. However, since numerical and visual modalities originate from the same signal, their representations contain substantial shared information mixed with modality-specific semantics. Such entanglement may cause redundant shared content to dominate the interaction, weakening the contribution of modality-unique cues. Thus, another challenge is to disentangle shared and modality-specific semantics after alignment, and integrate complementary signals for synergistic reasoning rather than simple feature aggregation.

To address these challenges, we propose **MADI**, a multi-modal LLM (MLLM) for TSUR

tasks, enhanced with fine-grained alignment and disentangled interaction across heterogeneous modalities. The distinctions between MADI and prior methods are highlighted in Figure 1. Specifically, we introduce a *Patch-level Alignment (PA)* module to perform fine-grained contrastive alignment between numerical time series and their visualized line plots at the patch level, further enriched with patch-wise textual captions derived from the original values. This grounds numerical representations in an MLLM-compatible vision-language semantic space, enabling more reliable cross-modal reasoning. Building on the aligned representations, we devise a *Discrete Disentangled Interaction (DDI)* module to exploit the complementary strengths of numerical precision and visual abstraction. DDI employs hierarchical vector quantization to construct a compact discrete latent space that captures multi-scale modality-common features, thereby separating shared redundancies from modality-unique semantics. The resulting unique components are then fused through cross-attention for synergistic interaction with reduced interference. Finally, we introduce a lightweight *Critical-token Highlighting (CTH)* module that identifies question-relevant informative tokens across modalities and prepends them to the token sequences, improving generalization. By incorporating these modules into a pre-trained MLLM, MADI enables interpretable and flexible TSUR across diverse domains and query types, consistently outperforming both general-purpose LLMs and time-series-specialized baselines.

In summary, our contributions are threefold: (1) We propose PA to enforce fine-grained physical consistency between numerical values and visual plots, preventing localized MLLM hallucinations. (2) We introduce DDI to filter shared redundancies and leverage the complementarities of multi-modal signals for robust reasoning. (3) We conduct extensive experiments on synthetic and real-world data demonstrating the advances of MADI in TSUR.

2 Related Work

Multi-modal Classical Time Series Analysis. Recent studies augment time series with auxiliary modalities, such as text or images, to improve classical analysis, especially forecasting. One line of work combines numerical series with textual descriptions or exogenous contexts (Liu et al., 2025a; Zhang et al., 2025b), often using dual-branch ar-

architectures that separately encode each modality before fusion (Liu et al., 2024; Zhou et al., 2025; Liu et al., 2025a; Ding et al., 2025; Jiang et al., 2025). With the rise of LLMs, recent methods further enable joint modeling of textual and numerical inputs, either by tokenizing numerical values (Gruver et al., 2023; Wang et al., 2025a) or by aligning time series representations with LLM-compatible embeddings through contrastive learning (Zhang et al., 2025b), cross-attention (Jin et al., 2024; Liu et al., 2025c), or graph neural networks (Hu et al., 2025). Another line incorporates visual modalities from time series plots or auxiliary sources. For example, Time-VLM (Zhong et al., 2025) converts series into heatmaps and textual summaries for forecasting, while PIPE (Li et al., 2025a) uses satellite imagery as physical cues for typhoon intensity prediction. However, these methods are mostly task-specific and constrained by fixed output formats, limiting their applicability to flexible reasoning scenarios such as TSUR.

LLM-based TSUR. Beyond classical analysis, TSUR focuses on interpretable analysis of temporal dynamics under flexible natural language queries. Existing methods can be roughly grouped into numerical-centric, visual-centric, and hybrid paradigms. Numerical-centric methods represent time series as textual or time-series-specific tokens (Merrill et al., 2024; Wang et al., 2025a,b; Xie et al., 2025), preserving numerical precision but often struggling to capture high-level temporal structures due to the text-centric pre-training of LLMs (Tan et al., 2024; Liu et al., 2025b). Visual-centric methods (Liu et al., 2025b; Merrill et al., 2024; Zhang et al., 2025a) transform time series into plots and cast analysis as visual question answering, enabling MLLMs to capture trend and periodic patterns, but often losing fine-grained numerical details (Xu et al., 2026; Zhao et al., 2025). Hybrid methods, such as GEM (Lan et al., 2025), combine both modalities, but typically rely on simple concatenation and remain domain-specific, such as ECG analysis, limiting principled cross-modal alignment and interaction. Our work addresses these limitations by developing a generalizable MLLM that integrates numerical precision with visual abstraction through fine-grained alignment and disentangled cross-modal interaction, supporting diverse domains and flexible query types.

3 Preliminary

Problem Statement. TSUR aims to analyze time series based on a natural language query and generate a textual answer. Formally, given a set of time series $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{|\mathcal{X}|}\}$, where each $\mathbf{x}_i = (x_{i;1}, x_{i;2}, \dots, x_{i;T_i}) \in \mathbb{R}^{T_i}$ consists of T_i observations, along with textual inputs including a context C and a question Q , the objective is to learn a mapping $f : (\mathcal{X}, C, Q) \rightarrow A$, producing an answer A that explains or reasons about the relevant information in $\mathcal{X}$ with respect to Q .

Vector Quantization (VQ). VQ maps continuous features to a finite set of discrete prototypes, enabling compact, structured representations (van den Oord et al., 2017). Given input features $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_N]^\top \in \mathbb{R}^{N \times D}$ and a codebook $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^K \subseteq \mathbb{R}^D$, each feature is quantized via nearest-neighbor assignment: $\mathbf{q}_j = \arg \min_{\mathbf{c}_k \in \mathcal{C}} \|\mathbf{e}_j - \mathbf{c}_k\|_2^2$. To enable gradient-based optimization, VQ employs a straight-through estimator (STE) together with a commitment loss, resulting in the objective $\mathcal{L}_{\text{vq}} = \|\mathbf{e}_j - \text{sg}[\mathbf{q}_j]\|_2^2$, where $\text{sg}[\cdot]$ denotes the stop-gradient operator. The codebook is commonly updated using exponential moving average (EMA). *Residual Vector Quantization (RVQ)* (Lee et al., 2022) extends VQ by sequentially quantizing residuals, increasing representation capacity. Using M codebooks $\{\mathcal{C}^{(m)}\}_{m=1}^M$, RVQ initializes $\mathbf{r}_j^{(0)} = \mathbf{e}_j$ and iteratively performs $\mathbf{q}_j^{(m)} = \arg \min_{\mathbf{c} \in \mathcal{C}^{(m)}} \|\mathbf{r}_j^{(m-1)} - \mathbf{c}\|_2^2$ and $\mathbf{r}_j^{(m)} = \mathbf{r}_j^{(m-1)} - \mathbf{q}_j^{(m)}$. The final quantized representation is $\mathbf{q}_j = \sum_{m=1}^M \mathbf{q}_j^{(m)}$.

4 Methodology

Figure 2 illustrates the overall architecture of MADI, which is built upon a pre-trained MLLM, reusing its vision encoder and LLM for TSUR. Specifically, *Patch-level Alignment (PA)* first expands each time series as patch-aligned numerical, visual and textual modalities, aligning them in a consistent embedding space. Based on aligned embeddings, *Discrete Disentangled Interaction (DDI)* separates modality-shared and unique semantics for numerical and visual inputs and synergizes their complementarities. *Critical-token Highlighting (CTH)* then identifies informative, question-relevant tokens from the PA-aligned embeddings and prepends them to the DDI-fused multi-modal sequence. The resulting multi-modal embeddings are concatenated with the textual input embeddings

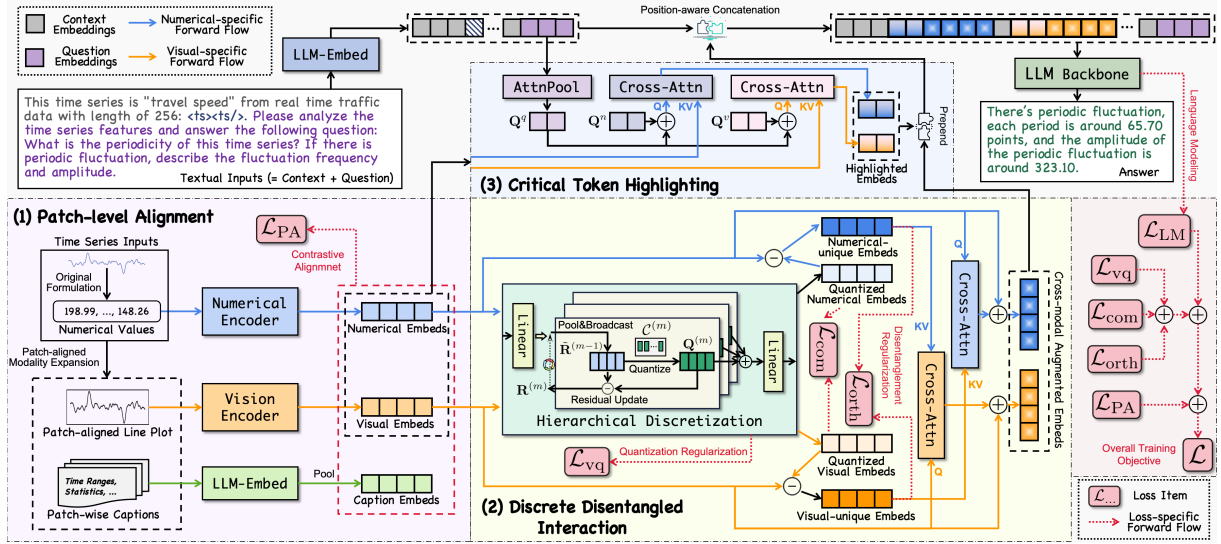


Figure 2: The model architecture of MADI.

and decoded by the LLM to generate the output.

4.1 Patch-level Alignment

To establish precise correspondence between numerical and visual modalities and enable accurate MLLM interpretation of numerical inputs, we propose a *Patch-level Alignment (PA)* module that enforces fine-grained, physically-grounded alignment among numerical, visual, and textual modalities. PA operates at a patch level and consists of three steps: (1) *patch-level modality expansion*, which constructs patch-aligned textual and visual counterparts; (2) *multi-modal encoding*, which embeds each modality independently; and (3) *contrastive alignment*, which employs patch-wise contrastive learning to reduce cross-modal representation gaps.

4.1.1 Patch-level Modality Expansion

Following (Nie et al., 2023), each time series $\mathbf{x}_i$ can be divided into non-overlapping patches to enable efficient and robust temporal modeling: $\mathbf{P}_i^n = \text{Patching}(\mathbf{x}_i) \in \mathbb{R}^{\tilde{T}_i \times p^n}$, where $\tilde{T}_i = \lceil T_i/p^n \rceil$ denotes the number of patches, and p^n is the patch size of time series. Based on such a patch-level granularity, the numerical modality is then expanded with textual and visual modalities.

Patch-aligned Visualization. We render each $\mathbf{x}_i$ as a separate line plot with adaptive resolution $\tilde{T}_i \cdot p^v \times p^v$ (width $\times$ height) where p^v is the patch size of the vision encoder in the backbone MLLM. We intentionally omit all decorative plotting elements (e.g., titles, axes, ticks, labels, and legends), thereby the resulting image $\mathbf{I}_i$ can be patchified into $\mathbf{P}_i^v \in \mathbb{R}^{\tilde{T}_i \times p^v \times p^v}$, ensuring a one-to-one correspon-

dence with numerical patches $\mathbf{P}_i^n$, and enabling fine-grained numerical-visual alignment.

Patch-wise Captioning. For numerical-text alignment, prior studies rely on time-series-agnostic texts, e.g., task prompts (Hu et al., 2025), external contexts (Zhang et al., 2025b), general word embeddings (Jin et al., 2024), which fail to capture nuanced physical variations of time series (e.g., amplitude, frequency), restricting semantic fidelity. We instead construct patch-wise captions grounded in numerical values. For each numerical patch $\mathbf{p}_{i,j}^n$, a structured caption $s_{i,j}$ consists of timestamp ranges and value statistics (e.g., max, min, mean, standard deviation), yielding a set of patch-wise captions $S_i = \{s_{i,1}, \dots, s_{i,\tilde{T}_i}\}$.

