AXIS: EXPLAINABLE TIME SERIES ANOMALY DETECTION WITH LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series anomaly detection (TSAD) increasingly demands explanations that articulate not only if an anomaly occurred, but also what pattern it exhibits and why it is anomalous. Leveraging the impressive explanatory capabilities of Large Language Models (LLMs), recent works have attempted to treat time series as text for explainable TSAD. However, this approach faces a fundamental challenge: LLMs operate on discrete tokens and struggle to directly process long, continuous signals. Consequently, naïve time-to-text serialization suffers from a lack of contextual grounding and representation alignment between the two modalities. To address this gap, we introduce AXIS, a framework that conditions a frozen LLM for nuanced time-series understanding. Instead of direct serialization, AXIS enriches the LLM's input with three complementary hints derived from the series: (i) a symbolic numeric hint for numerical grounding, (ii) a context-integrated, step-aligned hint distilled from a pretrained time-series encoder to capture fine-grained dynamics, and (iii) a task-prior hint that encodes global anomaly characteristics. Furthermore, to facilitate robust evaluation of explainability, we introduce a new benchmark featuring multi-format questions and rationales that supervise contextual grounding and pattern-level semantics. Extensive experiments, including both LLM-based and human evaluations, demonstrate that AXIS yields explanations of significantly higher quality and achieves competitive detection accuracy compared to general-purpose LLMs, specialized time-series LLMs, and time-series Vision Language Models. The code is available in `https://anonymous.4open.science/r/TimeSemantic-1742/main.py`

1 INTRODUCTION

Time Series Anomaly Detection (TSAD) is essential for safeguarding critical systems across domains (Iqbal et al., 2019; Zeufack et al., 2021; Hundman et al., 2018). While deep learning models can detect anomalies with high accuracy (Fig. 1(a)), their adoption in real-world systems is limited by two challenges. First, their reasoning process is essentially a black box. Experts remain in the dark when asking the most practical question: why was this anomaly flagged? Post-hoc attribution methods such as SHAP (Fig. 1(b)) attempt to fill this void, but they merely repackage correlations into statistical features. Such attributions reveal what inputs influenced the model, but they stop short of offering why the underlying anomaly event occurred. Second, these models are brittle. Trained narrowly - often on a single dataset - they capture dataset-specific features rather than generalizable patterns. In fast-changing environments where anomalies manifest in diverse forms, this rigidity is crippling: models must be retrained at high cost, yet still fail to transfer across domains. What is missing is the ability to handle anomalies universally—to recognize and adapt across diverse failure patterns without exhaustive retraining.

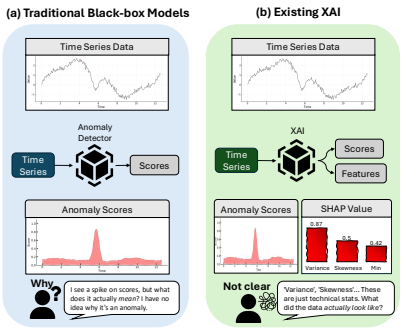

Figure 1: Deep learning method for TSAD: (a) Opaque anomaly scores fail to explain why; (b) XAI features lack intuitive semantics;

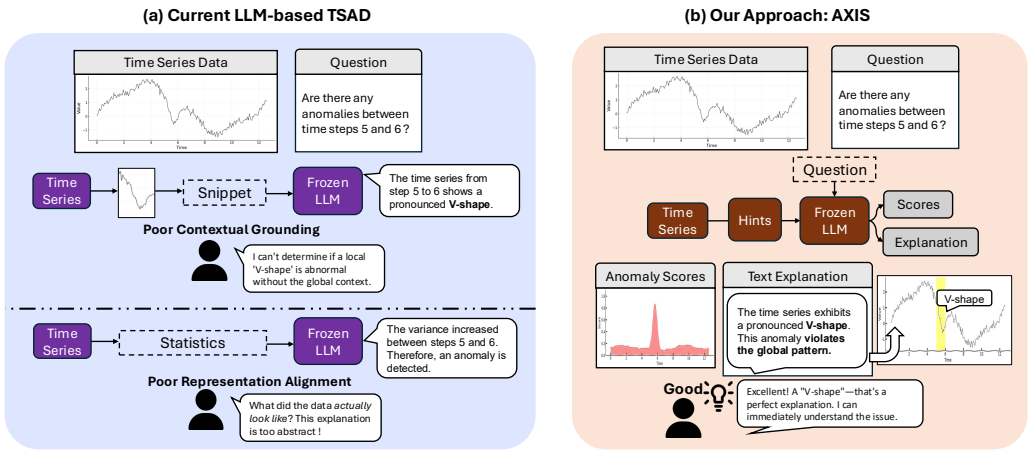

Figure 2: Bridging the Semantic Gap in Time Series Anomaly Explanation. (a) Current LLM-based methods fail due to: (i) poor **Contextual Grounding**, where observing a local pattern (e.g., the "V-shape") in isolation prevents a meaningful diagnosis; and (ii) **Representation Misalignment**, where inputs of abstract statistics (e.g., "variance increased") lead to uninformative, circular explanations. (b) Our approach overcomes these limitations by producing contextualized, pattern-level explanations that align with expert reasoning.

In response to these limitations, the community has turned to Large Language Models (LLMs), celebrated for their fluency and generalization. Yet critical obstacles remain: LLMs operate on discrete tokens, making them ill-suited for the long, continuous nature of time series (Dong et al., 2024). Attempting to fit these signals into tokenized inputs often incurs lossy serialization, forcing workarounds that undercut the LLM's capabilities. Common strategies—feeding isolated fragments or pre-aggregated statistics—reduce the model to a mere post-hoc translator. As our motivating example illustrates (Fig. 2(a)), this naive approach, however, suffers a critical semantic gap, driven by two fundamental failures.

The first is a lack of **Contextual Grounding**. By analyzing only a narrow snippet of the series, the LLM is deprived of the broader temporal context required to discern whether a local pattern is genuinely anomalous or merely a benign fluctuation. The second is a failure of **Representation Alignment**, which creates a chasm between the model's analytical basis and human intuition. When an LLM is fed abstract statistical summaries instead of the data's intrinsic shape, its explanations degenerate into shallow echoes of its inputs, failing to provide the qualitative, pattern-level insights that domain experts require to understand what truly happened in the data.

Overcoming these failures requires a paradigm shift. Explanations must move beyond statistical paraphrasing toward a native integration of temporal dynamics and linguistic reasoning. This reduces to two core challenges: the **Contextual Grounding Challenge**, which demands interpreting local events in the context of the full series to explain not only what the data looks like but **why** it is abnormal; and the **Representation Alignment Challenge**, which requires bridging the semantic gap between low-level numerical signals and the rich, shape-based concepts underlying human reasoning.

In this paper, we introduce `AXIS`, a framework designed to address these challenges and unlock the explanatory potential of LLMs for TSAD. Our approach rest on two synergistic contributions. First, to establish the necessary semantic foundation, we construct a novel benchmark with pattern-level labels and rich contextual cues, providing the semantic foundation essential for both grounding and alignment. Second, at the core of our framework is a Hint Tuner that systematically tackles both challenges. For contextual grounding, it distills global time-series information into a compact, informative "hint." For representation alignment, it maps this temporal hint into the LLM's native semantic space. This integrated design transforms a frozen, general-purpose LLM into a context-aware diagnostic expert, capable of generating correct and high reasonal quality answers for TSAD, as illustrated in Fig. 2(b). In summary, our main contributions are threefold:

- **A Benchmark for Semantic Explanations:** To bridge the "semantic gap" between raw time series signals and linguistic concepts, we construct the first benchmark dedicated to semantic time series anomaly explanation. This benchmark ensures both anomaly diversity and explanation fidelity, providing a principled testbed for evaluating the semantic explainability of TSAD.
- **A Novel Cross-Modal Alignment Framework:** We present AXIS, a framework that aligns a frozen LLM with time-series dynamics. It conditions the LLM on three synergistic inputs: a symbolic numeric hint for numerical grounding, a context-integrated step-aligned hint for fine-grained dynamics, and a task-prior hint for global task priors.
- **Extensive Empirical Validation:** Comprehensive experiments show that AXIS establishes a new state of the art in semantic anomaly explanation, substantially outperforming strong baselines including general LLM, specialized time-series LLM, time series VLM.

2 Related Work

State-of-the-Art TSAD Models and Interpretability Challenges Classical and statistical methods remain competitive baselines yet produce pointwise scores with weak semantics and limited support for multivariate structure (Liu et al., 2008; Yeh et al., 2016). Deep TSAD improves accuracy via reconstruction (Audibert et al., 2020; Su et al., 2019; Zhang et al., 2019), prediction-residual modeling (Tuli et al., 2022), and attention-centric architectures (Xu et al., 2021; Yang et al., 2023; Shen et al., 2020; Lan et al., 2025), with recent work exploring unified/foundation-style formulations (Shentu et al., 2024; Gao et al., 2024) and diffusion-based detectors (Wang et al., 2025b). Explanations, however, are largely post hoc and tied to low-level contributions (e.g., time-step or feature importance), limiting mechanism-oriented diagnosis. Time-series XAI extends attribution (Bento et al., 2021) and investigates prototype/shapelet/motif views and counterfactual recourse (Bahri et al., 2022; Yeh et al., 2016), but explanations remain grounded in signal-level statistics rather than pattern-level concepts. This motivates treating TSAD explainability as a semantic alignment problem.

Large Language Models for TSAD LLMs can function as zero-shot anomaly detectors under appropriate prompting and input scaling (Alnegheimish et al., 2024; Dong et al., 2024; Zhou & Yu, 2024; Wang et al., 2025a). Performance degrades on long-horizon series due to context-length limits, lossy time-to-text serialization, and chunked inference, which together induce memory decay and boundary artifacts. A complementary line uses LLMs as post-hoc reasoners that verbalize anomaly scores, SHAP attributions, or raw subsequences, or coordinate multi-agent annotation (Liu et al., 2025; Lin et al., 2024). In both paradigms, LLMs act mainly as summarizers of low-level signals, yielding descriptive rather than semantically grounded explanations. Our approach directly aligns temporal representations with language via soft-prompt-based conditioning, aiming for faithful, pattern-level explanations.

Benchmarks for TSAD Existing benchmarks for time-series question answering, which are adjacent to our task, can be broadly categorized into two paradigms. The first relies on fully synthetic data generation, where normal time series are composed from trends, seasonality, and noise, after which localized anomalies are injected to generate templated or LLM-augmented labels (Cai et al., 2024; Xie et al., 2024; Wang et al., 2025a; Kong et al., 2025). The second paradigm uses real-world datasets, pairing authentic time-series data with corresponding semantic information to create evaluation suites (Kim et al., 2024; Cai et al., 2025; Liu et al., 2024a; Williams et al., 2024; Chen et al., 2025). However, synthetic benchmarks often lack the contextual richness required for robust grounding and representation alignment, while real-world data yields domain-specific explanations that limit model generalizability. To our knowledge, a dedicated benchmark for semantic time series anomaly explanation has remained a critical gap, which our work directly addresses by introducing a benchmark designed for this task.

3 Methodology

This section presents our AXIS framework for the semantic anomaly explanation task. We begin by formalizing the task in Sec. 3.1. Next, we introduce the core architecture in Sec. 3.2. Sec. 3.3

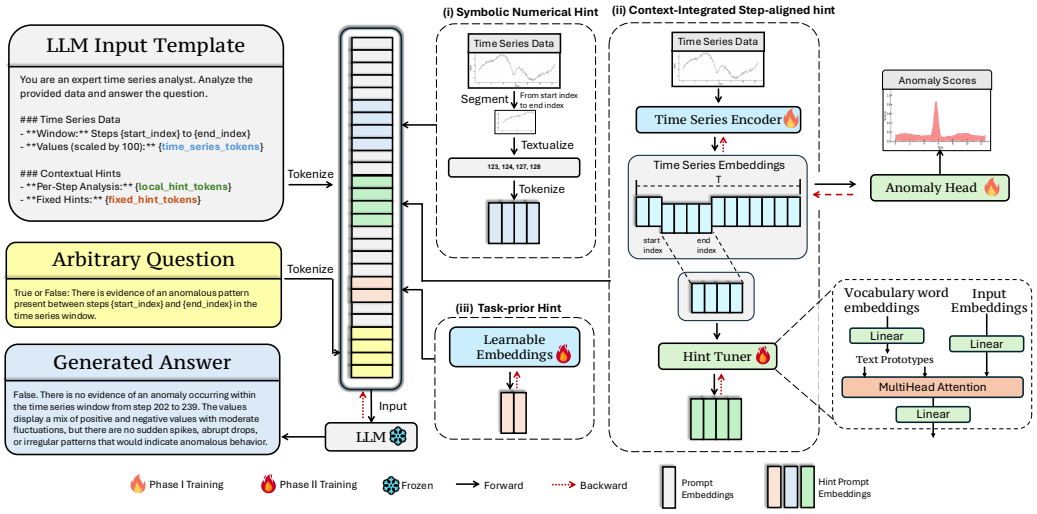

Figure 3: AXIS constructs the prompt by three representation pathways: (i) symbolic numeric grounding via window values, (ii) context-integrated local dynamics through step-aligned hints to capture contextual information, and (iii) task-prior hints encoding global priors.

describes the two-phase training paradigm—encoder pretraining followed by hint tuning with the LLM frozen—and the inference procedure. Finally, to enable systematic supervision, we synthesize a benchmark with pattern-level annotations in Sec. 3.4.

3.1 Problem Formulation

Conventional TSAD methods typically output point-wise anomaly scores for a series of length T, but such signals rarely provide human-understandable insights. In practice, anomalies often span contiguous intervals rather than isolated timestamps, and users are chiefly concerned with understanding **why** an interval is anomalous. To address this, we reformulate the task by introducing a target interval (s, e) and defining the goal as generating a natural-language explanation for it. This window-based formulation respects the temporal continuity of anomalies and makes the explanation task well-posed by localizing reasoning to a specific region within the series. We formalize the problem as follows:

Semantic time series anomaly explanation

Given a univariate time series $\mathbf{x}_{1:T} \in \mathbb{R}^T$ and a natural-language query q, the objective is to explain the pattern within an interval $[s, e)$; in our setup, (s, e) is provided as input. The model learns a mapping $\mathcal{G}$ that, while conditioning on the entire series $\mathbf{x}_{1:T}$ to leverage global context, generates an explanation $\mathbf{y}$ for the target window:

$$\mathcal{G} : (\mathbf{x}_{1:T}, q, s, e) \mapsto \mathbf{y}.$$

3.2 AXIS Framework

We now propose our novel framework called AXIS for semantic anomaly explanation task. AXIS conditions a frozen LLM through three representation pathways: symbolic numeric hint, context-integrated step-aligned hint, and task-prior hint. The overall framework is shown in Fig. 3. We instantiate this conditioning through three pathways that jointly provide numeric grounding, step-aligned dynamics under global context, and compact task-level priors, without expanding the context length or modifying the LLM.

Symbolic Numeric Hint. LLMs possess native reasoning capabilities over discrete numerals when presented symbolically, even in zero-shot settings. To exploit this capability without exhaust-

ing the context budget, we textualize only the target window $[s, e)$ after Z-score normalization of the full series $\mathbf{x}_{1:T}$. Values are scaled by a factor r (default r=100) to preserve precision while avoiding decimal tokens (Liu et al., 2024b), rounded to integers, and serialized as a delimiter-separated string (e.g., "123, 124, 127, 128") to constitute {`time_series_tokens`}. This pathway is compact—its position cost scales as $\alpha(e-s)+c$ where α is the average subword tokens per integer and c the delimiter overhead—yet it preserves step-wise numeric grounding.

Context-Integrated Step-aligned Hint. While the above textualization provides direct numeric access, it cannot capture long-range dependencies essential for TSAD such as regime shifts, seasonality interactions, and boundary effects. We therefore condense global information into step-aligned local representations via a pretrained time-series encoder and a *Hint Tuner*, in the spirit of (Jin et al., 2023). A Transformer encoder f_θ consumes $\mathbf{x}_{1:T}$ and outputs embeddings $\mathbf{H}_{1:T} \in \mathbb{R}^{T\times d_{\text{proj}}}$ where $d_{\text{proj}} = 256$ is the projection dimension (the details are shown in Appx. B.1). We slice $\mathbf{H}_{s:e}$ and map it into the LLM space using a multi-head cross-attention mechanism over a prototype bank derived from the LLM vocabulary. The prototype bank is defined as $\mathbf{S}_{\text{proto}} = \mathbf{M}\mathbf{E}_{\text{vocab}} \in \mathbb{R}^{P\times d_h}$, where $\mathbf{E}_{\text{vocab}} \in \mathbb{R}^{|\mathcal{V}|\times d_h}$ denotes fixed word embeddings and $\mathbf{M} \in \mathbb{R}^{P\times|\mathcal{V}|}$ is a learnable linear projection ($P = 1024$). Specifically, we employ scaled dot-product attention with $H_{\text{tuner}} = 8$ heads. The queries, keys, and values are computed as: $\mathbf{Q} = \mathbf{H}_{s:e}\mathbf{W}_q, \quad \mathbf{K} = \mathbf{S}_{\text{proto}}, \quad \mathbf{V} = \mathbf{S}_{\text{proto}}$, where $\mathbf{W}_q \in \mathbb{R}^{d_{\text{proj}}\times d_h}$ projects the encoder output to the LLM hidden dimension ($d_h = 4096$). The attention output is then:

$$\tilde{\mathbf{H}}_{s:e} = \text{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{(e-s)\times d_h}.$$

The resulting $\tilde{\mathbf{H}}_{s:e}$ acts as step-aligned local hints that inject global context and temporal structure into the LLM embedding space while keeping both the LLM and f_θ frozen. This pathway adds $(e-s)$ positions linearly while supplying detailed global context and temporal alignment.

Task-Prior Hint. To regularize decoding and inject task-level priors that remain stable across instances, we introduce a small set of shared queries $\mathbf{P}_{\text{fix}} \in \mathbb{R}^{K\times d_h}$ that attend to the same prototype source: $\tilde{\mathbf{F}} = \text{Attn}\big(\mathbf{P}_{\text{fix}}, \mathbf{S}_{\text{proto}}, \mathbf{S}_{\text{proto}}\big) \in \mathbb{R}^{K\times d_h}$.

Final Prompt. The three hint pathways are integrated into a unified prompt that conditions the frozen LLM, as illustrated in Fig. 3. The final input sequence is constructed from a template containing the user's query q, the textualized window values, and special placeholder tokens. At input time, the embeddings for the K task-prior hints ($\tilde{\mathbf{F}}$) and the $(e-s)$ step-aligned hints ($\tilde{\mathbf{H}}_{s:e}$) replace the embeddings of $K+(e-s)$ placeholder tokens. The symbolic numeric hint is inserted directly as text. This process yields a single, coherent input sequence for the LLM that combines natural language with rich, multi-faceted temporal information, all without requiring architectural changes to the base model.

3.3 Training Objective

Our method is trained in two phases. First, we pretrain the time-series encoder f_θ using a joint objective that combines masked reconstruction and anomaly classification to learn robust temporal representations. The total loss is defined as:

$$\mathcal{L}_{\text{pretrain}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{class}}\mathcal{L}_{\text{class}},$$

where $\mathcal{L}_{\text{recon}}$ is the Mean Squared Error (MSE) over masked timesteps, $\mathcal{L}_{\text{class}}$ is the Binary Cross-Entropy (BCE) loss, and we set $\lambda_{\text{recon}} = 1.0$ and $\lambda_{\text{class}} = 1.0$.

In the second phase, we freeze both the encoder and the LLM, training only the Hint Tuner and its associated parameters ($\mathbf{M}$, $\mathbf{P}_{\text{fix}}$). This phase optimizes a standard next-token prediction objective:

$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{N} \log P\big(y_i \mid y_{<i}, q, \tilde{\mathbf{H}}_{s:e}, \tilde{\mathbf{F}}\big),$$

where $\mathbf{y}$ is the target explanation sequence. This two-phase strategy maintains computational efficiency by training only the lightweight Hint Tuner while preserving the time series encoder's pretrained capabilities. The decoupled design enables stable training by separating temporal representation learning from cross-modal alignment. The detailed training process and additional hyperparameters are given in Appx. B.2 and B.3 .

3.4 A BENCHMARK FOR SEMANTIC TIME SERIES ANOMALY EXPLANATION

Existing methods reveal a foundational limitation: the community lacks a benchmark that teaches models to speak the language of temporal patterns. To address both the contextual grounding and representation alignment challenges outlined earlier, we synthesize a benchmark specifically designed to train models to reason about anomalies like human experts. Rather than a mere collection of time series, our benchmark constitutes a carefully curated curriculum built around three core design principles.

Pattern-Level Anomaly Vocabulary. To address the representation alignment challenge, we introduce a procedural engine that moves beyond abstract statistical deviations to a vocabulary of interpretable, pattern-level anomalies. As illustrated in Figure 4, our engine synthetically composes canonical anomaly primitives—such as *sudden spikes*, *level shifts*, and *periodicity breaks*—onto clean baseline series. A key advantage of our approach is the generation of **paired time series**: for every abnormal series created, a corresponding normal counterpart is preserved. This methodology establishes an unambiguous, verifiable link between anomaly time series and its linguistic label, forming the bedrock for teaching models to reason about the semantics of temporal events.

Contextual and Comparative Reasoning. To overcome the contextual grounding challenge, we designed our benchmark to compel models to reason about local events within a global and comparative framework. Naively presenting isolated time-series windows is insufficient. Instead, our engine first generates a **global descriptor**, a textual summary of the series' overall dynamics (e.g., trends, seasonality), which provides essential context. Second, we employ a comparative windowing strategy. A model is presented not only with a window containing a potential anomaly but also with the corresponding temporal window from its "healthy" paired series. This core design choice is a significant advantage, as it inherently frames the task as a discriminative one: the model must learn to articulate **why** a specific pattern deviates from an explicit, provided norm, rather than merely describing a segment in isolation.

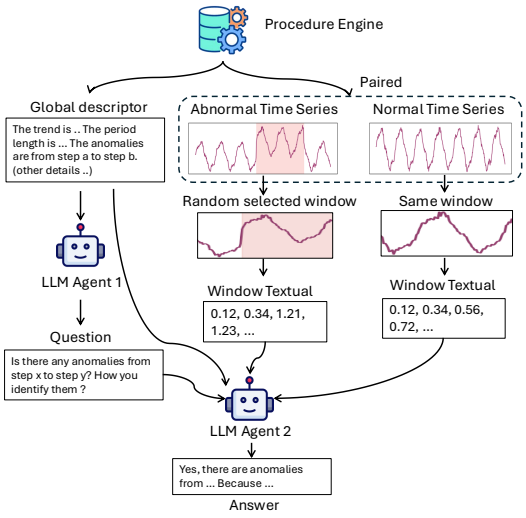

Figure 4: The architecture of our procedural engine for generating context-aware and comparative anomaly explanation benchmarks.