4.1.2 Multi-modal Encoding

Given expanded multi-modal inputs $\{\mathbf{x}_i, \mathbf{I}_i, S_i\}_{i=1}^{|\mathcal{X}|}$, three modalities are encoded independently into continuous patch-level embeddings. Each time series is processed independently to accommodate variable sequence lengths, while cross-series dependencies are captured by the backbone LLM.

Numerical Encoding. We adopt a lightweight time series encoder consisting of linear projection, positional embedding, and transformer blocks:

$$\begin{aligned} \tilde{\mathbf{E}}_i^n &= \text{Concat}(\mathbf{P}_i^n, \text{PosEmbed}(\mathbf{P}_i^n)), \\ \mathbf{E}_i^n &= \text{Transformer}(\text{Linear}(\tilde{\mathbf{E}}_i^n)). \end{aligned} \quad (1)$$

Visual Encoding. The time series line plot is processed by the pre-trained vision encoder of the backbone MLLM: $\mathbf{E}_i^v = [\mathbf{e}_{i,1}^v, \dots, \mathbf{e}_{i,\tilde{T}_i}^v]^\top = \text{VisionEncoder}(\mathbf{I}_i) \in \mathbb{R}^{\tilde{T}_i \times D}$.

Caption Encoding. To encode the patch-wise caption $s_{i,j}$, we reuse the tokenizer and the input embedding layer of the backbone LLM. Then a mean pooling layer is applied to obtain patch-specific caption embeddings. This process is formulated as $\mathbf{e}_{i,j}^s = \text{MeanPool}(\text{LLM-Embed}(s_{i,j}))$, stacked as $\mathbf{E}_i^s = [\mathbf{e}_{i,1}^s, \dots, \mathbf{e}_{i,\tilde{T}_i}^s]^\top \in \mathbb{R}^{\tilde{T}_i \times D}$.

4.1.3 Contrastive Alignment

To strengthen semantic correspondence across modalities, we adopt a patch-wise contrastive learning objective. Numerical patch embeddings serve as anchors, while patch-aligned visual and textual caption embeddings are positives; all other patches within the same instance act as negatives. The *numerical-visual* alignment loss is defined with an InfoNCE objective (Oord et al., 2018): $\mathcal{L}_{\text{align}}^{n-v} = -\sum_{i,j} \log \frac{\exp(\text{sim}(\mathbf{e}_{i,j}^n, \text{sg}(\mathbf{e}_{i,j}^v))/\tau)}{\sum_{j'} \exp(\text{sim}(\mathbf{e}_{i,j}^n, \text{sg}(\mathbf{e}_{i,j'}^v))/\tau)}$, where sim denotes cosine similarity, sg denotes the stop-gradient operator, τ is a temperature parameter, and (i, j') indexes all visual tokens within the i -th time series instance. The *numerical-caption* alignment loss $\mathcal{L}_{\text{align}}^{n-s}$ is computed analogously. We stop the gradients of visual and textual embeddings to optimize the numerical encoder only. The overall alignment objective is: $\mathcal{L}_{\text{PA}} = \mathcal{L}_{\text{align}}^{n-v} + \mathcal{L}_{\text{align}}^{n-s}$.

4.2 Discrete Disentangled Interaction

Building upon the aligned features, effective TSUR requires consolidating complementary semantics from numerical and visual modalities. We propose a *Discrete Disentangled Interaction (DDI)* module that estimates compact, modality-shared components in a discrete latent space, and uses the remaining residual signals as complementary cross-modal evidence for interaction. DDI consists of two stages: (1) *discrete modality disentangle-ment*, which extracts compact modality-common semantics via hierarchical vector quantization and derives modality-unique signals by residual decomposition; and (2) *unique-centric interaction*, which synergizes the unique signals via cross-attention.

4.2.1 Discrete Modality Disentanglement

A straightforward approach to identify modality-unique semantics is to apply disentanglement techniques (Qian et al., 2026), which use separate projection heads to extract modality-common and modality-specific representations, alongside regularization encouraging high correlation among common components and low correlation with unique

ones. However, existing methods (Xia et al., 2023; Huang et al., 2025) operate in continuous embedding spaces, which are unbounded and prone to capturing redundant or noisy correlations. In contrast, we perform disentanglement in a *discrete latent space* using vector quantization (VQ), which enforces compactness and consistency for modality-common semantics. Given numerical/visual embeddings $\mathbf{E}_i^n, \mathbf{E}_i^v \in \mathbb{R}^{\tilde{T}_i \times D}$, we employ a shared hierarchical discretization block to extract modality-common representations $\mathbf{Z}_i^n$ and $\mathbf{Z}_i^v$. The remaining complementary cues are then modeled by residual decomposition: $\mathbf{U}_i^o = \mathbf{E}_i^o - \mathbf{Z}_i^o$, $o \in \{n, v\}$.

Hierarchical Discretization. To capture the multi-scale time series semantics, we propose a hierarchical RVQ strategy featured by M hierarchical codebooks $\{\mathcal{C}^{(m)}\}_{m=1}^M$, where the m -th codebook $\mathcal{C}^{(m)}$ contains $K \cdot 2^{m-1}$ codes and operates at a temporal resolution downsampled by 2^{M-m} tokens. Specifically, we first follow (Yu et al., 2022) and project continuous embeddings $\mathbf{E}_i^n \in \mathbb{R}^{\tilde{T}_i \times D}$ to a lower-dimensional space via $\tilde{\mathbf{E}}_i^n = \text{Linear}(\mathbf{E}_i^n) \in \mathbb{R}^{\tilde{T}_i \times d}$, which increases the density of codes and reduces the quantization error. At each quantization stage m , given the residual token sequence $\mathbf{R}_i^{n;(m-1)} = [\mathbf{r}_{i,1}^{n;(m-1)}, \dots, \mathbf{r}_{i,\tilde{T}_i}^{n;(m-1)}]^\top \in \mathbb{R}^{\tilde{T}_i \times d}$ from the last round, we partition it into non-overlapping segments of length $W = 2^{M-m}$, and compute segment-level representations by average pooling over the tokens within each segment. These pooled representations are then broadcasted back to all token positions in the corresponding segment. This process can be formulated as $\tilde{\mathbf{r}}_{i,j}^{n;(m-1)} = \frac{1}{W} \sum_{b=j^*}^{j^*+W} \mathbf{r}_{i,b}^{n;(m-1)}$, where $j^* = [(j-1)/W]$ denotes the segment index of the j -th token. Afterwards, we follow the steps illustrated in Section 3, iteratively quantizing the residual features, from coarse to fine scales:

$$\begin{aligned} \mathbf{q}_{i,j}^{n;(m)} &= \arg \min_{\mathbf{c} \in \mathcal{C}^{(m)}} \|\tilde{\mathbf{r}}_{i,j}^{n;(m-1)} - \mathbf{c}\|_2^2, \\ \mathbf{r}_{i,j}^{n;(m)} &= \mathbf{r}_{i,j}^{n;(m-1)} - \mathbf{q}_{i,j}^{n;(m)}, \end{aligned} \quad (2)$$

where the initial residual token embeddings are $\mathbf{R}_i^{n;(0)} = \tilde{\mathbf{E}}_i^n$. The final discretized features are accumulated by $\mathbf{q}_{i,j}^n = \sum_{m=1}^M \mathbf{q}_{i,j}^{n;(m)}$. After quantization, the discrete representations should be projected back to D -dimensional space via $\mathbf{z}_{i,j}^n = \text{Linear}(\mathbf{q}_{i,j}^n) \in \mathbb{R}^D$, which serve as the modality-common tokens, constituting $\mathbf{Z}_i^n \in \mathbb{R}^{\tilde{T}_i \times D}$. The modality-common visual embeddings $\mathbf{Z}_i^v$ are de-

rived in a similar way. The quantization process is supervised the $\mathcal{L}_{\text{vq}}$ illustrated in Section 3.

Disentanglement Regularization. To decouple modality-common and unique information, we introduce two regularization terms. First, to align modality-common semantics, we apply a contrastive objective between quantized numerical and visual embeddings: $\mathcal{L}_{\text{com}} = \frac{1}{2}(\mathcal{L}_{\text{com}}^{n-v} + \mathcal{L}_{\text{com}}^{v-n})$, where $\mathcal{L}_{\text{com}}^{n-v} = -\sum_{i,j} \log \frac{\exp(\text{sim}(\mathbf{z}_{i;j}^n, \mathbf{z}_{i;j}^v)/\tau)}{\sum_{j'} \exp(\text{sim}(\mathbf{z}_{i;j}^n, \mathbf{z}_{i;j'}^v)/\tau)}$. Moreover, to discourage shared semantics from being redundantly encoded in the residual component, we add a lightweight cosine orthogonality regularizer between the common and unique embeddings: $\mathcal{L}_{\text{orth}} = \frac{1}{2} \sum_{i,j} (|\text{sim}(\mathbf{z}_{i;j}^n, \mathbf{u}_{i;j}^n)| + |\text{sim}(\mathbf{z}_{i;j}^v, \mathbf{u}_{i;j}^v)|)$. The overall loss is: $\mathcal{L}_{\text{DDI}} = \mathcal{L}_{\text{vq}} + \alpha \cdot \mathcal{L}_{\text{com}} + \beta \cdot \mathcal{L}_{\text{orth}}$, where α, β control the regularization strengths.

4.2.2 Unique-centric Interaction

After disentanglement, modality-unique semantics are integrated via cross-modal attention. Each modality’s original tokens are preserved and augmented by attending to the residual complementary cues from the opposite modality: $\bar{\mathbf{E}}_i^n = \mathbf{E}_i^n + \text{CrossAttn}(\mathbf{E}_i^n, \mathbf{U}_i^v, \mathbf{U}_i^v)$, $\bar{\mathbf{E}}_i^v = \mathbf{E}_i^v + \text{CrossAttn}(\mathbf{E}_i^v, \mathbf{U}_i^n, \mathbf{U}_i^n)$. This design enables fine-grained numerical dynamics and high-level visual patterns to complement each other.

4.3 Critical-token Highlighting

Although PA and DDI produce aligned and complementary representations, directly feeding them into the LLM may dilute question-relevant or inherently salient information, thereby limiting generalization. To address this, we propose a *Critical-token Highlighting (CTH)* module, which preserves all original tokens while prepending a few informative tokens to explicitly emphasize key signals. These tokens are selected based on both question-aware and modality-intrinsic saliency. Specifically, the question Q is tokenized and embedded as $\mathbf{E}^q = \text{LLM-Embed}(Q) \in \mathbb{R}^{N^q \times D}$. Given $\mathbf{E}^q$ and the continuous numerical/visual embeddings $\mathbf{E}_i^o \in \mathbb{R}^{\tilde{T}_i \times D}$ ($o \in n, v$), CTH adopts two parallel cross-attention branches.

Question-conditioned Branch. We first compress the variable-length question embeddings into H learnable query embeddings via an attention-based pooling layer: $\tilde{\mathbf{Q}}^q = \text{CrossAttn}(\mathbf{Q}^q, \mathbf{E}^q, \mathbf{E}^q)$, and then use the compressed question-aware queries to attend over the modality tokens by $\mathbf{H}_i^{o;q} = \text{CrossAttn}(\tilde{\mathbf{Q}}^q, \mathbf{E}_i^o, \mathbf{E}_i^o)$, $o \in \{n, v\}$.

Modality-intrinsic Branch. In parallel, we introduce another set of H learnable queries $\mathbf{Q}^o \in \mathbb{R}^{H \times D}$ to identify modality-intrinsic patterns: $\mathbf{H}_i^{o;s} = \text{CrossAttn}(\mathbf{Q}^o, \mathbf{E}_i^o, \mathbf{E}_i^o)$.