LLM-Powered Explanation Generation. Building on this structured foundation, we leverage LLMs to generate high-quality, multi-format supervision signals. As depicted in our pipeline, this is a multi-agent process. One LLM agent uses the global descriptor to formulate a targeted diagnostic question. A second, more powerful agent is then tasked with answering this question, conditioned on the global descriptor, the abnormal and normal window data. The primary motivation here is to generate rich, conceptual explanations. Our prompts are meticulously engineered to discourage superficial strategies (e.g., quoting raw values) and instead elicit reasoning based on the intrinsic, morphological characteristics of the anomaly. This process yields a diverse and consistent set of questions and detailed rationales, creating a powerful supervisory signal for training.

Ensuring Benchmark Integrity. To guarantee the scientific utility and integrity of our benchmark, we implement a rigorous quality control pipeline. This process verifies the agreement between ground-truth labels and generated answers, enforces stylistic consistency, and filters potential redundancies. We provide comprehensive dataset statistics and are releasing all generation metadata to ensure the full reproducibility of our benchmark. The detailed integrity check protocols are described in Appx. D. The result is not merely a dataset but a robust training environment engineered to finally bridge the semantic gap in time series anomaly explanation. Some examples of the generated Q&A pairs are given in Appx. C.2.

4 EXPERIMENTS

To validate the effectiveness of AXIS, we conduct a series of experiments designed to answer three central research questions: **RQ1: Explanation Quality.** How does AXIS compare against state-of-the-art LLM-based methods in generating high-quality, semantic anomaly explanations? **RQ2: Component Importance.** How do the core components of our framework—the symbolic numeric hint, the context-integrated step-aligned hint, and the task-prior hint—contribute to the final explanation quality? **RQ3: Architectural Generality.** How robust is the AXIS framework when applied to different underlying frozen LLMs? **RQ4: Generalization.** How well does AXIS generalize to external public datasets representing real-world scenarios? Finally, in Appx. B.2, we demonstrate that our Phase I TSAD model achieves results comparable to state-of-the-art methods on real-world public TSAD datasets.

4.1 EXPERIMENTAL SETUP

Dataset. All experiments are conducted on our newly created **Semantic Anomaly Benchmark** (detailed in Sec. 3.4 and Appx. F). This benchmark is specifically designed for the task of semantic time series anomaly explanation, containing diverse anomaly patterns, multi-format questions (Multiple Choice, True/False, Open-Ended), and detailed, pattern-level ground-truth explanations. The all hyperparameters for AXIS is given in Appx. B.3.

Baselines. We compare AXIS with a comprehensive set of strong baselines, categorized as follows: **Timeseries VLM:**(He et al., 2025) `Image LLM` is supported by gpt-4o, which analyzes plots of the full time series with highlighted window, treating the explanation task as a visual reasoning problem. **Specialized TS-LLM Methods:** We include several recent models designed for time series analysis with LLMs: `ChatTS` (Xie et al., 2024), `LLMAD` (Liu et al., 2025), `ChatTime` (Wang et al., 2025a), and `AnomLLM` (Dong et al., 2024). We evaluate `AnomLLM` in two settings: providing the full series (`AnomLLM(Full)`) and providing only the target window (`AnomLLM(Window)`). **Heuristic Baselines:** To contextualize benchmark difficulty, we introduce two non-learning baselines: (i) `Random Template`, which fills sophisticated-sounding templates with random values to test if the task is solvable by "hallucination"; and (ii) `Simple Heuristic`, a rule-based system using statistical thresholds (z-score, volatility) to flag anomalies.

Evaluation Metrics. Following recent work on evaluating LLM-generated content, we use an LLM-as-a-judge approach, specifically **G-eval** (Liu et al., 2023) with Gemini-2.5 as the arbiter. The quality of explanations is assessed across multiple dimensions tailored to each question type, including Correctness (Corr.), Reasoning Quality (Rsn. Qual.), Accuracy (Acc.), Completeness (Comp.), Relevance (Rel.), and Justification Quality (Justif.). A final, holistic score (Final) is also computed. The detailed definition for evaluation metrics is given in Appx. G

4.2 MAIN RESULTS: EXPLANATION QUALITY (RQ1)

Table 1 presents the main results comparing our model, AXIS, against all baselines. Our method demonstrates superior performance across all metrics and question types, establishing a new state-of-the-art for the task.

Specifically, AXIS achieves the highest final scores on Multiple Choice (4.19), Open-Ended (3.02), and True/False (3.65) questions. This consistent top-ranking performance highlights its robust ability to generate accurate, complete, and well-reasoned explanations regardless of the question format. Compared to specialized TS-LLM baselines like `ChatTS` and the `AnomLLM` variants, our method shows a significant improvement, underscoring the effectiveness of our proposed hint-based conditioning strategy. The strong performance against the `Image LLM` baseline further suggests that our multi-pathway representation provides richer, more aligned signals for the LLM than raw visual serialization.

Visualization. To qualitatively illustrate these performance gains, Fig. 5 presents a comparative case study. In the Fig. 5(a), the target window (steps 444 to 473) exhibits pronounced oscillations. AXIS correctly contextualizes these dynamics against the broader series, identifying them as part of a normal periodic pattern and concluding there is no anomaly. In stark contrast, a baseline like

Table 1: Main Results: AXIS vs Baselines

Model	Multiple Choice			Open Ended				True False		
	Final	Corr.	Rsn. Qual.	Final	Acc.	Comp.	Rel.	Final	Corr.	Justif.
AXIS	**4.19**	**4.21**	**4.14**	**3.02**	**2.87**	**2.93**	**3.31**	**3.65**	**3.60**	**3.74**
Image LLM	4.09	4.12	4.02	2.68	2.53	2.49	3.07	2.64	2.57	2.74
ChatTS	3.29	3.40	3.05	2.19	1.67	2.13	2.87	2.79	2.76	2.83
LLMAD	2.73	2.70	2.79	2.09	2.09	1.89	2.31	2.49	2.52	2.43
ChatTime	1.33	1.49	0.98	0.96	0.95	0.98	0.95	1.04	1.07	1.00
AnomLLM(Full)	3.13	2.98	3.49	2.86	2.53	2.89	3.20	2.88	2.60	3.31
AnomLLM(Window)	3.78	3.81	3.70	2.84	2.78	2.55	3.24	3.32	3.45	3.12
Baseline 1 (Random)	1.02	1.03	1.00	1.21	1.21	1.05	1.41	1.29	1.36	1.18
Baseline 2 (Heuristic)	2.44	2.81	1.58	1.72	1.98	1.15	2.07	2.64	2.74	2.50

Table 2: Ablation Studies of Hint Components

Model	Multiple Choice			Open Ended				True False		
	Final	Corr.	Rsn. Qual.	Final	Acc.	Comp.	Rel.	Final	Corr.	Justif.
AXIS	**4.19**	**4.21**	**4.14**	**3.02**	**2.87**	**2.93**	**3.31**	**3.65**	**3.60**	**3.74**
w/o-task-hint	3.82	3.93	3.56	2.33	2.13	2.22	2.69	3.25	3.31	3.17
w/o-context-hint	4.09	4.16	3.91	2.75	2.56	2.58	3.16	2.44	2.48	2.38
w/o-windows	3.95	4.00	3.84	2.41	2.00	2.36	2.95	2.87	2.83	2.93

AnomLLM or ChatTS, when limited to the window view, lacks this broader context and erroneously flags the internal deviations as potential outliers. In Fig. 5(b), AXIS provides a precise characterization by explicitly identifying the brief increase at steps 6–7 (2.27, 1.73, 2.38) and correctly interpreting it as a transient fluctuation rather than a sustained anomaly, concluding that the pattern is stable and anomaly-free. In contrast, alternative approaches tend to give vague or generalized descriptions, often noting fluctuations or moderate oscillations without distinguishing whether they indicate normal behavior or anomalies. This comparison highlights that our representation alignment framework enables fine-grained, context-aware interpretation of time series behavior, avoiding ambiguous assessments and ensuring robust anomaly detection.

Human evaluation. To further validate these quantitative results from a human-centric perspective, we also performed a statistical analysis based on expert rankings of the model-generated explanations. We conducted a survey where human evaluators were asked to rank the outputs from all competing models for each question type. Fig. 6 corroborates the findings from Table 1. Note that `ChatTime` is excluded from this analysis as it exhibited fundamental task-comprehension failures (e.g., inability to parse prompts), rendering its outputs non-comparable. Our model, AXIS, consistently achieves the best (lowest) average rank across all three tasks, positioning it furthest to the right. The details for experiment setup and results are shown in Appx. H.

Validity Check via Heuristic Baselines. As shown in Table 1, `Baseline 1 (Random)` performs near the lower bound across all metrics (1.00–1.41), confirming that fabricated explanations fail. `Baseline 2 (Heuristic)` shows modest improvement but still lags significantly behind semantic models, particularly on reasoning quality (1.58 vs 4.14 for AXIS), demonstrating that rule-based systems cannot generate coherent explanations. The substantial gap between these baselines and AXIS validates that the benchmark effectively discriminates between trivial heuristics and genuine semantic reasoning.

4.3 Ablation Studies (RQ2)

To understand the contribution of each component in AXIS, we conducted an ablation study, with results summarized in Table 2. The study confirms that all three input pathways are crucial for optimal performance. Removing the context-integrated step-aligned hint ('w/o-context-hint') degrades performance most severely on True/False questions (Final score drops from 3.65 to 2.44), confirming that binary judgments rely on precise, step-aligned temporal cues. In contrast, Open-Ended questions are most sensitive to the symbolic numeric hint from the window ('w/o-windows'); without it, accuracy drops markedly (2.00 vs 2.87) and overall quality declines (Final 2.41 vs 3.02), indicating

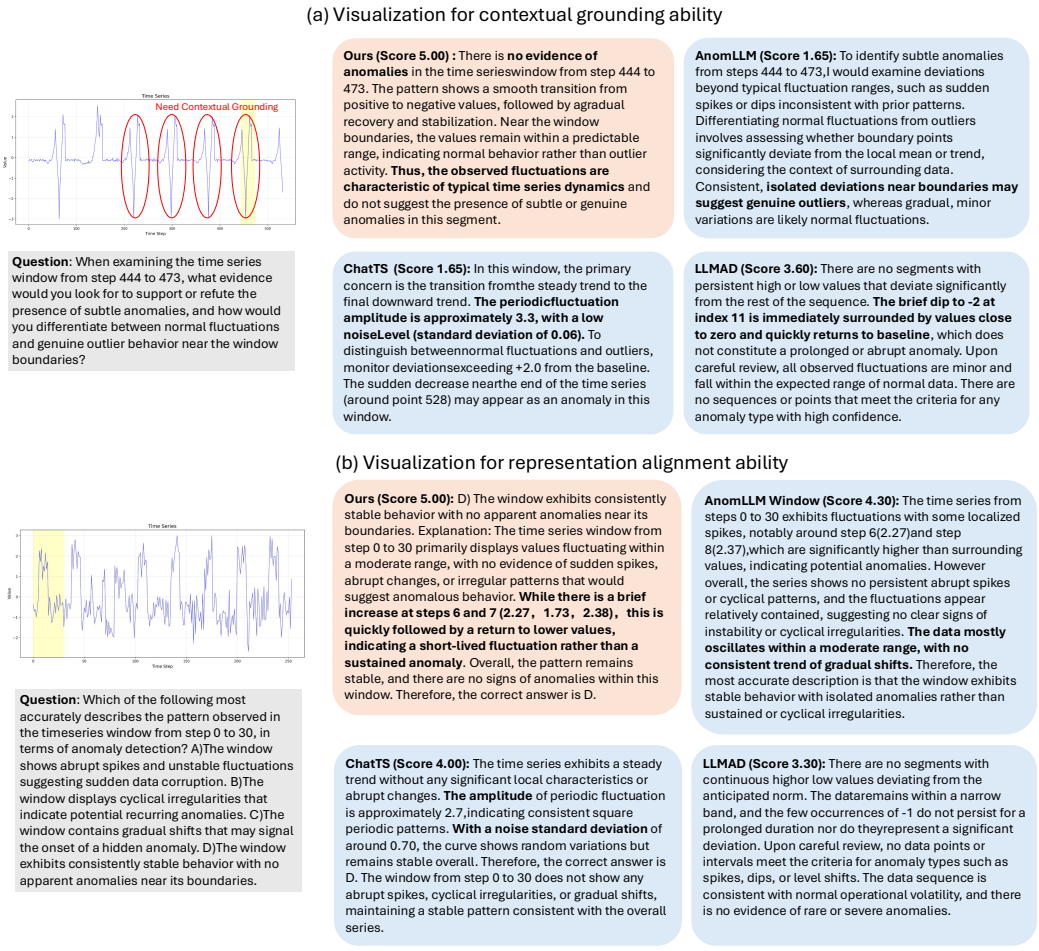

Figure 5: Visualization of (a) contextual grounding and (b) representation alignment ability