The highlighted token embeddings are fused as $\mathbf{H}_i^o = \mathbf{H}_i^{o;q} + \mathbf{H}_i^{o;s}$ and prepended to the modality token sequence via $\hat{\mathbf{E}}_i^o = \text{Concat}(\mathbf{H}_i^o, \bar{\mathbf{E}}_i^o) \in \mathbb{R}^{(H+\tilde{T}_i) \times D}$, $o \in \{n, v\}$. This results in holistic and question-aware modality representations for LLM understanding and reasoning.

4.4 LLM Backbone and Training Objective

The enhanced numerical and visual embeddings $\{\hat{\mathbf{E}}_i^n, \hat{\mathbf{E}}_i^v\}_{i=1}^{|\mathcal{X}|}$ are concatenated with textual input tokens, including the question embeddings $\mathbf{E}^q$ and context embeddings $\mathbf{E}^c$ (derived from the context C using the LLM’s embeddings), via a position-aware concatenation operator (Xie et al., 2025): $\mathbf{E}^* = \text{PosConcat}(\mathbf{E}^c, \mathbf{E}^q, \{\hat{\mathbf{E}}_i^n, \hat{\mathbf{E}}_i^v\}_{i=1}^{|\mathcal{X}|})$, where $\text{PosConcat}(\cdot)$ integrates multi-modal embeddings in the textual inputs according to the positions of the time series instances in the original input. The fused sequence $\mathbf{E}^*$ is fed into the pre-trained backbone LLM to generate the answer A autoregressively: $P(A \mid \mathbf{E}^*) = \prod_{b=1}^{|A|} P(A_b \mid A_{<b}, \mathbf{E}^*)$, where A_b denotes the b -th token of the target answer sequence. For model training, the language modeling loss is: $\mathcal{L}_{\text{LM}} = -\sum_{b=1}^{|A|} \log P(A_b \mid A_{<b}, \mathbf{E}^*)$, and the overall training objective incorporates the regularization terms from PA and DDI: $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_1 \cdot \mathcal{L}_{\text{PA}} + \lambda_2 \cdot \mathcal{L}_{\text{DDI}}$, where λ_1, λ_2 denote the loss weights.

5 Experiment

5.1 Experimental Setups

Dataset. We use the datasets released with ChatTS (Xie et al., 2025), which contains synthetic language-time series pairs for training and a hybrid synthetic/real-world evaluation set. The evaluation includes two task types: (1) *Understanding tasks*, which examine basic properties of uni- or multi-variate time series, such as noise, local fluctuations, seasonality, trends, correlations, and clustering patterns; and (2) *Reasoning tasks*, which cover inductive, deductive, causal, and comparative reasoning. Understanding tasks involve categorical or numerical answers, while reasoning tasks are mainly categorical, with some open-ended inductive questions. More details are provided in Appendix B.1. We further evaluate cross-domain generalization on an independent benchmark in Appendix C.

Table 1: Performance (%) on understanding tasks. The light-blue background denotes results from general-purpose LLMs/MLLMs accessed via APIs, while the light-red background denotes results from time-series-specific MLLMs.

Dataset	Type	Model	Noise		Local		Season		Trend		Corr.	Clus.	Overall	
			Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Num.	Cate.	Cate.	Cate.	Num.
Synthetic	Numerical	DeepSeek-V3.2	74.39	47.15	31.70	23.13	78.35	67.16	78.57	85.04	36.64	30.20	43.81	37.74
		GPT-4o	56.10	43.21	14.77	9.72	88.66	55.16	72.62	83.80	34.81	30.09	38.36	26.81
		Qwen3	63.41	40.64	24.23	15.69	85.57	51.25	66.67	83.48	30.11	29.36	39.01	30.67
		GPT-5.2	73.17	43.03	41.41	34.88	96.91	72.09	78.57	93.01	47.20	37.27	52.27	47.88
		Gemini 3 Pro	84.15	59.67	53.12	45.82	95.88	84.10	76.19	83.26	54.12	49.16	60.48	55.63
		ChatTime	84.15	13.74	67.14	35.49	95.88	49.08	91.67	12.65	71.95	69.84	74.94	32.21
		ChatTS	93.90	47.20	<u>84.38</u>	<u>76.02</u>	100.00	86.64	96.43	<u>96.34</u>	81.07	<u>79.27</u>	<u>85.32</u>	<u>79.54</u>
		ITFormer	85.37	23.32	20.69	13.71	71.13	26.92	70.24	94.19	56.66	47.50	50.32	28.59
		InstructTime	79.27	19.34	22.32	15.24	70.10	11.23	72.62	96.27	52.29	48.06	49.34	28.60
	Visual	GPT-4o	64.63	31.72	34.39	26.97	87.63	65.01	50.00	48.23	39.54	43.90	46.59	33.79
		Qwen3-VL	53.66	18.92	34.37	26.31	90.72	49.71	80.95	49.08	38.81	41.70	47.71	31.92
		GPT-5.2	75.61	66.64	52.28	44.12	96.91	74.54	78.57	84.61	61.03	52.49	62.55	54.07
		Gemini 3 Pro	91.46	60.52	53.84	46.49	94.85	<u>89.58</u>	80.95	79.89	65.39	55.38	66.01	56.06
	Num.+Visual	GPT-4o	58.54	40.02	24.60	19.20	93.81	69.00	46.43	82.19	33.72	37.38	40.89	34.41
		Qwen3-VL	48.78	27.21	23.41	18.34	86.60	54.67	83.33	70.71	36.64	36.91	42.73	30.40
		GPT-5.2	73.17	61.05	43.25	36.42	98.97	79.38	75.00	93.61	59.21	48.32	58.58	50.27
		Gemini 3 Pro	86.59	<u>70.51</u>	53.92	46.88	97.94	86.52	76.19	83.88	63.57	54.72	64.90	56.96
		GEM	<u>96.34</u>	56.25	80.81	70.08	100.00	85.01	91.67	95.77	<u>82.15</u>	78.45	84.36	75.27
		MADI	98.78	78.39	91.45	84.73	100.00	93.27	<u>94.05</u>	96.60	86.89	84.35	89.99	87.22
Real-world	Numerical	DeepSeek-V3.2	85.71	29.30	60.72	46.14	81.08	0.00	65.85	88.09	61.11	51.73	63.86	50.35
		GPT-4o	95.24	19.97	32.15	24.16	78.38	0.00	70.73	84.49	57.03	46.12	54.80	34.86
		Qwen3	78.57	37.44	46.78	29.73	56.76	49.22	56.10	81.31	76.67	44.31	55.54	40.49
		GPT-5.2	92.86	31.88	57.86	45.06	<u>97.30</u>	0.00	60.98	96.02	81.11	56.90	69.47	52.33
		Gemini 3 Pro	92.86	24.76	86.07	72.08	91.89	70.27	56.10	81.16	84.08	76.88	80.25	67.36
		ChatTime	50.00	2.92	83.21	48.55	86.49	43.71	97.56	5.97	85.56	54.51	75.36	34.75
		ChatTS	88.10	<u>47.09</u>	86.78	78.40	<u>97.30</u>	<u>84.60</u>	<u>92.68</u>	90.01	68.52	62.94	82.39	77.01
		ITFormer	69.05	19.85	40.12	28.63	78.38	44.59	70.73	90.84	43.70	40.58	52.04	38.74
		InstructTime	54.76	11.22	32.73	24.67	<u>97.30</u>	0.00	75.61	<u>96.04</u>	52.59	30.89	51.53	36.16
	Visual	GPT-4o	97.62	20.10	64.41	53.05	94.59	0.00	34.15	68.48	55.18	62.75	64.22	50.17
		Qwen3-VL	97.62	30.93	63.81	50.41	59.46	20.54	80.49	53.75	54.45	61.80	66.97	47.60
		GPT-5.2	95.24	43.38	83.21	71.77	<u>97.30</u>	0.00	65.85	90.66	87.78	75.43	82.33	70.39
		Gemini 3 Pro	97.62	43.27	86.31	73.28	<u>97.30</u>	30.28	75.61	90.42	92.96	<u>81.23</u>	<u>86.58</u>	71.95
	Num.+Visual	GPT-4o	95.24	28.12	44.65	36.79	91.89	0.00	34.15	86.28	60.37	51.24	56.00	43.63
		Qwen3-VL	95.24	29.21	40.35	30.80	72.97	42.70	80.49	67.15	66.29	48.39	59.97	37.30
		GPT-5.2	88.10	43.86	82.98	71.40	<u>97.30</u>	0.00	65.85	96.14	87.41	68.44	79.44	71.03
		Gemini 3 Pro	97.62	46.32	86.31	72.76	<u>97.30</u>	68.75	63.41	82.95	92.96	82.47	85.18	70.99
		GEM	90.48	45.76	<u>90.48</u>	<u>80.30</u>	94.59	78.10	87.80	92.06	70.00	70.43	84.04	<u>78.29</u>
		MADI	97.62	55.95	93.81	86.30	100.00	94.42	87.80	88.48	77.78	70.14	88.54	83.78

Baselines. We compare our model with baselines covering three time-series modality formulations: (1) *numerical-centric models*, including general-purpose LLMs prompted with serialized values: GPT-4o (OpenAI, 2024), Qwen3 (Yang et al., 2025), DeepSeek-V3.2 (DeepSeek-AI, 2025), Gemini 3 Pro (Google DeepMind, 2026), GPT-5.2 (OpenAI, 2025); and time-series-specific MLLMs with dedicated encoders, such as ChatTime (Wang et al., 2025a), ChatTS (Xie et al., 2025), ITFormer (Wang et al., 2025b), and InstructTime (Cheng et al., 2025); (2) *visual-centric models*, which reason over time-series plots using general-purpose MLLMs: GPT-4o, Qwen3-VL (Bai et al., 2025a), Gemini 3 Pro, GPT-5.2; and (3) *numerical+visual models*, which jointly use serialized values and plots, including GEM (Lan et al., 2025) and general-purpose MLLMs: GPT-4o, Qwen3-VL, Gemini 3 Pro, GPT-5.2. General-purpose LLMs/MLLMs are accessed via APIs, while time-series-specialized MLLMs are fine-

tuned under unified LLM backbones and training data. We defer the detailed setup of these baselines to Appendix B.2.

Additional details of the evaluation metrics, implementation, and repeated-run analysis are provided in Appendix B.3, Appendix B.4, and Appendix F, respectively.

5.2 Comparison Results

Table 1 reports performance on understanding tasks across synthetic and real-world datasets, while Table 2 summarizes reasoning results on the hybrid dataset. We draw the following observations. (1) MADI achieves consistent gains across both understanding and reasoning tasks, demonstrating the importance of carefully designed multi-modal alignment and interaction mechanisms for TSUR. (2) Visual plots are generally more effective than numerical sequences, with visual-centric models outperforming numerical-centric ones in most settings, including GPT-4o’s up to 17% gain in cate-

Table 2: Performance (%) on reasoning tasks.

Type	Model	Induct.	Deduct.	Causal	MCQ2	Overall
Numerical	DeepSeek-V3.2	33.66	69.77	72.83	56.00	40.99
	GPT-4o	32.07	60.47	66.30	49.00	38.22
	Qwen3	32.38	48.84	68.48	46.00	37.93
	GPT-5.2	30.32	0.00	73.91	43.00	34.23
	Gemini 3 Pro	34.46	46.51	63.04	<u>68.00</u>	40.90
	ChatTime	49.18	48.84	53.26	44.00	49.07
	ChatTS	55.66	51.16	58.70	53.00	55.50
	ITFormer	42.54	44.19	47.83	46.00	43.48
	InstructTime	47.02	53.49	50.00	52.00	48.25
Visual	GPT-4o	28.86	62.79	68.48	51.00	36.03
	Qwen3-VL	33.11	51.16	60.87	55.00	38.75
	GPT-5.2	32.09	51.16	67.39	51.00	37.93
	Gemini 3 Pro	36.57	62.79	70.65	<u>68.00</u>	43.87
Num.+Visual	GPT-4o	26.81	55.81	69.57	47.00	33.93
	Qwen3-VL	29.70	51.16	69.57	54.00	36.76
	GPT-5.2	29.93	34.88	<u>75.00</u>	46.00	35.74
	Gemini 3 Pro	30.12	53.49	58.70	69.00	37.49
	GEM	58.73	<u>67.44</u>	66.30	53.00	<u>59.30</u>
	MADI	<u>57.68</u>	<u>67.44</u>	76.09	62.00	61.98

gorical accuracy on real-world understanding tasks. (3) For general-purpose LLMs, simply concatenating multi-modal inputs slightly degrades performance compared with visual-centric inputs, highlighting the need for structured and fine-grained multi-modal learning rather than naive combination. (4) General-purpose LLMs generalize better to real-world TSUR settings despite modest synthetic performance. For example, visually augmented Gemini 3 Pro outperforms several training-based MLLMs, such as ChatTS and GEM, on categorical understanding, whereas synthetic-data fine-tuned models often degrade on real-world benchmarks, particularly for cross-variable correlation and clustering queries. This suggests the inherent difficulty of robust real-world generalization in TSUR. (5) Reasoning tasks remain much harder than understanding tasks, with substantially lower performance highlighting the intrinsic difficulty of high-level time series reasoning. The computational cost analysis and case studies are provided in Appendix D and Appendix G, respectively.

5.3 Ablation Study

We evaluate the contribution of each component by comparing MADI with several ablated variants. “w/o PA”, “w/o DDI”, and “w/o CTH” remove the PA, DDI, and CTH modules, respectively. To examine fine-grained alignment and disentanglement, “w/o NVA” removes numerical-visual alignment, “w/o NCA” removes numerical-caption alignment, and “w/o MD” removes modality disentanglement. In addition, “w/o VQ” replaces hierarchical discretization in DDI with modality-shared continuous encoders, while “w/o Num.” removes the numerical modality and disables PA and DDI. As shown in Figure 3, the results on categorical understand-

ing, numerical understanding, and reasoning tasks reveal several findings. (1) Cross-modal alignment and interaction are essential, as removing PA or DDI causes substantial drops. (2) CTH improves robustness and generalization by highlighting critical question-relevant information, especially for unseen reasoning tasks. (3) Both numerical-visual and numerical-caption alignments are important, and their combined use brings synergistic benefits. (4) Modality disentanglement is pivotal, since isolating modality-specific signals facilitates cross-modal interaction. (5) VQ is crucial for effective disentanglement and interaction, as evidenced by the degradation of “w/o VQ”. (6) Numerical information improves robustness and generalization, but its benefits rely on well-designed multi-modal alignment and interaction, as shown by the gap between “w/o Num.” and the full model.

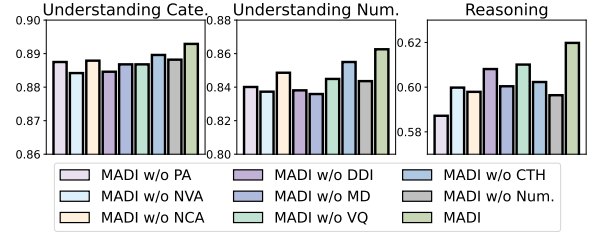


Figure 3: Ablation results of MADI.

5.4 Visualized Analysis

In this section, we empirically evaluate MADI’s ability to address the two key challenges.

Alignment Analysis. To verify fine-grained cross-modal grounding, we visualize the patch-wise cosine similarity matrices between numerical embeddings and their visual/textual counterparts in Figure 4. Compared with the ablated variant (w/o PA), MADI shows clearer diagonal high-similarity patterns, especially for numerical-visual token pairs, indicating that the PA module enforces precise patch-level cross-modal correspondence. The numerical-caption alignment is relatively weaker, likely because template-based captions introduce semantic redundancy and make patch-to-text consistency more ambiguous.

Disentanglement Analysis. To evaluate semantic decoupling in the DDI module, we analyze the KDE distributions of cross-modal cosine similarities in Figure 5. We compare original continuous embeddings (*Continuous-Continuous*), modality-common signals (*Common-Common*), and modality-unique residuals (*Unique-Unique*).

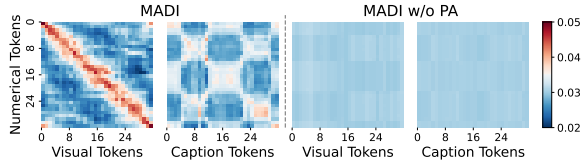


Figure 4: Heatmaps of embedding similarities.

Modality-common pairs achieve higher similarity than continuous ones, suggesting better concentration of shared semantics, while modality-unique pairs show substantially lower correlation, confirming that DDI effectively separates unique information from the shared subspace.

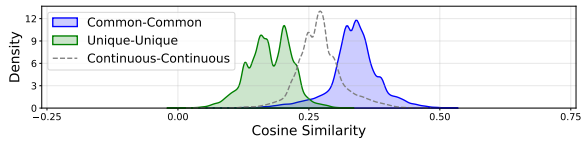


Figure 5: Distributions of embedding similarities.

6 Conclusion

We present MADI, a multi-modal LLM for TSUR tasks that improves fine-grained alignment and disentangled interaction between numerical and visual modalities. Specifically, the PA module establishes physically grounded correspondence among numerical, visual, and textual tokens; the DDI module separates modality-common and modality-specific semantics for complementary fusion; and the CTH module highlights informative, query-relevant tokens to strengthen reasoning robustness. Extensive experiments on synthetic and real-world benchmarks show that MADI consistently outperforms general-purpose LLMs and time-series-specialized MLLMs, demonstrating the effectiveness of structured multi-modal integration and its potential for interpretable and flexible time series analysis.

Limitations

Visualization Scope. This work mainly uses line plots as the visual representation of time series, since they provide an intuitive and widely used view of temporal trends, local fluctuations, and cross-variable relations. Other visualization forms, such as heat maps, may expose different temporal structures and are left for future exploration.

Task Output Scope. This work focuses on TSUR tasks that take numerical/visual time series as input and produce textual answers. We do not study

generation-style tasks (*e.g.*, forecasting) that directly output future values or new temporal sequences, as these tasks may require different objectives, uncertainty modeling, and even model families beyond LLMs.

Model Backbone Coverage. For reproducibility and fair comparison, our training-based experiments use a unified MLLM backbone and controlled training data across specialized baselines. The proposed alignment and disentangled interaction mechanisms may interact differently with larger backbones, alternative vision encoders, or future MLLM architectures, which we leave to broader follow-up studies.

Preprocessing and Input Conditions. MADI assumes that time series can be reasonably normalized, patched, and rendered into aligned visual plots. More challenging settings, such as irregular sampling, severe missingness, very long sequences, or high-dimensional multivariate streams, may require additional preprocessing or hierarchical representation designs.

Evaluation Robustness. Our evaluation follows established automatic metrics for categorical, numerical, and open-ended reasoning questions. For high-stakes domain applications, automatic scores may not fully capture factual adequacy, causal validity, or user utility, so future work can incorporate reliable expert assessment and task-specific validation protocols.

Ethical Considerations

This work uses publicly available datasets and pre-trained models, including time-series data and model checkpoints released through public repositories such as Hugging Face. We do not collect private user data, personally identifiable information, or human-subject records. The time series considered in our experiments are used for general understanding and reasoning evaluation, and the benchmark annotations follow the data sources and licenses of the original datasets.

MADI is intended as a research system for interpretable time series understanding rather than an autonomous decision maker. Although the studied data modality does not introduce direct ethical risks by itself, time-series analysis can be used in sensitive domains such as healthcare, finance, and infrastructure monitoring. Any deployment in such settings should include domain-expert validation, privacy-preserving data handling, and human over-

sight to avoid harmful consequences from model errors, hallucinated explanations, or over-reliance on automated outputs. If code, checkpoints, or processed data are released, we will preserve the usage conditions and attribution requirements of the original public resources.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62572417, No.92370204), National Key R&D Program of China (Grant No.2023YFF0725004), and Open Funding Programs of State Key Laboratory of AI Safety.

References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhi-fang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. [Qwen3-vl technical report](#). *Preprint*, arXiv:2511.21631.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.
- Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassioulas, Yifeng Gao, and Rex Ying. 2025. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*.
- Mingyue Cheng, Yiheng Chen, Qi Liu, Zhiding Liu, Yucong Luo, and Enhong Chen. 2025. [Instructime: Advancing time series classification with multimodal language modeling](#). In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM '25*, page 792–800, New York, NY, USA. Association for Computing Machinery.
- DeepSeek-AI. 2025. [Deepseek-v3.2: Pushing the frontier of open large language models](#). *Preprint*, arXiv:2512.02556.
- Kuiye Ding, Fanda Fan, Yao Wang, Ruijie Jian, Xiaorui Wang, Luqi Gong, Yishan Jiang, Chunjie Luo, and Jianfeng Zhan. 2025. [Dualsg: A dual-stream explicit semantic-guided multivariate time series forecasting framework](#). In *Proceedings of the 33rd ACM International Conference on Multimedia, MM '25*, page 508–517, New York, NY, USA. Association for Computing Machinery.
- Sixun Dong, Wei Fan, Teresa Wu, and Yanjie Fu. 2025. Teaching time series to see and speak: Forecasting with aligned visual and textual perspectives. *arXiv preprint arXiv:2506.24124*.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Google DeepMind. 2026. Gemini 3 pro model card. <https://deepmind.google/models/model-cards/gemini-3-pro>. Accessed 2026-05-25.
- Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. 2025. [Context-alignment: Activating and enhancing LLMs capabilities in time series](#). In *The Thirteenth International Conference on Learning Representations*.
- Hai Huang, Yan Xia, Shengpeng Ji, Shulei Wang, Hanting Wang, Minghui Fang, Jieming Zhu, Zhenhua Dong, Sashuai Zhou, and Zhou Zhao. 2025. [Enhancing multimodal unified representations for cross modal generalization](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2353–2366, Vienna, Austria. Association for Computational Linguistics.
- Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. 2025. [TimeXL: Explainable multi-modal time series prediction with LLM-in-the-loop](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. 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 *The Twelfth International Conference on Learning Representations*.
- Xiang Lan, Feng Wu, Kai He, Qinghao Zhao, Shenda Hong, and Mengling Feng. 2025. [GEM: Empowering MLLM for grounded ECG understanding with time series and images](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532.