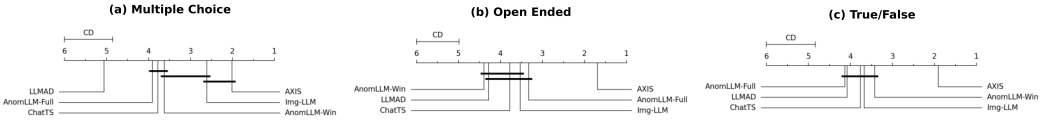

Figure 6: Critical Difference diagrams illustrating the statistical comparison of model performance based on human rankings for (a) Multiple Choice, (b) Open-Ended, and (c) True/False questions.

that direct numeric access provides the fine-grained grounding needed for detailed answers. Eliminating the task-prior hint ('w/o-task-hint') also substantially harms Open-Ended completeness and relevance (2.22/2.69 vs 2.93/3.31), suggesting that these global priors help structure the explanation and ensure comprehensive coverage. The full model consistently yields the best scores, validating our design choices. Additional experiments for causal contribution of hints are given in Appx. I.

4.4 Analysis of Architectural Variants (RQ3)

To assess its generality, we instantiated AXIS across multiple LLM families and settings, using a standardized *Family + Size + Variant* naming scheme and a fixed data schedule (R1) to isolate architectural effects. As shown in Table 3, our framework demonstrates robust cross-family adaptation (Qwen, Llama, Mistral) and its performance scales with model size (e.g., Qwen-14B $>$ 7B $>$ 1.5B). The results also reveal complementary strengths among model variants: code-pretrained models like *Qwen2.5-7B Coder* excel at structured discrimination tasks, achieving the highest Multiple Choice score (4.40), whereas instruction-tuned versions such as *Qwen2.5-7B Instruct* lead in free-form explanatory quality, with top scores in Open-Ended relevance and completeness (3.50/3.00). This

Table 3: AXIS variants across LLM families and settings (standardized naming: family + size + variant; *Instruct* denotes instruction-tuned, *Coder* denotes code-pretrained

		Multiple Choice			Open Ended				True False		
Family	Variant	Final	Corr.	Rsn. Qual.	Final	Acc.	Comp.	Rel.	Final	Corr.	Justif.
Deepseek-Llama	8B (Instruct)	4.28	4.30	4.23	3.02	2.84	2.84	3.45	3.64	3.55	**3.79**
Deepseek-Qwen	14B (Instruct)	4.31	4.28	4.37	3.03	2.80	2.93	3.42	3.60	3.55	3.69
Deepseek-Qwen	7B (Instruct)	4.19	4.21	4.14	3.02	**2.87**	2.93	3.31	3.65	3.60	3.74
Deepseek-Qwen	1.5B (Instruct)	4.07	4.12	3.95	2.72	2.65	2.55	3.00	3.18	3.17	3.19
Qwen2.5	7B (Coder)	**4.40**	**4.40**	**4.42**	2.72	2.64	2.58	2.98	2.96	2.98	2.93
Qwen2.5	7B (Base)	4.30	4.37	4.12	2.75	2.73	2.45	3.13	**3.66**	**3.69**	3.62
Qwen2.5	7B (Instruct)	4.17	4.22	4.06	**3.08**	2.80	**3.00**	**3.50**	3.11	3.00	3.27
Mistral	7B (Base)	2.97	3.05	2.79	2.89	2.69	2.82	3.20	2.69	2.55	2.90
Mistral	7B (Instruct)	3.36	3.33	3.44	2.77	2.58	2.45	3.35	3.27	3.17	3.43

Table 4: Performance comparison on Multiple Choice (MC) and True/False (TF) tasks.

Dataset	**Metric**	**AnomLLM**		**LLMAD**	**ChatTS**	**AXIS 14B**	**Image LLM**
		Window	**Full**				
YAHOO (W=0.651)	MC Score	2.78	1.69	2.48	2.83	**3.07**	2.80
	MC Acc.	0.48	0.18	0.47	0.52	**0.55**	0.52
	TF Score	2.62	1.90	2.54	2.13	**3.28**	2.29
	TF Acc.	0.53	0.38	0.51	0.45	**0.67**	0.52
TODS (W=0.672)	MC Score	2.07	1.77	2.33	2.67	**2.90**	2.53
	MC Acc.	0.37	0.30	0.33	0.47	**0.53**	0.47
	TF Score	2.67	2.53	2.87	2.53	**3.07**	2.27
	TF Acc.	0.53	0.52	0.58	0.53	**0.62**	0.50
NEK (W=0.713)	MC Score	3.31	1.94	1.44	3.06	3.25	**3.41**
	MC Acc.	**0.63**	0.31	0.13	0.56	**0.63**	**0.63**
	TF Score	2.88	1.69	2.94	2.25	**3.69**	2.25
	TF Acc.	0.56	0.35	0.59	0.47	**0.74**	0.50

highlights that AXIS not only universally enhances different base models but also allows for trade-offs between discriminative and explanatory objectives through strategic variant selection.

4.5 Generalization to Public Real-world Datasets (RQ4)

To rigorously evaluate the generalization capability of AXIS beyond our synthetic benchmark, we extend our evaluation to established public datasets representing real-world operational scenarios. We construct a curated evaluation set derived from 108 time series sampled from three distinct sources: YAHOO, TODS, and NEK (details in Appendix H.3). We formulate two specific tasks—True/False and Multiple-Choice Questions—and employed expert annotation to establish high-quality ground truth, ensuring a robust testbed for real-world explanation quality.

The results, summarized in Table 4, demonstrate that AXIS consistently delivers superior explanation quality across these diverse domains. Our model achieves the highest scores in both MC and TF categories on the YAHOO and TODS datasets, and leads in TF performance on NEK. Notably, the strong inter-rater agreement (Kendall's $W > 0.65$) across all datasets confirms the reliability of our human evaluation. These findings validate that AXIS effectively generalizes to real-world data distributions, bridging the gap between synthetic training and practical operational deployment.

5 Conclusion

We introduce a novel cross-modal framework that effectively adapts frozen Large Language Models for semantic time series anomaly explanation. By using a three-stream conditioning strategy that combines a symbolic numeric hint, a context-integrated step-aligned hint, and a task-prior hint, our method achieves strong performance in both detection accuracy and explanation quality. Future work will explore incorporating domain-specific knowledge graphs to enhance causal reasoning and generating multi-modal explanations that include visualizations alongside text.

ETHICS STATEMENT

This work focuses on explainable time series anomaly detection and does not involve personally identifiable information or other sensitive attributes. Our benchmark primarily uses procedurally synthesized data and publicly available datasets; no private logs are collected, and no re-identification is attempted. Human evaluation was conducted with informed consent, anonymized responses, and fair compensation in line with institutional guidelines, without storing any personally identifying information.

REPRODUCITY STATEMENT

We aim for full reproducibility. Upon publication, we will release code, configuration files, and scripts to reproduce: (i) the benchmark synthesis pipeline (including prompts, fixed random seeds, and parameter settings); (ii) Phase I encoder pretraining and Phase II hint tuning with exact hyperparameters, token budgets, and training schedules (see Appendix B.2 and Appendix B.3); and (iii) evaluation pipelines, including baseline configurations, G-Eval judge prompts, scoring scripts, and human-evaluation materials. We will provide model checkpoints where licensing permits or otherwise specify exact model identifiers and initialization procedures. The code is available in `https://anonymous.4open.science/r/TimeSemantic-1742/main.py`

REFERENCES

Sarah Alnegheimish, Linh Nguyen, Laure Berti-Equille, and Kalyan Veeramachaneni. Can large language models be anomaly detectors for time series? In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. IEEE, 2024.

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

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3395–3404, 2020.

Omar Bahri, Soukaina Filali Boubrahimi, and Shah Muhammad Hamdi. Shapelet-based counterfactual explanations for multivariate time series. *arXiv preprint arXiv:2208.10462*, 2022.

João Bento, Pedro Saleiro, André F Cruz, Mário AT Figueiredo, and Pedro Bizarro. Timeshap: Explaining recurrent models through sequence perturbations. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2565–2573, 2021.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

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

Yifu Cai, Xinyu Li, Mononito Goswami, Michał Wiliński, Gus Welter, and Artur Dubrawski. Timeseriesgym: A scalable benchmark for (time series) machine learning engineering agents. *arXiv preprint arXiv:2505.13291*, 2025.

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

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

Manqing Dong, Hao Huang, and Longbing Cao. Can llms serve as time series anomaly detectors? *arXiv preprint arXiv:2408.03475*, 2024.

Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj Dayama, Wesley M Gifford, and Jayant Kalagnanam. Tspulse: Dual space tiny pre-trained models for rapid time-series analysis. *arXiv preprint arXiv:2505.13033*, 2025.

Shanghua Gao, Teddy Koker, Owen Queen, Tom Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Zelin He, Sarah Alnegheimish, and Matthew Reimherr. Harnessing vision-language models for time series anomaly detection. *arXiv preprint arXiv:2506.06836*, 2025.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Rahat Iqbal, Tomasz Maniak, Faiyaz Doctor, and Charalampos Karyotis. Fault detection and isolation in industrial processes using deep learning approaches. *IEEE Transactions on Industrial Informatics*, 15(5):3077–3084, 2019.

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

Kai Kim, Howard Tsai, Rajat Sen, Abhimanyu Das, Zihao Zhou, Abhishek Tanpure, Mathew Luo, and Rose Yu. Multi-modal forecaster: Jointly predicting time series and textual data. *arXiv preprint arXiv:2411.06735*, 2024.

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

Tian Lan, Yifei Gao, Yimeng Lu, and Chen Zhang. Cicada: Cross-domain interpretable coding for anomaly detection and adaptation in multivariate time series. *arXiv preprint arXiv:2505.00415*, 2025.

Minhua Lin, Zhengzhang Chen, Yanchi Liu, Xujiang Zhao, Zongyu Wu, Junxiang Wang, Xiang Zhang, Suhang Wang, and Haifeng Chen. Decoding time series with llms: A multi-agent framework for cross-domain annotation. *arXiv preprint arXiv:2410.17462*, 2024.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Prabhakar Kamarthi, Aditya Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. Time-mmd: Multi-domain multimodal dataset for time series analysis. *Advances in Neural Information Processing Systems*, 37:77888–77933, 2024a.