- Haobo Li, Eunseo Jung, Zixin CHEN, Zhaowei Wang, Yueya WANG, Huamin Qu, and Alexis Kai Hon Lau. 2025a. [PIPE: Physics-informed position encoding for alignment of satellite images and time series in typhoon forecasting](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zechen Li, Shohreh Deldari, Linyao Chen, Hao Xue, and Flora D. Salim. 2025b. [SensorLLM: Aligning large language models with motion sensors for human activity recognition](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 354–379, Suzhou, China. Association for Computational Linguistics.
- Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. 2025a. [Timecma: towards llm-empowered multivariate time series forecasting via cross-modality alignment](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25. AAAI Press.
- Haoxin Liu, Chenghao Liu, and B. Aditya Prakash. 2025b. [A picture is worth a thousand numbers: Enabling LLMs reason about time series via visualization](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7486–7518, Albuquerque, New Mexico. Association for Computational Linguistics.
- Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. 2024. [Time-mmd: multi-domain multimodal dataset for time series analysis](#). In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS ’24, Red Hook, NY, USA. Curran Associates Inc.
- Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. 2025c. [Calf: aligning llms for time series forecasting via cross-modal fine-tuning](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25. AAAI Press.
- Mike A Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. 2024. [Language models still struggle to zero-shot reason about time series](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3512–3533, Miami, Florida, USA. Association for Computational Linguistics.
- Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. [A time series is worth 64 words: Long-term forecasting with transformers](#). In *The Eleventh International Conference on Learning Representations*.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- OpenAI. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- OpenAI. 2025. Introducing gpt-5.2. <https://openai.com/index/introducing-gpt-5-2/>. Accessed 2026-05-25.
- Chengxuan Qian, Shuo Xing, Li Li, Yue Zhao, and Zhengzhong Tu. 2026. [Decalign: Hierarchical cross-modal alignment for decoupled multimodal representation learning](#). In *The Fourteenth International Conference on Learning Representations*.
- Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024. Are language models actually useful for time series forecasting? In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS ’24, Red Hook, NY, USA. Curran Associates Inc.
- Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 6309–6318, Red Hook, NY, USA. Curran Associates Inc.
- 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 Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25. AAAI Press.
- 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.
- Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. 2026. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Yan Xia, Hai Huang, Jieming Zhu, and Zhou Zhao. 2023. Achieving cross modal generalization with multimodal unified representation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA. Curran Associates Inc.

- 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](#). *Proc. VLDB Endow.*, 18(8):2385–2398.
- Xiong Xiao Xu, Haoran Wang, Yueqing Liang, Philip S. Yu, Yue Zhao, and Kai Shu. 2026. [Can multimodal llms perform time series anomaly detection?](#) In *Proceedings of the ACM Web Conference 2026*, WWW ’26, page 5392–5403, New York, NY, USA. Association for Computing Machinery.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Fangxu Yu, Xingang Guo, Lingzhi Yuan, Haoqiang Kang, Hongyu Zhao, Lianhui Qin, Furong Huang, Bin Hu, and Tianyi Zhou. 2026. [TSRBench: A comprehensive multi-task multi-modal time series reasoning benchmark for generalist models](#). In *Forty-third International Conference on Machine Learning*.
- Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2022. [Vector-quantized image modeling with improved VQGAN](#). In *International Conference on Learning Representations*.
- Junru Zhang, Lang Feng, Xu Guo, Yuhan Wu, Yabo Dong, and Duanqing Xu. 2025a. [Timemaster: Training time-series multimodal llms to reason via reinforcement learning](#). *arXiv preprint arXiv:2506.13705*.
- Yang Zhang, Wenbo Yang, Jun Wang, Qiang Ma, and Jie Xiong. 2025b. [Camef: Causal-augmented multi-modality event-driven financial forecasting by integrating time series patterns and salient macroeconomic announcements](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD ’25, page 3867–3878, New York, NY, USA. Association for Computing Machinery.
- Ziming Zhao, ChengAo Shen, Hanghang Tong, Dongjin Song, Zhigang Deng, Qingsong Wen, and Jingchao Ni. 2025. [From images to signals: Are large vision models useful for time series analysis?](#) *arXiv preprint arXiv:2505.24030*.
- Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. 2025. [Time-vlm: exploring multimodal vision-language models for augmented time series forecasting](#). In *Proceedings of the 42nd International Conference on Machine Learning*, ICML’25. JMLR.org.
- Shiqiao Zhou, Holger Schöner, Huanbo Lyu, Edouard Fouché, and Shuo Wang. 2025. [Balm-tsf: Balanced multimodal alignment for llm-based time series forecasting](#). In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM ’25, page 4498–4508, New York, NY, USA. Association for Computing Machinery.

A Notation

The key notations used in this paper are summarized in Table 3.

B Experimental Details

B.1 Dataset

Our experiments utilize the datasets released by ChatTS (Xie et al., 2025), which are designed for time series understanding and reasoning (TSUR). The data are divided into training and evaluation sets and are sourced from both synthetic generation and real-world domains. We additionally assess generalization on an independent cross-domain benchmark in Appendix C.

B.1.1 Training Data Construction

Due to the scarcity of high-quality, well-aligned time series-text pairs, the training data are primarily constructed through a synthetic generation pipeline introduced by ChatTS (Xie et al., 2025), consisting of two key modules:

- **Attribute-Based Generation:** To ensure precise alignment between numerical values and textual descriptions, time series are generated from a predefined *Attribute Pool* covering four major categories: *Trend*, *Periodicity*, *Noise*, and *Local Fluctuations* (e.g., spikes and phase changes). An *Attribute Selector* samples attributes to construct time series arrays while simultaneously generating ground-truth textual descriptions based on the selected attributes.
- **Time Series Evol-Instruct (TSEvol):** To enhance data diversity and reasoning complexity, we adopt an evolutionary instruction generation strategy. Starting from seed question-answer pairs, an LLM progressively evolves questions into more complex forms, while querying the attribute pool to ensure factual consistency.

Table 3: Summary of Notations.

Notation	Description
General & Problem Definition	
$\mathcal{X}$	Set of time series inputs, containing instances $\mathbf{x}_i$
$\mathbf{x}_i$	The i -th time series instance
T_i	Number of observations in time series $\mathbf{x}_i$
C, Q	Context and question textual inputs
$\mathbf{E}^c, \mathbf{E}^q$	Embeddings for context and question, composed of tokens $\mathbf{e}^c, \mathbf{e}^q$
A	Generated textual answer
Patch-level Alignment (PA)	
p^n	Patch size for numerical time series
p^v	Patch size for the vision encoder
$\tilde{T}_i$	Number of patches for $\mathbf{x}_i$
$\mathbf{P}_i^n$	Numerical time series patches, composed of $\mathbf{p}_{i,j}^n$
$\mathbf{I}_i$	Visualized line plot image
S_i	Set of patch-wise textual captions $\{s_{i,j}\}$
D	Embedding dimension of above encoders and the LLM backbone
$\mathbf{E}_i^n$	Encoded numerical embeddings, composed of tokens $\mathbf{e}_{i,j}^n$
$\mathbf{E}_i^v$	Encoded visual embeddings, composed of tokens $\mathbf{e}_{i,j}^v$
$\mathbf{E}_i^s$	Encoded caption embeddings, composed of tokens $\mathbf{e}_{i,j}^s$
Discrete Disentangled Interaction (DDI)	
$\mathbf{Z}_i^n, \mathbf{Z}_i^v$	Modality-common representations, composed of tokens $\mathbf{z}_{i,j}^n, \mathbf{z}_{i,j}^v$
$\mathbf{U}_i^n, \mathbf{U}_i^v$	Modality-unique representations, composed of tokens $\mathbf{u}_{i,j}^n, \mathbf{u}_{i,j}^v$
d	Reduced embedding dimension during vector quantization
$\mathcal{C}^{(m)}$	The m -th codebook in hierarchical vector quantization
$\mathbf{R}_i^{n;(m)}, \mathbf{R}_i^{v;(m)}$	Residuals at quantization stage m , composed of tokens $\mathbf{r}_{i,j}^{n;(m)}, \mathbf{r}_{i,j}^{v;(m)}$
$\tilde{\mathbf{R}}_i^{n;(m)}, \tilde{\mathbf{R}}_i^{v;(m)}$	Pooled and broadcasted residuals at stage m , composed of tokens $\tilde{\mathbf{r}}_{i,j}^{n;(m)}, \tilde{\mathbf{r}}_{i,j}^{v;(m)}$
$\mathbf{Q}_i^{n;(m)}, \mathbf{Q}_i^{v;(m)}$	Quantized discrete modality representations, composed of $\mathbf{q}_{i,j}^{n;(m)}, \mathbf{q}_{i,j}^{v;(m)}$
$\bar{\mathbf{E}}_i^n, \bar{\mathbf{E}}_i^v$	Augmented modality tensors after cross-modal interaction
Critical-token Highlighting (CTH)	
$\mathbf{Q}^q$	Learnable queries for the question-conditioned branch
$\mathbf{Q}^n, \mathbf{Q}^v$	Learnable queries for the modality-intrinsic branch
$\mathbf{H}_i^{n;q}, \mathbf{H}_i^{v;q}$	Highlighted tokens from the question-conditioned branch
$\mathbf{H}_i^{n;s}, \mathbf{H}_i^{v;s}$	Highlighted tokens from the modality-intrinsic branch
$\mathbf{H}_i^n, \mathbf{H}_i^v$	Highlighted tokens fused from question-conditioned and modality-intrinsic branches
$\hat{\mathbf{E}}_i^n, \hat{\mathbf{E}}_i^v$	Final enhanced modality embeddings prepended with highlighted tokens
$\mathbf{E}^*$	Final fused multi-modal sequence input to the LLM
Optimization Objectives & Hyperparameters	
$\mathcal{L}_{PA}$	Total Patch-level Alignment loss
$\mathcal{L}_{align}^{n-v}, \mathcal{L}_{align}^{n-s}$	Contrastive alignment losses for numerical-visual and numerical-caption pairs
$\mathcal{L}_{DDI}$	Total Discrete Disentangled Interaction loss
$\mathcal{L}_{vq}$	Vector quantization commitment loss
$\mathcal{L}_{com}$	Bidirectional contrastive loss for modality-common semantics
$\mathcal{L}_{orth}$	Orthogonality loss between common and unique components
$\mathcal{L}_{LM}$	Language modeling loss
τ	Temperature hyperparameter for contrastive learning
α, β	Weights for disentanglement regularization terms
λ_1, λ_2	Loss weights for PA and DDI modules

In this work, we adopt the ready-made training data released by ChatTS, which contain 111,408 TSUR samples.

B.1.2 Evaluation Data Composition

The evaluation dataset is a hybrid of 411 real-world samples and 985 synthetic samples, designed to assess out-of-distribution generalization.

The synthetic samples are generated using the same pipeline described in Appendix B.1.1. The real-world data are collected from diverse domains, including AIOps (system monitoring metrics), weather (temperature and humidity), finance (stock trends), and traffic flow. All real-world samples are manually annotated by domain experts to ensure annotation accuracy (Xie et al., 2025).

Concretely, the evaluation dataset comprises two primary task categories:

- **Understanding Tasks:** These tasks evaluate the model’s ability to ground textual concepts in time series features. Subtasks include identifying *Trend* (increase/decrease), *Seasonality* (period length), *Noise Levels*, and *Local Fluctuations* (spike/dip detection). For multivariate time series, additional tasks include *Correlation* analysis and *Clustering*. Questions are presented in both categorical (multiple-choice) and numerical (value extraction) formats.
- **Reasoning Tasks:** These tasks assess higher-order reasoning capabilities. Subtasks include *Inductive Reasoning* (inferring physical meanings from patterns), *Deductive Reasoning* (predicting behavior under given conditions), *Causal Reasoning* (identifying root causes of anomalies), and *Comparison* (contrasting distinct time series). Questions are provided in categorical or open-ended formats.

B.2 Baselines

To ensure rigorous comparisons, we organize baselines according to the modality formulation used to represent time series: *numerical-centric*, *visual-centric*, and *numerical+visual*. Across these settings, general-purpose LLMs/MLLMs are accessed via their official APIs, while time-series-specialized MLLMs are reproduced and fine-tuned under controlled settings.

Numerical-Centric Baselines. Numerical-centric baselines operate directly on time-series values. For general-purpose LLMs, including GPT-4o (OpenAI, 2024), Qwen3 (Yang et al., 2025), DeepSeek-V3.2 (DeepSeek-AI, 2025), Gemini 3 Pro (Google DeepMind, 2026), and GPT-5.2 (OpenAI, 2025), we serialize time-series values into textual prompts and query the models through their official APIs. We also include time-series-specialized MLLMs that introduce dedicated time-series encoders or instruction tuning mechanisms, including ChatTime (Wang et al., 2025a), ChatTS (Xie et al.,

2025), ITFormer (Wang et al., 2025b), and Instruct-Time (Cheng et al., 2025).