Jun Liu, Chaoyun Zhang, Jiaxu Qian, Minghua Ma, Si Qin, Chetan Bansal, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Large language models can deliver accurate and interpretable time series anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 4623–4634, 2025.

Qingxiang Liu, Xu Liu, Chenghao Liu, Qingsong Wen, and Yuxuan Liang. Time-ffm: Towards lm-empowered federated foundation model for time series forecasting. *Advances in Neural Information Processing Systems*, 37:94512–94538, 2024b.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in neural information processing systems*, 33:13016–13026, 2020.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. *arXiv preprint arXiv:2405.15273*, 2024.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6): 1201–1214, 2022.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025a.

Tao Wang, Cong Zhang, Xingguang Qu, Kun Li, Weiwei Liu, and Chang Huang. Diffad: A unified diffusion modeling approach for autonomous driving. *arXiv preprint arXiv:2503.12170*, 2025b.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, et al. Context is key: A benchmark for forecasting with essential textual information. *arXiv preprint arXiv:2410.18959*, 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.

Shi Xiaoming, Wang Shiyu, Nie Yuqi, Li Dianqi, Ye Zhou, Wen Qingsong, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *ICLR 2025: The Thirteenth International Conference on Learning Representations*. International Conference on Learning Representations, 2025.

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

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3033–3045, 2023.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1317–1322. Ieee, 2016.

Vannel Zeufack, Donghyun Kim, Daehee Seo, and Ahyoung Lee. An unsupervised anomaly detection framework for detecting anomalies in real time through network system's log files analysis. *High-Confidence Computing*, 1(2):100030, 2021.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1409–1416, 2019.

Zihao Zhou and Rose Yu. Can llms understand time series anomalies? *arXiv preprint arXiv:2410.05440*, 2024.

A USE OF LLMS

Our manuscript preparation involved the use of large language models (LLMs) primarily for refining language, improving grammar, readability, and stylistic elements. Crucially, LLMs also constituted a core component of our research process. They were leveraged as agents to participate in data generation, established as benchmarks for experimental baselines, and contributed to the comprehensive evaluation of experimental results. For a detailed account of LLMs' specific applications within our methodology, please refer to Section 3.4 and Section 4. We confirm that the contributions of LLMs, whether in writing or research, do not impede the reproducibility of our reported findings.

B DETAILS FOR AXIS

B.1 TIME-SERIES ENCODER DETAILS

This section details the architecture of the time-series encoder f_θ used in AXIS. The overall structure is shown in Fig. 7.

Patchify. Let P denote the patch size. We pad $\mathbf{x}_{1:T}$ to length $T' = \lceil T/P \rceil P$ with zeros and form $N = T'/P$ contiguous, non-overlapping patches:

$$\mathbf{X}^{(n)} = [x_{(n-1)P+1}, \ldots, x_{nP}] \in \mathbb{R}^P, \quad n = 1, \ldots, N.$$

Stacking yields $\mathbf{X}_p \in \mathbb{R}^{N \times P}$. The patch-level attention mask is obtained by aggregating the step mask: a patch is valid if it contains at least one valid step,

$$\mathbf{m}_n^{\text{patch}} = \mathbb{I}\Big(\sum_{t=(n-1)P+1}^{nP} m_t > 0 \Big), \quad n = 1, \ldots, N.$$

Patch Embedding. Each patch is projected to the model dimension d_{model} with a linear layer

$$\mathbf{Z} = \mathbf{X}_p \mathbf{W}_e + \mathbf{1}\, \mathbf{b}_e^\top \in \mathbb{R}^{N \times d_{\text{model}}}, \quad \mathbf{W}_e \in \mathbb{R}^{P \times d_{\text{model}}}.$$

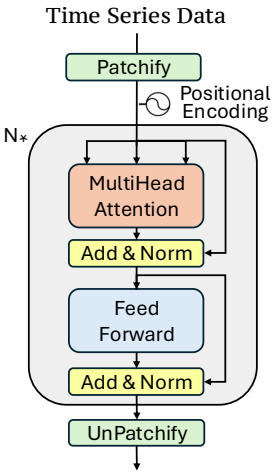

Figure 7: The structure of time-series encoder.

Positional Encoding via RoPE. The encoder uses rotary positional encoding (RoPE) applied *inside* self-attention to the query and key of each head. Concretely, for a sequence of N patch tokens, RoPE generates frequency pairs that rotate the d_{model} channels in 2D subspaces, providing translation-friendly relative position information.

Transformer Encoder (Non-causal). We stack L pre-norm Transformer encoder layers. Each layer comprises: (i) multi-head self-attention with H heads and head dimension $d_{\text{head}} = d_{\text{model}}/H$; (ii) an MLP block of the LLaMA style with gated GELU activation. RMSNorm is used before attention and MLP. Attention is *non-causal* and operates over all N patches, enabling global context aggregation. The attention mask derived from $\mathbf{m}^{\text{patch}}$ prevents padded tokens from contributing. In the univariate case there is no inter-feature bias term; the attention is standard scaled dot-product with RoPE.

UnPatchify and Projection to Step Alignment. The encoder output $\mathbf{U} \in \mathbb{R}^{N \times d_{\text{model}}}$ is mapped to a patch-level representation of size $P \times d_{\text{proj}}$ through a linear projection

$$\mathbf{Y} = \mathbf{U}\mathbf{W}_p \in \mathbb{R}^{N \times (P\, d_{\text{proj}})}, \quad \mathbf{W}_p \in \mathbb{R}^{d_{\text{model}} \times (P\, d_{\text{proj}})}.$$

Reshaping and reordering yield step-aligned embeddings $\tilde{\mathbf{H}} \in \mathbb{R}^{T' \times d_{\text{proj}}}$. We finally drop the padded tail to obtain

$$\mathbf{H}_{1:T} = \tilde{\mathbf{H}}_{1:T} \in \mathbb{R}^{T \times d_{\text{proj}}}.$$

B.2 Training Process Details

Our model is trained in a two-phase process designed to first build a robust time-series representation and then align it with the frozen LLM's semantic space.

B.2.1 Phase I: Time-Series Encoder Pretraining

In the first phase, we pretrain the time-series encoder f_θ to learn a versatile representation of temporal dynamics. This is achieved by optimizing a joint objective function that combines masked reconstruction and anomaly classification. The encoder is a Transformer-based architecture that processes the entire input time series $\mathbf{x}_{1:T}$.

Masked Reconstruction. Following the principles of self-supervised learning for sequential data, we employ a masked reconstruction objective. A portion of the input time series is randomly masked, and the encoder is tasked with reconstructing the masked values. Let $\mathbf{x}$ be the input series and $\mathbf{m}$ be a binary mask where $m_i = 1$ if the i-th timestep is masked and 0 otherwise. The encoder f_θ, followed by a reconstruction head g_ψ, outputs a reconstructed series

$$\hat{\mathbf{x}} = g_\psi\big(f_\theta(\mathbf{x} \odot (\mathbf{1} - \mathbf{m}))\big).$$

The reconstruction loss is the Mean Squared Error (MSE) over the masked timesteps:

$$\mathcal{L}_{\text{recon}} = \frac{1}{\sum_i m_i} \sum_{i=1}^{T} m_i (x_i - \hat{x}_i)^2.$$

This objective forces the encoder to learn the underlying patterns and dependencies within the time series.

Anomaly Classification. To explicitly teach the encoder to distinguish between normal and anomalous patterns. Given a time series $\mathbf{x}$ and its corresponding binary anomaly label $\mathbf{l} \in \{0, 1\}^T$, the model predicts the probability of an anomaly $\hat{\mathbf{l}}_t = h_\zeta(f_\theta(\mathbf{x}))$. We use the BCE loss:

$$\mathcal{L}_{\text{class}} = -\sum_{t=1}^{T} \big(l_t \log(\hat{l}_t) + (1 - l_t) \log(1 - \hat{l}_t)\big).$$

Joint Objective. The total loss for the pretraining phase is a weighted sum of the reconstruction and classification losses:

$$\mathcal{L}_{\text{pretrain}} = \lambda_{\text{recon}} \mathcal{L}_{\text{recon}} + \lambda_{\text{class}} \mathcal{L}_{\text{class}},$$

where λ_{recon} and λ_{class} are hyperparameters that balance the two objectives. After this phase, the weights of the encoder f_θ are frozen (only f_θ is retained for the next phase).

B.2.2 Phase II: Hint Tuner Training

In the second phase, we focus on aligning the pretrained time-series representations with the LLM. Both the time-series encoder f_θ and the LLM are kept frozen to maintain their powerful, pre-existing capabilities. The only trainable components are the Hint Tuner and its associated parameters, namely the prototype bank selection matrix $\mathbf{M}$ and the learnable task-prior queries $\mathbf{P}_{\text{fix}}$.

The goal of this phase is to train the model to generate a natural language explanation $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ conditioned on the time series $\mathbf{x}$, the query q, and the derived hints. The training objective is a standard causal language modeling loss, which maximizes the likelihood of the ground-truth explanation.

Specifically, the input to the LLM is constructed by embedding the query q, the symbolic-numeric hint, and replacing placeholder token embeddings with the outputs of the Hint Tuner ($\tilde{\mathbf{H}}_{s:e}$ and $\tilde{\mathbf{F}}$). Here, s and e denote the start and end indices of the selected window within the full series $\mathbf{x}_{1:T}$. The model then autoregressively predicts the next token in the explanation sequence. The loss function is the negative log-likelihood of the target sequence:

$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{N} \log P\big(y_i \mid y_{<i}, q, \tilde{\mathbf{H}}_{s:e}, \tilde{\mathbf{F}}\big).$$

By optimizing this objective, the Hint Tuner learns to map temporal features from the frozen encoder into meaningful "soft prompts" that effectively guide the frozen LLM to generate accurate and relevant explanations for the given time-series window. During this phase, gradients update only the Hint Tuner and its associated parameters ($\mathbf{M}$ and $\mathbf{P}_{\text{fix}}$); both f_θ and all LLM parameters remain frozen.

B.3 AXIS Hyperparameter Configuration

This section provides a comprehensive specification of the hyperparameter configuration used in our AXIS framework. The architecture consists of two primary components: the time-series encoder and the hint tuner, each with distinct parameter settings optimized through systematic ablation studies.

Time-Series Encoder Configuration. The time-series encoder f_θ employs a Transformer-based architecture with patch-based tokenization. The patch size P is set to 16, enabling efficient processing of long sequences while preserving temporal granularity. The encoder utilizes $L = 6$ transformer layers, each with a model dimension $d_{\text{model}} = 512$ and $H = 8$ attention heads, resulting in a head dimension $d_{\text{head}} = 64$. The projection dimension d_{proj} is configured to 256.

Hint Tuner Architecture. The Perceiver-based hint tuner serves as the cross-modal alignment module between time-series representations and the frozen LLM embedding space. The prototype bank consists of $P = 1024$ prototype embeddings derived from the LLM vocabulary through a learned linear mapping. The number of fixed task-prior tokens K is set to 8, providing sufficient capacity for global task-level priors while maintaining computational efficiency. The cross-attention mechanism within the hint tuner employs 8 attention heads, matching the encoder configuration for architectural consistency.

Language Model Integration. Our framework supports multiple LLM families with varying parameter scales. The experiments primarily utilize Qwen2.5-7B-Instruct as the base model, with the hidden dimension $d_h = 4096$ for the 7B variant. The vocabulary size varies by model family, typically ranging from 32,000 to 152,000 tokens. Special tokens `<|local_hint|>` and `<|fixed_hint|>` are added to the vocabulary for hint injection, with corresponding token IDs dynamically assigned during initialization.

Training Configuration. The two-phase training procedure employs distinct hyperparameter settings for each phase. During Phase I (encoder pretraining), the reconstruction loss weight λ_{recon} and classification loss weight λ_{class} are both set to 1.0, providing balanced supervision across objectives. The masking ratio for reconstruction is configured to 0.25, following established practices in self-supervised time-series learning. Phase II (hint tuner training) utilizes standard causal language modeling with a learning rate of 2×10^{-4} and weight decay of 0.01, applied exclusively to the trainable parameters of the hint tuner and prototype bank.

Inference Parameters. During generation, the model employs beam search with 5 beams and a repetition penalty of 1.15 to ensure diverse and coherent explanations. The maximum generation length is capped at 1000 tokens to accommodate detailed explanations while preventing excessive verbosity. No-repeat n-gram size is set to 3 to avoid repetitive patterns, and length penalty is maintained at 1.0 to balance explanation completeness with conciseness.

C Prompt Templates and Q&A Exemplars

C.1 Prompt Templates

C.1.1 Question Generation Prompt

```
System: You generate precise and relevant questions for time series
    anomaly detection.

User:
```

```
Generate a {question_type} question focused on anomaly detection for the
    window [ {window_start}, {window_end} ].

Context:
- Task: time series anomaly detection on a windowed segment
- The window {may / does} contain anomalies: {has_anomaly}
- Canonical tag (if available): {canonical_tag}
- Global information: {global_information}

Requirements:
1) Output ONLY the question text (no answers, no explanations).
2) Focus on anomaly identification and pattern analysis within the window
    .
3) Consider boundary effects near window edges.
4) Multiple choice (if applicable):
  - Provide exactly 4 options (A, B, C, D).
  - Options must be mutually exclusive, same style.
  - Include both normal and anomalous descriptions; avoid exact numeric
      values.
5) True/False (if applicable):
  - Make a specific statement about a potential anomaly pattern.
6) Open-ended (if applicable):
  - Ask about pattern evidence and reasoning for/against anomalies.
```

C.1.2 Answer Generation Prompt

```
System: You analyze time series patterns and generate concise answers.

User:
You are given a time series window [ {window_start}, {window_end} ],
    which belongs to a longer series.
Global information: {global_information}
Canonical tag (if available): {canonical_tag}
Anomaly description (if any): {anomaly_description}
Data [current_value(normal_value)]: [{data_str}]
Question: {question}

Constraints:
- Focus on the pattern of current_values; avoid relying on normal_values.
- Keep the answer concise (<= 150 words), pattern-first (e.g., sustained
    level change, volatility burst).
- MCQ: start with the correct option letter, then explanation (e.g., "B)
    ...").
- True/False: start with "True" or "False", then explanation.
- Do not quote exact numeric values; reason from shape, persistence,
    variability.
- If no anomaly, state it clearly with supporting evidence.
```

C.2 Q&A Exemplars

C.2.1 Multiple Choice

```
Type: multiple_choice
Window: [668, 707]
Has anomaly: false
Anomaly descriptions: []

Question:
Which of the following best describes the anomaly detection results for
    the time series window from step 668 to 707?
```

```
A) The window contains a sudden spike suggesting an isolated anomaly near
     its center.
B) The window shows a consistent pattern without any signs of anomalous
    behavior.
C) The window includes a gradual drift at the boundary, indicating a
    possible transition to an anomaly.
D) The window exhibits irregular fluctuations that are characteristic of
    multiple overlapping anomalies.

Answer:
B) The window shows a consistent pattern without any signs of anomalous
    behavior.
The time series window from step 668 to 707 displays a steady, gradual
    decrease in values, with each step typically dropping by 0.01 or
    remaining constant for a brief interval before continuing the gradual
     trend. There are no sudden spikes, irregular fluctuations, or abrupt
     changes that would indicate an anomaly. The pattern is smooth and
    consistent throughout the window, suggesting stable behavior and the
    absence of any anomalous events within this interval.
```

C.2.2 TRUE/FALSE

```
Type: true_false
Window: [418, 458]
Has anomaly: true
Anomaly descriptions:
["A local continuous upward spike anomaly, featuring 5 consecutive spikes
     with amplitudes from 3.03 to 5.93"]

Question:
True or False: The presence of a cluster of consecutive upward spikes
    within the window from step 418 to 458 suggests an anomalous
    deviation from the expected time series pattern.

Answer:
True. Within the window from step 418 to 458, there is a cluster of
    consecutive upward spikes-specifically, a sequence of five elevated
    values (2.26, 2.05, 2.05, 2.05, 0.71) that stand out sharply against
    the otherwise stable and low baseline pattern (around 0.7). This
    sudden, localized series of spikes deviates significantly from the
    expected behavior of the time series, which is generally flat and
    consistent. The presence of these consecutive upward anomalies
    suggests a notable and abrupt deviation from the normal pattern,
    indicating an anomalous event within this segment of the data.
```

C.2.3 OPEN-ENDED

```
Type: open_ended
Window: [294, 311]
Has anomaly: false
Anomaly descriptions: []

Question:
How would you assess the presence or absence of anomalies in the time
    series window from step 294 to 311, and what evidence from the data
    supports your conclusion regarding the normality or abnormality of
    this segment?

Answer:
There is no evidence of anomalies in the time series window from step 294
     to 311. The values show a gradual decrease from 0.88 to 0.62,
    followed by a mild recovery back up to 0.73. This pattern reflects a
```

```
smooth and consistent trend without sudden spikes, abrupt drops, or
irregular fluctuations. The transitions between values are steady and
 do not indicate any unusual or unexpected behavior. Overall, the
segment demonstrates normal and stable behavior within the observed
window.
```

D BENCHMARK INTEGRITY CHECK PROTOCOLS

To ensure the scientific validity and reproducibility of our benchmark, we implemented a rigorous three-stage automated filtering pipeline to identify and remove low-quality or biased samples.

1. Label Consistency Check. This step ensures that the semantic explanation aligns with the ground-truth anomaly label. We parse the generated answer to extract the definitive judgment (e.g., "Yes, there is an anomaly" or "No anomaly detected"). We then compare this judgment against the procedural injection log. If the generator claims an anomaly exists in a window that the procedural engine marked as normal (or vice versa), the sample is discarded. This eliminates "hallucinated" anomalies where the LLM invents patterns not present in the signal.

2. Information Leakage Check. Since the question generation agent has access to metadata about the injected anomaly, there is a risk that the question itself might reveal the answer (e.g., "Why is there a sudden spike at step 50?"). We use a heuristic filter to detect questions that contain specific anomaly types or location indicators that should be part of the answer. Questions that inadvertently leak the ground truth are removed to ensure the task remains a valid test of detection and reasoning.

3. Stylistic Normalization. LLM-generated text can sometimes contain distinct stylistic artifacts ("LLM-isms") or repetitive phrasing. If left unchecked, these could serve as spurious correlations that a model might exploit (e.g., learning that longer answers always correspond to anomalies). We apply a normalization step to standardize the output format, remove repetitive filler phrases, and ensure a neutral tone. This minimizes the risk of the evaluator model relying on superficial textual cues rather than the semantic content of the explanation.

E PHASE I TSAD RESULTS

Datasets. We benchmark nine public univariate time-series anomaly detection corpora: IOPS, MGAB, NAB, NEK, Power, SED, TODS, UCR, and YAHOO. These datasets span industrial KPIs, synthetic chaotic dynamics, web traffic, power consumption, network/system logs, and curated mixed-domain series. Each series is standardized via z-score normalization using training-set statistics.

Baselines. The comparison covers classical, deep generative, and foundation-style models. IForest (Liu et al., 2008) and LOF (Breunig et al., 2000) serve as nonparametric unsupervised baselines. OmniAnomaly (Su et al., 2019), USAD (Audibert et al., 2020), and TranAD (Tuli et al., 2022) represent deep unsupervised anomaly detectors. DADA (Shentu et al., 2024) and TSPulse (Ekambaram et al., 2025), MOMENT_ZS (Goswami et al., 2024), Chronos (Ansari et al., 2024), TimesFM (Das et al., 2024), and Time_MOE (Xiaoming et al., 2025) are pretrained time-series models evaluated in zero-shot or lightly adapted settings. AXIS denotes a semantic representation-based detector. Implementations follow public releases or authors' configurations when feasible.

Metrics. We report three standard metrics. (i) PA-F1: the point-adjusted F1 that credits a hit if any point within an anomalous window is flagged, computed from precision and recall after point adjustment; threshold for PA-F1 is selected on a validation split when available, or via an unsupervised percentile heuristic on training scores. (ii) AUC-ROC: area under the receiver operating characteristic, threshold-free and aggregated over all test points. (iii) AUC-PR: area under the precision–recall curve, which is more informative under severe class imbalance. Higher values indicate better performance for all metrics.

Table 5: Model Performance Comparison Across Univariate Datasets

		Foundation Models							Deep Learning			Classical	
Dataset	Metric	AXIS	Chronos	DADA	MOMENT_ZS	TS_Pulse	Time_MOE	TimesFM	OmniAnomaly	TranAD	USAD	IForest	LOF
IOPS	PA-F1	57.19	64.17	55.52	44.76	50.59	41.58	**87.89**	53.50	53.04	41.24	15.31	41.38
	AUC-ROC	71.13	72.16	78.61	**82.67**	49.71	61.87	70.12	80.89	69.44	81.31	48.29	66.80
	AUC-PR	18.13	25.65	29.59	22.45	1.65	24.15	28.33	**41.63**	25.51	21.96	6.49	25.63
MGAB	PA-F1	7.13	10.16	3.11	2.90	11.39	2.64	7.74	7.62	7.80	3.49	5.85	**12.31**
	AUC-ROC	**69.97**	50.78	54.12	47.47	49.94	37.73	49.97	60.15	61.04	51.12	43.88	50.27
	AUC-PR	0.55	0.31	0.31	0.25	0.27	0.20	0.27	0.41	0.42	**1.74**	0.24	0.34
NAB	PA-F1	93.73	97.08	96.18	94.61	97.76	90.48	**99.09**	98.02	97.93	93.42	42.81	86.79
	AUC-ROC	51.79	53.82	56.23	64.01	50.28	50.82	53.33	56.26	54.33	**73.09**	55.26	49.93
	AUC-PR	21.00	20.62	21.87	42.73	13.91	20.32	21.12	24.87	21.92	**53.14**	20.48	18.55
NEK	PA-F1	87.19	98.44	98.68	81.10	88.14	50.82	**98.79**	88.50	86.78	66.26	72.47	87.35
	AUC-ROC	53.28	57.60	82.51	**92.02**	53.94	37.41	62.64	88.85	81.25	91.65	85.66	81.38
	AUC-PR	30.32	30.63	44.90	55.75	16.34	7.13	34.05	**73.10**	57.39	52.91	52.76	55.63
Power	PA-F1	**98.94**	95.97	89.53	90.25	98.77	83.29	97.58	86.66	87.51	68.06	0.00	87.92
	AUC-ROC	62.47	48.82	46.19	43.23	50.68	38.92	46.78	59.88	57.05	**66.46**	50.00	38.93
	AUC-PR	**20.52**	10.44	10.07	10.68	11.09	9.01	9.98	13.64	12.37	17.30	10.97	8.98
SED	PA-F1	**93.59**	38.77	18.69	1.92	70.26	33.29	32.49	2.25	4.49	0.00	31.62	0.00
	AUC-ROC	**96.35**	53.03	14.85	1.43	49.23	69.08	29.64	28.93	12.89	10.71	27.96	27.35
	AUC-PR	**73.76**	4.80	2.77	2.57	4.95	7.45	3.26	3.35	2.78	2.68	3.26	3.25
TODS	PA-F1	77.50	**83.40**	61.20	26.86	53.03	44.92	81.26	38.82	40.31	40.86	33.25	54.62
	AUC-ROC	**92.40**	68.45	68.16	46.17	50.20	51.68	72.96	49.69	48.87	53.92	51.16	53.12
	AUC-PR	**68.20**	35.60	20.33	10.03	6.98	10.36	36.69	6.89	6.74	17.37	7.43	19.74
UCR	PA-F1	47.58	43.71	34.90	25.55	56.57	31.63	**60.88**	31.44	32.67	22.57	23.02	32.43
	AUC-ROC	**79.34**	55.79	56.27	54.92	50.57	55.38	57.21	50.67	50.64	53.74	51.93	55.86
	AUC-PR	**24.06**	6.11	2.19	6.92	0.85	2.12	6.35	2.91	2.32	8.08	2.54	2.50
YAHOO	PA-F1	53.88	54.09	50.82	12.54	7.99	21.39	**87.18**	18.66	8.09	7.40	2.86	29.78
	AUC-ROC	**96.55**	94.64	95.95	74.37	49.02	63.08	96.35	77.71	56.47	62.54	48.90	72.85
	AUC-PR	**86.07**	70.75	76.66	4.84	1.49	22.08	79.23	16.28	2.90	3.72	1.66	44.26
Overall	Avg. Rank	**3.81**	4.74	5.52	7.48	7.70	8.70	4.41	5.48	7.00	6.78	9.07	7.30