Visual-Centric Baselines. Visual-centric baselines represent time series as plots and require models to reason from visual inputs. We evaluate general-purpose MLLMs, including GPT-4o (OpenAI, 2024), Qwen3-VL (Bai et al., 2025a), Gemini 3 Pro (Google DeepMind, 2026), and GPT-5.2 (OpenAI, 2025), using their official APIs. Following (Xie et al., 2025), we render time series as line plots; for multivariate data, plots of different variables are stacked into a single composite figure.

Numerical+Visual Baselines. Numerical+visual baselines integrate both serialized numerical values and rendered time series plots as input. For general-purpose MLLMs, we provide both modalities to GPT-4o (OpenAI, 2024), Qwen3-VL (Bai et al., 2025a), Gemini 3 Pro (Google DeepMind, 2026), and GPT-5.2 (OpenAI, 2025). We additionally include GEM (Lan et al., 2025), a time-series-specialized multi-modal baseline trained to leverage numerical and visual information jointly.

For all time-series-specialized baselines, we unify the LLM backbone to ensure a fair comparison that emphasizes architectural and modality-design differences rather than base model capacity. Specifically, specialized models are reproduced and fine-tuned using either Qwen2.5-7B-Instruct (Yang et al., 2024) or Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), depending on whether the method requires numerical-only or numerical–visual multimodal inputs. We maintain identical training data and experimental environments across all specialized baselines and our proposed MADI.

B.3 Evaluation Metrics

We adopt different evaluation metrics for categorical, numerical, and open-ended reasoning tasks.

Categorical Tasks. For classification-based questions (e.g., trend direction, fluctuation type), we report *Accuracy* and *F1-score*, measuring the agreement between predicted labels and ground truth.

Numerical Tasks. For tasks requiring numerical value estimation (e.g., spike amplitude or period length), we employ *Relative Accuracy* (Acc_{rel}) with thresholding to reduce sensitivity to outliers:

$$\text{Acc}_{\text{rel}} = \max \left(0, 1 - \left| \frac{v_{\text{pred}} - v_{\text{label}}}{v_{\text{label}}} \right| \right), \quad (3)$$

where v_{pred} denotes the value extracted from the model response and v_{label} is the ground-truth value.

Open-ended Reasoning Tasks. For open-ended reasoning tasks, we evaluate *Answer Correctness* using a criterion inspired by the RAGAS evaluation framework (Es et al., 2024). This metric uses a judge LLM (GPT-4o-mini) to extract key facts from the ground-truth answer and computes the recall of these facts in the model response, enabling semantic evaluation beyond naive keyword matching. The complete prompts used by the LLM evaluator for open-ended reasoning QA are provided in Appendix J. We further validate the reliability of this LLM-based evaluation through a blinded human study in Appendix E.

B.4 Implementation Details

This section describes the implementation details of MADI, including data preprocessing, model hyperparameters, and training/evaluation configurations. **Statistics-Preserved Normalization.** To improve training stability and reduce distribution shifts across heterogeneous time series domains, we apply zero-centered normalization to the input series prior to patch-level modality expansion. While normalization is essential for effective optimization, it removes absolute magnitude information that is crucial for precise numerical reasoning (e.g., answering queries such as “what is the maximum value?”). To address this limitation, we introduce a *Statistics-Preserved Prompt* that explicitly encodes global statistics of the original time series and exposes them to the LLM during reasoning. Specifically, for each time series instance $\mathbf{x}_i$, we construct a structured textual prompt containing its normalization parameters (offset and scaling factor), sequence length, and key boundary statistics, including the maximum, minimum, start, and end values. This prompt is prepended to the processed multi-modal tokens $\hat{\mathbf{E}}_i^n$, enabling the LLM to recover the original scale and perform accurate value-based reasoning.

The statistics-preserved prompt follows the format: “[offset=... |scaling=... |length=... |max=... |min=... |left=... |right=...]”.

- offset: the offset applied during normalization (e.g., the negative mean);
- scaling: the scaling factor for normalization;
- length: the length of the time series sequence;
- max / min: the maximum and minimum values of the original series;

- left / right: the first/last values in the sequence.

By exposing these global statistics, MADI leverages the arithmetic reasoning capabilities of the LLM backbone to interpret normalized temporal patterns with respect to their original physical magnitudes. For consistency across modalities, visual plots and patch-level captions are also generated from the normalized series, ensuring alignment with the numerical inputs.

Time series Rendering. For constructing the visual modality, we use Matplotlib to render each normalized time series as a line plot. Following the patch-aligned visualization design in Section 4.1.1, we remove decorative elements, including titles, axes, ticks, labels, and legends, and use the default line-plot rendering settings unless otherwise specified, so that visual patches maintain one-to-one correspondence with numerical patches.

Model Hyperparameters. The embedding dimension is set to $D = 3584$ to match the LLM backbone. We use a numerical patch size of $p^n = 8$, a discretization dimension of $d = 512$, and $m = 3$ hierarchical codebook levels. The overall loss is a weighted sum of multiple components, with weights $\alpha = 5$ (modality-common alignment), $\beta = 1$ (common-unique orthogonalization), $\lambda_1 = 0.02$ (PA regularization), and $\lambda_2 = 0.2$ (DDI regularization). The discretization dimension and loss weights are determined through a lightweight hyperparameter search over a small candidate set, while the remaining hyperparameters are kept fixed based on backbone compatibility and computational feasibility.

Training Configuration. In this study, we use Qwen2.5-VL-7B-Instruct as the pre-trained backbone. We implement our training pipeline using the LLaMA-Factory framework on $4 \times$ NVIDIA A800 GPUs, with DeepSpeed acceleration and FlashAttention enabled for efficient distributed fine-tuning and optimized attention computation. Specifically, we adopt a per-device batch size of 2 with 64 gradient accumulation steps, resulting in an effective batch size of $512 = 4 \times 2 \times 64$. Optimization is performed using AdamW with a learning rate of 1×10^{-5} and a cosine learning rate scheduler, together with a warmup ratio of 0.02. Training proceeds for 1200 steps. For all other configurations, we use the default settings provided by LLaMA-Factory. We employ a two-stage training strategy: during the initial warmup phase (0.02 of total train-

ing), the LLM backbone and vision encoder (for Qwen2.5-VL-based models) are frozen, and only newly introduced modules are trained; afterward, all parameters are unfrozen for full fine-tuning. For our MADI, the first training stage takes around 56 minutes, while the second stage requires about 73.5 hours using 4 GPUs.

Evaluation Configuration. During evaluation, we set the decoding temperature to 0.01 to ensure stable and reproducible results across all benchmarks. For categorical and numerical questions, we compute Accuracy, F1-Score, and Relative Accuracy using scikit-learn implementations with default settings. For open-ended reasoning questions, we follow the evaluation protocol of ChatTS, using GPT-4o-mini as the judge LLM with the same configuration for open-ended evaluation. Evaluation is conducted every 50 steps, and the checkpoint with the best evaluation performance is selected for final reporting. Due to the large dataset scale and the need for repeated checkpoint evaluations during training/evaluation, we report results from a single run under the above fixed configuration. To validate the robustness of our model, we additionally repeat the training and evaluation of MADI and representative baselines to quantify run-to-run variation, as detailed in Appendix F.

C Cross-domain Evaluation

To evaluate cross-dataset and cross-domain generalization, we use TSRBench (Yu et al., 2026), an independent benchmark spanning applications such as finance, energy, traffic, agriculture, meteorology, astronomy, seismology, and human activity. We evaluate its perception, decision-making, and reasoning tasks in the form of multiple-choice questions, and report accuracy for each task group in Table 4. MADI achieves the best decision and reasoning accuracy while remaining competitive on perception, demonstrating a strong ability to generalize across datasets and application domains.

Table 4: Comparison performance (%) on TSRBench.

Method	Perception	Decision	Reasoning
ChatTime	41.89	29.69	27.54
ChatTS	41.94	36.36	30.45
ITFormer	41.11	33.33	23.56
InstructTime	47.19	39.02	27.81
GPT-5.2 (Visual)	73.12	41.98	31.86
GEM	52.69	30.23	24.68
MADI	<u>70.22</u>	50.62	37.61

D Token Cost Analysis

In this section, we evaluate the computational efficiency of different paradigms by calculating the average token consumption per query in real-world understanding tasks, as presented in Table 5. This reveals significant disparities driven by tokenization strategies and input modalities.

In the *Numerical-centric* setting, general-purpose LLMs exhibit prohibitively high consumption, consistently exceeding 7,000 tokens per question. This inefficiency stems from their reliance on digit-level tokenization, where a single time-series value is decomposed into multiple tokens. Even time-series-specific adaptations like ChatTime, which utilizes value-level tokenization, incur a noticeably higher computational cost compared to patch-based models (e.g., ITFormer, ChatTS, MADI) that efficiently aggregate temporal segments into compact vector representations. The computational burden is further exacerbated in the *Numerical+Visual* setting. For standard Multimodal LLMs such as Qwen3-VL and GPT-4o, the straightforward concatenation of lengthy numerical text sequences with visual embeddings results in excessive sequence lengths, often surpassing 10,000 tokens and significantly increasing the inference overhead. In contrast, MADI incorporates rich multi-modal semantics with only a marginal increase in token count compared to the most efficient baselines. Consequently, MADI achieves the best performance trade-off, delivering robust reasoning capabilities with a modest token cost.

E Human Evaluation

We conduct a blinded human evaluation to assess whether the LLM-based Answer Correctness scores reliably reflect the quality of open-ended reasoning responses. Internal experts from the author team, who have relevant research backgrounds and are familiar with the topic, voluntarily annotate the responses using the same evaluation instructions and criteria as the LLM judge. They receive no payment and are informed that the evaluation data are used solely for research. No separate ethics review board approval was sought because this was an internal expert annotation study rather than a new human-subject data collection procedure. As shown in Table 6, the human score closely tracks the LLM score, with a high Spearman rank correlation of 0.828, supporting the reliability of the LLM-based evaluation.

Table 5: Average token cost per query across different kinds of models.

Type	Numerical								
Model	DeepSeek	GPT-4o	Qwen3	GPT-5.2	Gemini	InstructTime	ITFormer	ChatTime	ChatTS
Cost	7135.90	7140.04	9935.51	7139.04	9702.75	1108.16	1108.16	2503.01	1118.90

Type	Visual				Num.+Visual					
Model	GPT-4o	Qwen3-VL	GPT-5.2	Gemini	GPT-4o	Qwen3-VL	GPT-5.2	Gemini	GEM	MADI
Cost	1063.49	1390.13	1111.86	1643.47	7698.82	10800.71	7864.87	10800.93	1174.99	1517.79

Table 6: Human evaluation of open-ended reasoning.

LLM Eval. (%)	Human Eval. (%)	Spearman ρ
57.68	55.34	0.828

F Repeated Runs

To assess robustness to run-to-run variation, we rerun MADI and five representative baselines twice and report the mean and sample standard deviation over three runs in Table 7. The standard deviations are consistently small, while MADI retains the best mean performance on all four aggregate metrics. These results support the robustness and consistent performance advantage of MADI.

G Case Study

In this section, we present qualitative case studies across a spectrum of time series understanding and reasoning tasks. These examples serve to empirically demonstrate the capabilities of MADI compared to three competitive baselines: Gemini 3 Pro (Visual-centric), ChatTS (Numerical-centric), and GEM (Numerical+Visual).

Noise Understanding. Figure 6 evaluates noise quantification using solar radiation data, a domain characterized by subtle, high-frequency fluctuations. Although baseline models (*e.g.*, Gemini 3 Pro) fail to perceive these fine-grained perturbations, incorrectly classifying the series as noiseless, MADI exhibits superior sensitivity. It accurately detects the “noisy” nature of the signal and provides a precise standard deviation estimate of 0.05. This underscores the model’s ability to distinguish inherent signal stochasticity from clean trends, a capability often lacking in existing methods.