Table 6: Model Performance Comparison Across Multivariate Datasets

		Foundation Models							Deep Learning			Classical	
Dataset	Metric	AXIS	Chronos	DADA	MOMENT_ZS	TS_Pulse	Time_MOE	TimesFM	OmniAnomaly	TranAD	USAD	LOF	IForest
MSL	PA-F1	82.99	81.68	86.93	88.46	82.30	82.08	75.88	50.15	83.35	51.16	**89.56**	56.66
	AUC-ROC	72.17	69.79	**75.32**	52.15	51.05	64.59	70.45	62.28	49.02	61.98	56.65	52.88
	AUC-PR	20.56	20.21	**26.42**	11.60	7.46	19.83	20.34	17.19	7.42	17.24	15.18	11.31
PSM	PA-F1	93.50	96.82	**98.99**	93.97	96.10	94.32	95.18	60.53	84.45	62.26	96.87	80.10
	AUC-ROC	56.48	54.82	60.84	**67.85**	49.83	61.34	54.47	53.10	45.04	42.95	52.34	51.50
	AUC-PR	19.47	34.56	39.92	**42.83**	14.45	42.46	33.83	17.18	13.38	12.64	17.00	18.49
SMAP	PA-F1	**86.76**	66.89	69.18	81.03	81.55	71.00	64.56	25.74	62.93	26.23	85.58	58.47
	AUC-ROC	**62.61**	58.53	51.24	47.89	49.83	40.23	56.46	51.96	47.99	51.38	51.97	54.43
	AUC-PR	**24.13**	15.21	12.08	11.90	4.03	10.56	13.87	4.07	3.93	4.18	7.73	7.93
SMD	PA-F1	**96.15**	83.72	84.03	75.79	75.88	77.02	84.10	30.58	57.05	27.89	92.56	55.02
	AUC-ROC	**74.26**	69.88	72.02	66.28	49.93	66.57	70.15	55.76	52.59	49.78	65.75	60.53
	AUC-PR	**39.44**	11.68	13.16	11.33	4.58	11.31	13.32	5.96	5.45	5.47	15.72	11.04
SWaT	PA-F1	83.85	83.78	89.55	88.08	**97.32**	84.44	86.07	70.26	72.80	69.93	90.72	0.00
	AUC-ROC	55.49	29.85	79.96	**81.96**	50.38	80.74	24.67	42.56	41.55	41.99	50.38	50.00
	AUC-PR	17.48	8.84	50.60	70.79	15.69	**70.85**	8.58	15.21	14.72	16.04	16.64	15.49
Overall	Avg. Rank	**3.20**	5.53	3.27	5.13	7.87	5.13	5.53	9.07	9.93	9.73	4.93	8.60

Results on Univariate Benchmarks. The experimental results, summarized in Table 5, demonstrate that AXIS achieves competitive performance across nine public univariate benchmarks. To rigorously evaluate the statistical significance of these rankings, we present a Critical Difference (CD) diagram in Figure 8. AXIS achieves a Average Rank of 3.81. This validates that our Phase I encoder learns robust temporal representations essential for anomaly detection.

Extensibility to Multivariate Contexts. While our current work primarily focuses on univariate settings to establish the core "step-aligned hint" reasoning mechanism, our framework is inherently extensible to multivariate contexts. Specifically, we incorporate the Any-variate Attention mechanism (Woo et al., 2024), which allows the model to learn cross-channel interactions. By integrating this mechanism, we construct AXIS-multi, which effectively handles multivariate attention by enabling each time step to receive contextual hints incorporating information from related channels. To validate this extensibility, we further evaluated AXIS-multi on five standard multivariate datasets (MSL, PSM, SMAP, SMD, SWaT). As shown in Table 6, AXIS-multi achieves the best overall performance with an Average Rank of 3.20, outperforming recent time-series foundation models (e.g., Chronos, MOMENT). This confirms that our step-aligned hint mechanism scales effectively to high-dimensional signals, capturing cross-channel dependencies without performance degradation.

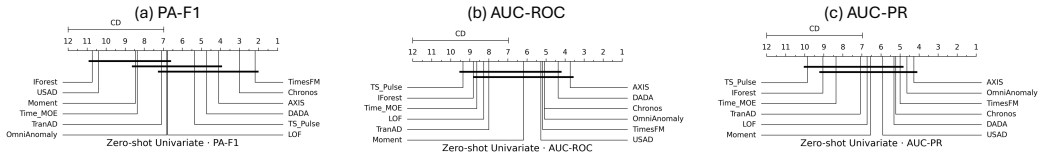

Figure 8: Critical Difference (CD) diagram comparing the statistical performance of AXIS against baselines across nine univariate datasets.

F Benchmark Analysis and Validity

Dataset Composition and Scale. Our benchmark comprises **140 unique time series** generating **280 evaluation windows**, of which **98 (35.0%)** contain anomalies and **182 (65.0%)** are normal. The dataset spans diverse domains and exhibits significant variability in temporal structure: original series lengths range from 64 to 1,020 time steps (median: 617), while evaluation windows range from 15 to 40 steps (median: 28). This distribution ensures the benchmark tests both short-term pattern recognition and longer-context reasoning capabilities. The benchmark is carefully balanced across three question types: **open-ended (34.64%)**, **multiple-choice (32.86%)**, and **true/false (32.50%)**. Importantly, the anomaly ratio is consistent across question types (31.87%–37.11%), preventing any systematic bias where certain question formats disproportionately test anomaly vs. normal cases.

Table 7: Performance of standard anomaly detection algorithms on benchmark synthetic data.

Dataset	F1-PA	ROC-AUC	ROC-PRC
IForest	0.3912	0.5851	0.1616
KNN	0.4708	0.6762	0.1993
LOF	0.4596	0.7011	0.2072
PCA	0.3537	0.5908	0.1666

Difficulty Validation via Synthetic Time Series. We evaluated four standard unsupervised anomaly detection algorithms implemented by TSB-AD on the same synthetic time series data used in our benchmark. Table 7 summarizes their performance using F1-score with point adjustment (F1-PA), ROC-AUC, and Precision-Recall AUC (ROC-PRC). None of the traditional algorithms achieve high scores across the board, with the best average F1-PA score being only 0.4708 (KNN). These metrics indicate that the underlying anomaly patterns are non-trivial and challenging for established methods to detect reliably.

G Detailed Procedure for G-Eval

This appendix documents the evaluation methodology, criteria, and prompts used by our G-Eval variant.

G.1 Methodological Overview

We adopt an advanced G-Eval judge that evaluates system answers along multiple dimensions and calibrates discrete scores using token-level log-probabilities. For each question-answer instance:

1. Determine the question type (e.g., multiple choice, open-ended, true/false) and instantiate the corresponding set of evaluation dimensions.
2. For each dimension, query a LLM judge with a criterion-specific instruction (see the exact prompt in Appx.) to produce a reasoning-based assessment and a final integer score $s \in \{1, 2, 3, 4, 5\}$. If score extraction is ambiguous, a conservative default is used.
3. Obtain token-level probabilities assigned to the score tokens $\{1, \ldots, 5\}$. Exponentiate and normalize these probabilities to construct a score distribution. Use its expectation as a log-probability-weighted score and compute a confidence via normalized entropy (Appx. G.3).

G.2 Evaluation Criteria and Weights

For each `question_type`, the evaluator applies a fixed set of dimensions with 5-point guidelines and combines them via type-specific weights:

Table 8: Dimension Weights by Question Type.

multiple_choice	correctness: 0.70, reasoning_quality: 0.30
open_ended	relevance: 0.30, completeness: 0.35, accuracy: 0.35
true_false	correctness: 0.60, justification_quality: 0.40

The dimension descriptions and scoring guidelines (1–5) are:

multiple_choice

- **correctness**: How accurate is the generated response compared to the expected answer?
  - 5: Perfect match, completely correct
  - 4: Mostly correct; minor deviations not affecting core meaning
  - 3: Partially correct; captures some key aspects but misses important details
  - 2: Somewhat relevant but with significant errors or omissions
  - 1: Incorrect or completely irrelevant
- **reasoning_quality**: How well does the response demonstrate logical reasoning and explanation?
  - 5: Clear, logical, comprehensive reasoning fully explains the choice
  - 4: Good reasoning with minor gaps
  - 3: Adequate reasoning; lacks depth or has inconsistencies
  - 2: Weak reasoning; significant gaps or flawed logic
  - 1: No clear reasoning or completely flawed logic

open_ended

- **relevance**: How relevant and on-topic is the generated response?
  - 5: Completely relevant; directly addresses all aspects
  - 4: Highly relevant; minor omissions
  - 3: Moderately relevant; addresses core aspects but misses details
  - 2: Somewhat relevant; off-topic content or key omissions
  - 1: Irrelevant or off-topic
- **completeness**: How complete and comprehensive is the response?
  - 5: Fully comprehensive; covers all necessary aspects
  - 4: Mostly complete; minor gaps
  - 3: Adequately complete; missing some important details
  - 2: Incomplete; significant information missing
  - 1: Very incomplete
- **accuracy**: How factually accurate is the response?
  - 5: Completely accurate; no factual errors
  - 4: Mostly accurate; very minor inaccuracies
  - 3: Generally accurate; some notable errors
  - 2: Several factual errors affecting reliability
  - 1: Major factual errors or mostly inaccurate

true_false

- **correctness**: How correct is the T/F judgment and supporting explanation?
  - 5: Perfect judgment; excellent supporting explanation
  - 4: Correct judgment; good explanation
  - 3: Correct judgment; adequate explanation
  - 2: Incorrect judgment; reasonable attempt at explanation
  - 1: Incorrect judgment; poor or no explanation
- **justification_quality**: How well does the response justify the decision?
  - 5: Excellent justification with clear evidence and reasoning
  - 4: Good justification with solid evidence
  - 3: Adequate justification with some evidence
  - 2: Weak justification with little evidence
  - 1: No meaningful justification