Local Fluctuation Understanding. Figure 7 illustrates the challenge of characterizing morphological features within traffic signal data. The series exhibits complex local behaviors, specifically a rapid “shake” followed by a sharp decline. Baselines

like Gemini 3 Pro and ChatTS oversimplify these patterns, hallucinating generic “convex” shapes or “upward spikes” In contrast, MADI demonstrates high morphological fidelity, accurately localizing and classifying the “shake” at point 149 and the “decrease after downward spike” at point 185. This highlights the model’s precision in parsing fine-grained structural changes in temporal data.

Seasonality Understanding. Figure 8 examines seasonality extraction in content marketing performance data, where regular cycles are disrupted by a significant anomaly. This irregularity causes baselines to falter: ChatTS and GEM are sensitive to the outlier, resulting in a gross overestimation of amplitude, while Gemini 3 Pro underestimates it. MADI, however, proves robust to such contamination, correctly identifying the period (25.00) and providing a reliable amplitude estimate (852.00). This demonstrates the model’s capacity to identify true periodic features from sporadic irregularities.

Trend Understanding. Figure 9 depicts a trend analysis task on external service call latency data. The series is defined by high volatility and a late-stage downward shift. Although baselines like Gemini 3 Pro and GEM misinterpret the global behavior as strictly “Decreasing”, and ChatTS fail to capture the precise values of the starting point and global shift, MADI correctly identifies the overall “Steady” state while accurately quantifying the specific trend change value (-62.01). This validates the model’s ability to maintain a global perspective despite local volatility.

Correlation Understanding and Inductive Reasoning. Figure 10 presents a complex task requiring both correlation analysis and inductive reasoning on network “Jitter” and “Throughput” metrics. Baselines (*e.g.*, Gemini 3 Pro, ChatTS) often rely on superficial signal characteristics (*e.g.*, smoothness) or rigid definitions to incorrectly deny a relationship. MADI, conversely, looks past local signal disparities to identify the shared long-term trends

Table 7: Repeated evaluation results (% , mean $\pm$ sample standard deviation over three runs).

Method	Synthetic		Real-world	
	Overall Cate.	Overall Num.	Overall Cate.	Overall Num.
ChatTime	75.15 $\pm$ 0.21	32.24 $\pm$ 0.18	75.08 $\pm$ 0.39	34.70 $\pm$ 0.17
ChatTS	85.33 $\pm$ 0.01	79.52 $\pm$ 0.02	82.39 $\pm$ 0.00	77.04 $\pm$ 0.05
ITFormer	50.22 $\pm$ 0.32	28.78 $\pm$ 0.83	52.19 $\pm$ 0.26	38.77 $\pm$ 0.08
InstructTime	49.30 $\pm$ 0.55	28.43 $\pm$ 1.07	51.42 $\pm$ 0.11	36.33 $\pm$ 0.15
GEM	84.38 $\pm$ 0.05	75.26 $\pm$ 0.03	83.86 $\pm$ 0.31	78.10 $\pm$ 0.45
MADI	89.98 $\pm$ 0.01	87.21 $\pm$ 0.03	88.54 $\pm$ 0.00	83.86 $\pm$ 0.17

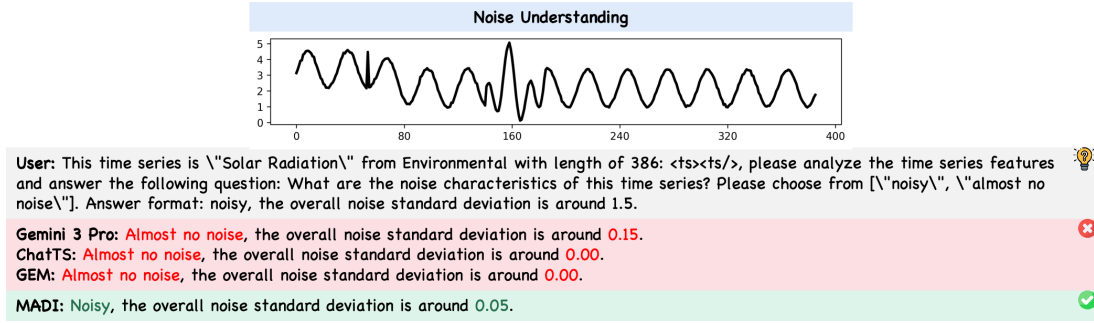


Figure 6: Case study of noise understanding.

driven by the underlying system physics. This confirms the model’s ability to perform semantic alignment between coupled time series variables.

Causal Reasoning. Figure 11 demonstrates a causal reasoning scenario using cost-per-click (CPC) data. The series features a precipitous drop around point 110, stabilizing at a lower level. MADI outperforms baselines by grounding its reasoning in domain logic, correctly attributing the shift to a “temporary drop in advertiser demand” and predicting a subsequent state of “stable market competition” Other models, such as Gemini 3 Pro, hallucinate technical failures (“system misconfiguration”), revealing a lack of domain-aware causal inference ability.

Comparison Reasoning. Figure 12 involves a comparative analysis of influencer spike frequency across two video upload time series. A common pitfall for baselines is conflating signal amplitude with event frequency. Consequently, all baselines (Gemini 3 Pro, ChatTS, GEM) hallucinate an increased frequency in the “Updated” series due to its visual characteristics. MADI avoids this error, correctly observing that the spikes occur with the same frequency despite amplitude variations, effectively decoupling temporal rate from intensity.

Deductive Reasoning. Figure 13 tests the model’s ability to apply strict logical constraints to travel

speed data, specifically regarding anomaly thresholds and noise patterns. While ChatTS and GEM fail to detect the anomaly, and Gemini 3 Pro fails to logically deduce the “typical” nature of the high-noise behavior in the follow-up question, MADI exhibits robust deductive capabilities. It explicitly verifies the amplitude condition ($54.14 > 50$) to detect the anomaly and correctly reasons that the observed fluctuations align with normal traffic variations given the problem’s specific constraints.

H Artifact Statement

Time-series QA data. We use the ChatTS datasets for training and evaluation. The training data are publicly hosted on Hugging Face¹. Their repository metadata indicates Apache-2.0, while the accompanying ChatTS project page states the MIT License². The evaluation sets are publicly hosted on Zenodo under CC BY 4.0³. We use these datasets only for research on time series understanding and reasoning, which is consistent with their released purpose. The artifacts consist of English language-time-series QA examples, including synthetic train-

¹<https://huggingface.co/datasets/ChatTSRepo/ChatTS-T raining-Dataset>

²<https://github.com/NetManAIOps/ChatTS/blob/main/LI CENSE>

³<https://doi.org/10.5281/zenodo.14349206>

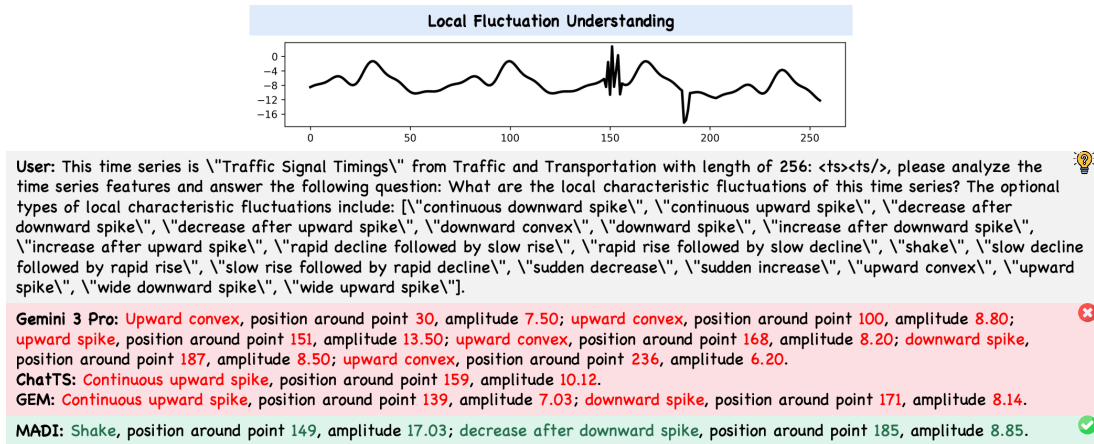


Figure 7: Case study of local fluctuation understanding.

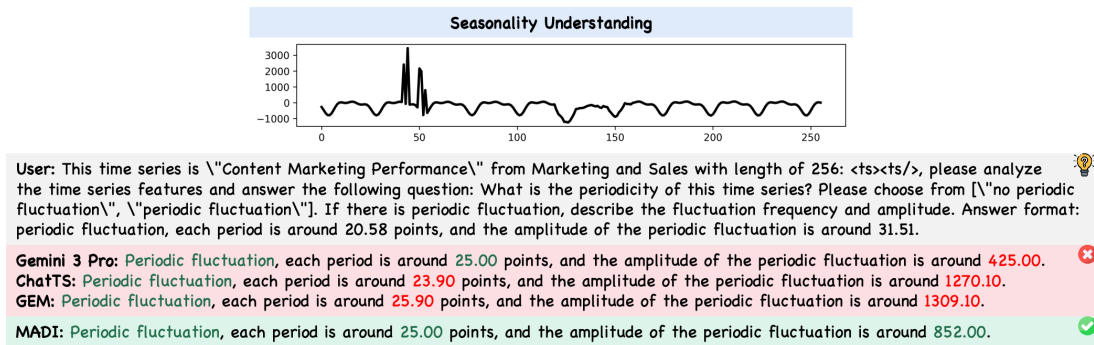


Figure 8: Case study of seasonality understanding.

ing examples generated by ChatTS and hybrid synthetic/real-world evaluation examples covering AIOps, weather, finance, and traffic-flow domains. The benchmark covers temporal understanding phenomena such as trend, seasonality, noise, local fluctuation, correlation, and clustering. It also covers reasoning abilities including inductive, deductive, causal, and comparative reasoning. Since the QA data are generated or manually reviewed by the original benchmark creators, they are not expected to contain private user records or personally identifying information. We additionally inspected a random 10% sample of the QA instances, including questions, answers, textual descriptions, and associated time series. We found no content that names or uniquely identifies individual people, and no offensive content was observed.

Pretrained backbone models. We use Qwen2.5-7B-Instruct⁴ and Qwen2.5-VL-7B-Instruct⁵ as pretrained LLM/MLLM backbones. Both models are publicly available on Hugging Face and are re-

leased under Apache-2.0. Their intended use includes research and development of language and vision-language applications, which is compatible with our use as backbones for controlled TSUR experiments. We do not redistribute the original model weights under different terms. All use of these models in this work is limited to research training, reproduction, and evaluation.

Baseline and processing code. We use code resources only for research reproduction, preprocessing, and evaluation. The official ChatTS GitHub repository⁶ provides the implementation together with training data, evaluation data, and model links, and is released under the MIT License. We use its data construction and evaluation-related code for TSUR research. The official ITFormer GitHub repository⁷ is released under the MIT License and provides training and inference scripts for temporal-textual multimodal QA. We use it only as a research baseline implementation. The official GEM

⁴<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct/blob/main/LICENSE>

⁵<https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct>

⁶<https://github.com/NetManAIOps/ChatTS/blob/main/LICENSE>

⁷<https://github.com/Pandalin98/ITFormer-ICML25/blob/main/LICENSE>

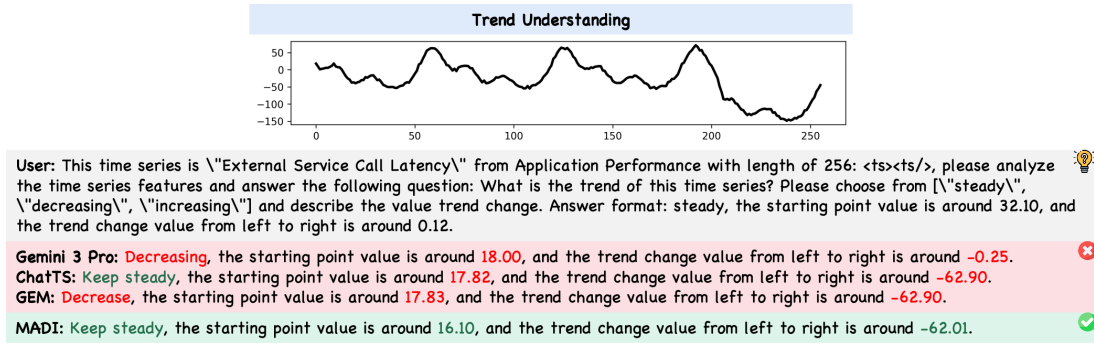


Figure 9: Case study of trend understanding.

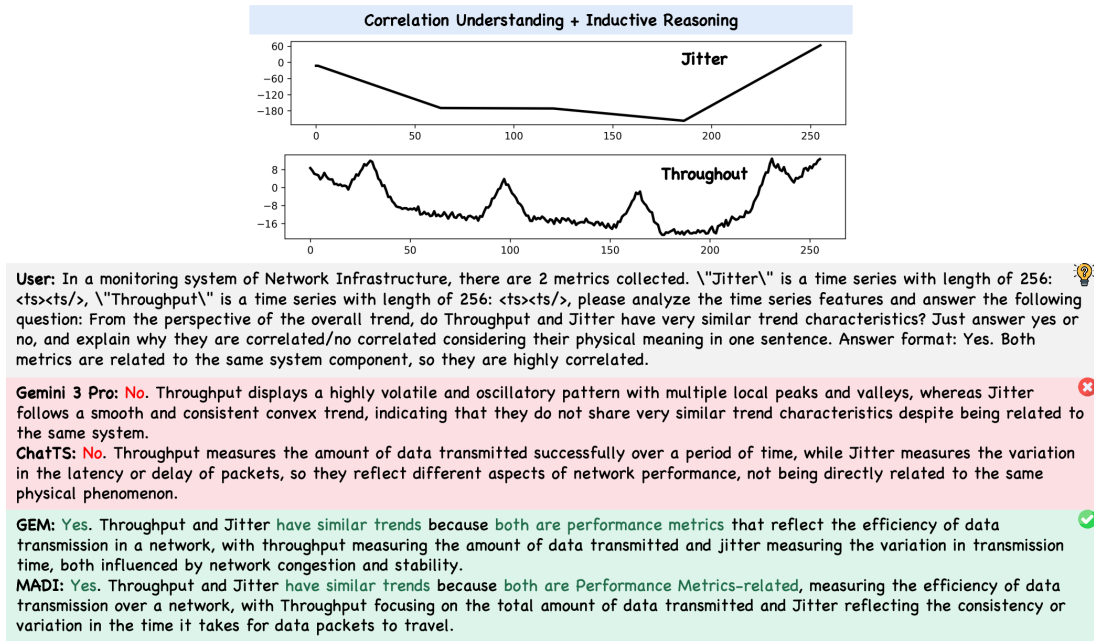


Figure 10: Case study of correlation understanding and inductive reasoning.

GitHub repository⁸ is released under Apache-2.0. We use its implementation only as a research baseline reference. For ChatTime⁹ and InstructTime¹⁰, we use the public repositories as baseline code references, but we did not find explicit license files in the repository files we accessed. Therefore, we will not redistribute their code, and only use them for internal research comparison. Our own code for MADI implementation, data processing, and model training is intended for research use. If released, it will be provided under a permissive research-compatible license consistent with common open-source practice.

⁸<https://github.com/lanxiang1017/GEM/blob/main/LICE>

⁹<https://github.com/ForestsKing/ChatTime>

¹⁰<https://github.com/ustc-time-series/InstructTime>

I Statement on the Use of AI Assistance

We independently conceived the core ideas, defined the research scope, performed all experimental analyses, constructed the logical structure, and finalized the figures and manuscript. LLMs (e.g., GPT and DeepSeek) were used as auxiliary tools, primarily for code development and text polishing. They did not play any role in hypothesis formation, scientific judgment, or result interpretation. This work was therefore led and completed by the authors, with AI assistance strictly limited to non-essential, supporting tasks.

J LLM Evaluator Prompt for Open-ended Reasoning

We use a two-stage LLM evaluator for open-ended reasoning QA. The first prompt extracts the minimal answer-bearing keywords from the reference

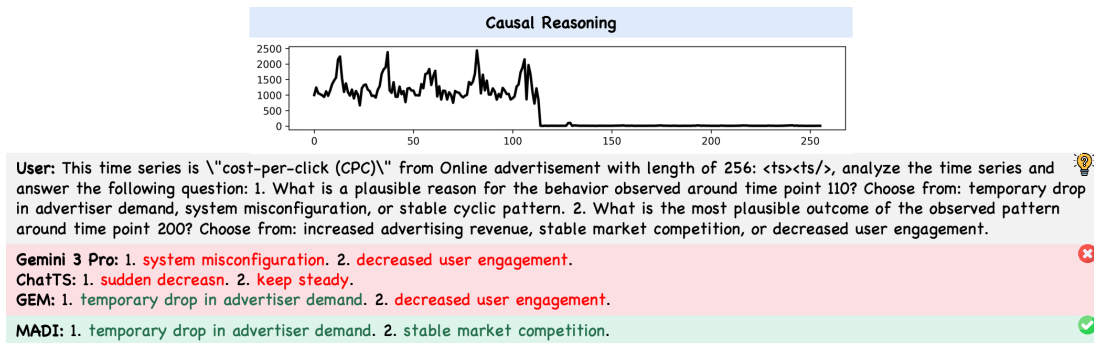


Figure 11: Case study of causal understanding.

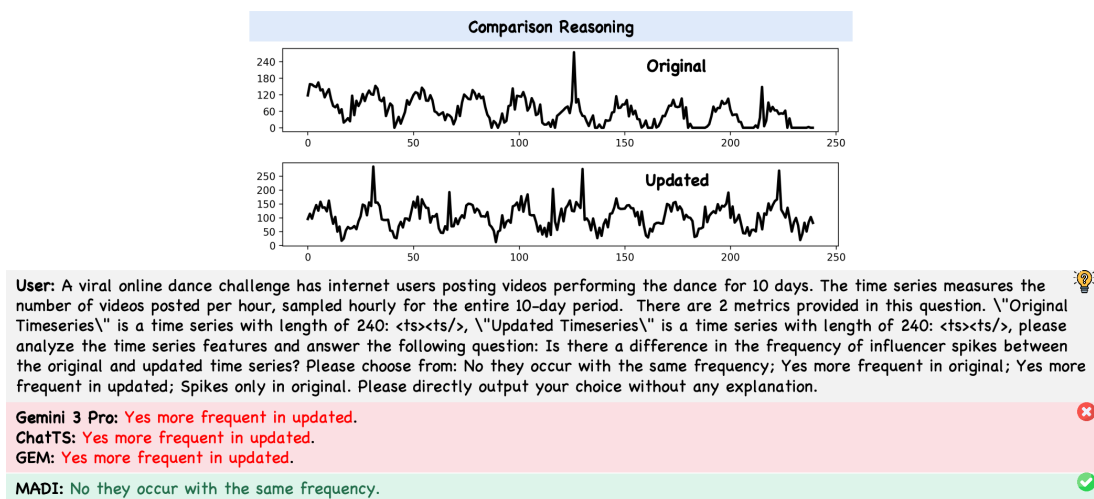


Figure 12: Case study of comparison understanding.

answer. The second identifies which of these keywords are covered by the model response, and the final score is the fraction of extracted keywords covered by the response.

Ground-truth Keyword Extraction

Given a question and the ground-truth answer, extract the following information:

\"gt_keywords\": Identify and return a list of keywords or phrases contained in the ground-truth answer. The list should include only the minimal combination of key points that directly answer the question and should avoid words unrelated to the question as much as possible. Each list should contain only 1 or 2 keywords, and the keywords should be as concise and as short as possible.

Inputs:

Question: {QUESTION}

Ground-truth answer: {GROUND_TRUTH_ANSWER}

Few-shot examples:

Example 1

Question: What protocol does AMF use to ensure correct time?

Ground-truth answer: The role of NTP is to synchronize the time of all clocked devices within the network, ensuring that the clock

times of all devices in the network remain basically consistent, so that the devices can provide various applications based on unified time. Possible applicable network elements: AMF, MME, SGSN.

Output: {\"gt_keywords\": [\"NTP\"]}

Example 2

Question: What is the purpose of MME assigning a TA List to UE?

Ground-truth answer: The purpose of MME assigning a TA List to UE is to manage the location of idle UEs. After MME assigns a TA List to UE, the UE will not initiate a TA update to MME when moving within this TA List. When the UE moves out of the TA List range, it will initiate a TA update, and the MME will know that the location of the idle UE is within the TA List range. When paging is needed, it can be done within the TA List range.

Output: {\"gt_keywords\": [\"manage the location of idle UEs\"]}

Answer format in JSON:

{\"gt_keywords\": [\"KEYWORD 1\", \"KEYWORD 2\"]}



Figure 13: Case study of deductive understanding.

Answer Coverage Evaluation

Given a question, an answer generated by the model, and the list of ground-truth keywords ("gt_keywords"), extract the following information:

"overlapping_keywords": From the list of "gt_keywords", identify any terms or phrases that also appear in the model's answer. These overlapping keywords indicate the points of agreement or coverage between the model's answer and the ground truth.

Inputs:

Question: {QUESTION}

Ground-truth keywords: {GROUND_TRUTH_KEYWORDS}

Model answer: {MODEL_ANSWER}

Few-shot examples:

Example 1

Question: What powers the sun and what is its primary function?

Ground-truth keywords: ["nuclear fusion", "energy", "light", "essential for life", "climate system", "weather", "ocean currents"]

Model answer: The sun is powered by nuclear fission, similar to nuclear reactors on Earth, and its primary function is to provide light to the solar system.

Output: {"overlapping_keywords": ["light"]}

Example 2

Question: What is the boiling point of water?

Ground-truth keywords: ["100 degrees Celsius", "212 degrees Fahrenheit", "sea level", "change

with altitude"]

Model answer: The boiling point of water is 100 degrees Celsius at sea level.

Output: {"overlapping_keywords": ["100 degrees Celsius", "sea level"]}

Example 3

Question: What information should be submitted when contacting technical support for a communication technology company?

Ground-truth keywords: ["fault details", "log files and alarm query results", "steps taken to address the issue", "commands executed", "results of those actions", "remote access details", "contact information for relevant personnel"]

Model answer: When contacting technical support for a communication technology company, the following information should be provided: 1. Fault details: Time, location, and event description.

Output: {"overlapping_keywords": ["fault details"]}

Example 4

Question: What are the benefits of a balanced diet?

Ground-truth keywords: ["provides essential nutrients", "maintains a healthy weight", "reduces risk of chronic diseases", "supports overall health"]

Model answer: A balanced diet helps maintain a healthy weight and supports overall health.

Output: {"overlapping_keywords": ["maintain a healthy weight", "supports overall health"]}

Example 5

Question: What is the capital of France?

Ground-truth keywords: ["Paris"]

Model answer: The capital of France is Paris.

Output: {"overlapping_keywords": ["Paris"]}

Example 6

Question: How does photosynthesis work?

Ground-truth keywords: ["process by which plants convert sunlight into energy", "involves chlorophyll", "produces oxygen", "occurs in the chloroplasts"]

Model answer: Photosynthesis is the process by which plants use sunlight to produce energy and oxygen.

Output: {"overlapping_keywords": ["process by which plants convert sunlight into energy", "produces oxygen"]}

Answer format in JSON:

{"overlapping_keywords": ["COVERED KEYWORD 1", "COVERED KEYWORD 2"]}