G.3 Advanced Scoring From Log-Probabilities

Let $s \in \{1, 2, 3, 4, 5\}$ denote the raw score extracted from the LLM's evaluation output. We also obtain token-level log-probabilities for score tokens $\{1, \ldots, 5\}$. The evaluator constructs a distribution over scores by exponentiating and normalizing these log-probabilities:

$$p(s) \propto \exp(\log p_s), \quad \tilde{p}(s) = \frac{\exp(\log p_s)}{\sum_{k=1}^{5} \exp(\log p_k)}.$$

The **weighted score** is the expectation under $\tilde{p}$:

$$\text{WeightedScore} = \sum_{s=1}^{5} s \cdot \tilde{p}(s).$$

G.4 Advanced G-Eval Prompt

```
You are an expert evaluator for natural language generation systems. Your
     task is to evaluate the quality of generated responses using the G-
    Eval methodology with chain-of-thought reasoning.

Control the Maximum Length to 500 words.

**Evaluation Criterion: {criterion.dimension}**
{criterion.description}

**Scoring Guidelines:**
{guidelines_text}

**Question:** {question}

**Expected Answer:** {expected_answer}

**Generated Response:** {generated_response}

**Instructions:**
1. Analyze the generated response step by step using chain-of-thought
    reasoning
2. Compare it against the expected answer for the specified criterion
3. Consider both content quality and alignment with the expected answer
4. Provide detailed reasoning for your evaluation
5. Conclude with a single score from 1-5
6. Ignore error in index mismatch, just focus on the content

Please follow this exact format for your response:
```

```
**Step-by-step Analysis:**
[Provide detailed chain-of-thought analysis]

**Comparison with Expected Answer:**
[Compare generated response with expected answer]

**Final Assessment:**
[Summarize your evaluation]

**Score:** [Single integer: 1, 2, 3, 4, or 5]
```

G.5 Detailed results of G-eval

Visualize and pair-wised comparsion Figures 9 to 11 present forest plots to visually summarize the pairwise model performance comparisons from Tables 1 to 3. The confidence intervals are calculated from the variance of the pairwise total scores.

As shown in Fig. 9, our AXIS model demonstrates a strong performance. On Multiple-choice questions, it significantly outperforms all models except for Image-LLM and Anon-LLM(Window). For True/False questions, AXIS significantly surpasses all other models, highlighting its advantage in objective answering. Furthermore, on Open-Ended questions, AXIS outperforms other models while achieving performance comparable to the General LLM (AnomLLM).

Fig. 10 details an ablation study of our proposed AXIS model. The results demonstrate that AXIS consistently and significantly outperforms its variants ("w/o-windows", "w/o-task-hint", and "w/o-context-hint") across all three question categories. The confidence intervals for the mean paired difference are entirely below zero in all subplots, indicating that the removal of any of these components leads to a statistically significant degradation in performance. This underscores the integral contribution of each component to the overall efficacy of the AXIS model.

Fig. 11 provides a comparative analysis of various open-source models against the Deepseek Llama 8B (Inst) baseline. In multiple-choice questions, Deepseek Llama 8B shows a significant performance advantage over the Mistral 7B models, while its performance is statistically comparable to the Qwen2.5 7B series. For open-ended and true/false questions, Deepseek Llama 8B shows comparable performance to most Qwen and Deepseek-Qwen variants, showing that the robustness of our AXIS design.

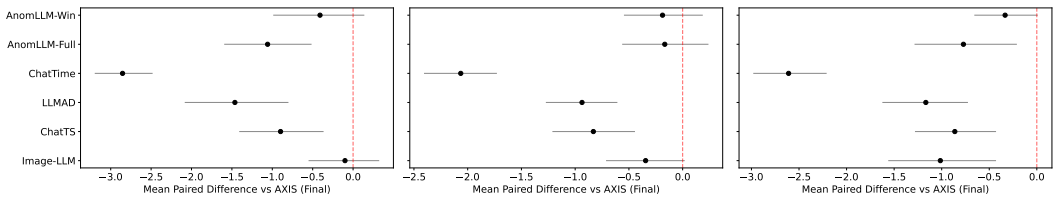

Figure 9: Comparative analysis of baseline models against AXIS via forest plot. Each horizontal line represents the 95% bootstrap confidence interval for the mean paired score difference relative to the AXIS baseline. The central dot marks the point estimate of the mean difference. The vertical red dashed line at zero indicates no difference; confidence intervals crossing this line suggest that the model's performance is not statistically different from the baseline.

Analysis for different LLM judge: We introduce gpt-4.1 as the second LLM judge to complement our primary judge, Gemini-2.5. We then conduct a comprehensive inter-rater reliability analysis using a human-annotated subset. Our analysis, summarized in Table 9, confirms the robustness of our evaluation methodology. First, we observe high consistency between our two LLM judges, Gemini-2.5 and Gpt-4.1, which achieve an overall Kendall's W of 0.882. This demonstrates that our evaluation prompts and criteria are stable and reproducible across different state-of-the-art models. Second, and more importantly, both LLM judges show strong alignment with human expert ratings. Our primary judge, Gemini-2.5, achieves a strong agreement of W=0.743 with human ratings, while

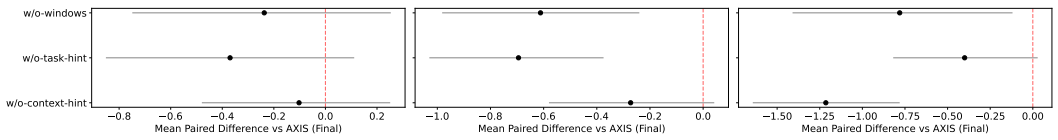

Figure 10: Forest plot comparing AXIS variants against the AXIS baseline. The plot displays the mean paired score difference and the 95% bootstrap confidence interval for each variant. The vertical red line at zero indicates no performance difference relative to AXIS.

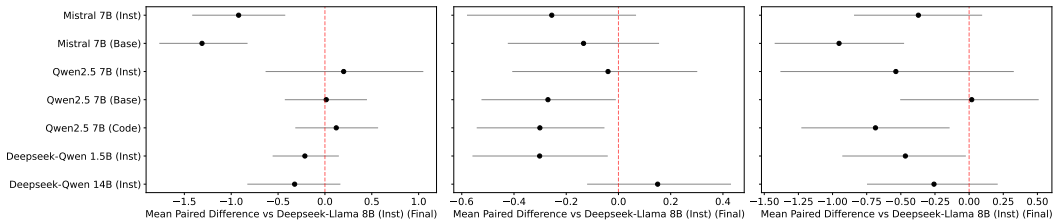

Figure 11: Forest plot for ablation studies, benchmarked against AXIS. This plot illustrates the mean paired score difference and the corresponding 95% bootstrap confidence interval for each ablated variant. The vertical red line at zero serves as the reference for no difference from the baseline.

our second judge, Gpt-4.1, also achieves a strong agreement of W=0.750. This alignment remains robust even for the most challenging open-ended questions, where the judges score W=0.670 and W=0.679, respectively. To contextualize these results, we measure the inter-rater agreement among the human experts themselves, finding a Kendall's W of 0.587. This indicates that our LLM judges not only agree with each other but also perform at a level of consistency comparable to human experts for this subjective task. We report all correlation coefficients using Kendall's W, as shown in the table below.

Table 9: Inter-rater agreement (Kendall's W) between LLM judges and human experts.

Question Type	Gpt-4.1 vs. Human	Gemini-2.5 vs. Gpt-4.1	Gemini-2.5 vs. Human	Human Inter-rater Agreement
Overall	0.750	0.882	0.743	0.587
Multiple Choice	0.857	0.833	0.750	0.667
Open-Ended	0.679	0.912	0.670	0.587
True/False	0.749	0.856	0.784	0.535

G.6 Computational Efficiency Analysis

We analyze the computational overhead of AXIS compared to baseline methods by measuring average prompt token counts and generation latency across question types (see Table 11).

Token Efficiency. AXIS (14B) demonstrates high token efficiency, averaging ∼335 prompt tokens. This is comparable to `ChatTS` (14B) (∼363 tokens) and significantly more efficient than text-serialization approaches like `LLMAD` (∼1370 tokens) and `Image LLM` (∼1479 tokens). While `AnomLLM(Window)` uses the fewest tokens, it sacrifices reasoning capability by limiting context to a local window.

Latency and Performance Balance. AXIS achieves a favorable balance between speed and performance. It incurs only a modest latency overhead (∼1.5–2s) compared to `ChatTS` while delivering superior reasoning, and is ∼7× faster than `ChatTime`, which suffers from prohibitive latency. Both token counts and latencies remain consistent across question types, indicating robust efficiency for diverse reasoning tasks.

Table 10: G-eval results by LLM judge gpt-4.1

Model	Multiple Choice			Open Ended				True False		
	Final	Corr.	Rsn. Qual.	Final	Rel.	Comp.	Acc.	Final	Corr.	Justif.
AXIS	**4.52**	**4.56**	**4.49**	**3.54**	**3.67**	**3.29**	**3.65**	**3.94**	**3.71**	**4.17**
Image LLM	4.40	4.35	4.44	3.40	3.58	3.24	3.38	2.98	2.71	3.24
ChatTS	3.71	3.77	3.65	3.11	3.27	2.95	3.11	3.05	2.90	3.19
LLMAD	3.66	3.30	4.02	2.86	2.93	2.80	2.85	3.50	3.31	3.69
ChatTime	1.47	1.91	1.02	1.07	1.02	1.04	1.15	1.10	1.17	1.02
AnomLLM(Full)	4.23	4.12	4.35	3.13	3.35	2.98	3.05	2.76	2.69	2.83
AnomLLM(Window)	2.80	2.81	2.79	2.93	3.05	2.58	3.15	2.55	2.57	2.52

Table 11: Comparison of Computational Efficiency: Prompt Tokens, Latency, and Memory

Model	**Multiple Choice**			**Open Ended**			**True False**		
	Tokens	**Lat (s)**	**Mem (GB)**	**Tokens**	**Lat (s)**	**Mem (GB)**	**Tokens**	**Lat (s)**	**Mem (GB)**
Image LLM	1548.42	9.57	-	1455.33	10.83	-	1433.33	9.21	-
LLMAD	1364.42	2.44	-	1375.16	2.36	-	1372.00	2.44	-
AnomLLM(Full)	3083.81	1.75	-	3075.64	2.01	-	2983.40	1.82	-
AnomLLM(Window)	312.53	0.73	-	258.00	0.74	-	240.43	0.70	-
ChatTime 14B	288.49	31.06	38.5	182.45	28.88	37.4	181.05	29.15	38.1
ChatTS 14B	411.95	5.02	32.3	347.24	5.77	31.9	330.10	4.75	32.1
AXIS 14B	376.05	13.70	41.6	321.29	13.76	40.5	303.90	12.98	41.0

H Procedure and detail results for Human-based evaluation

H.1 Designs of experiments

Questionnaire Design and Structure. Our human expert questionnaire is systematically designed to evaluate model explanations across multiple dimensions with rigorous controls for bias and consistency. Each questionnaire comprises: (1) **Question Information**: the original question, expected answer, and question type classification (multiple choice, open-ended, or true/false); (2) **Time Series Visualization**: a plot showing the full time series with the target window highlighted to provide visual context for assessment; (3) **Model Responses**: all baseline model outputs presented in a randomized order to eliminate position bias; (4) **Evaluation Criteria**: dimension-specific scoring guidelines adapted from the G-Eval methodology, employing 5-point Likert scales for correctness, reasoning quality, relevance, completeness, accuracy, and justification quality tailored to each question type; (5) **Scoring Tables**: structured evaluation forms for systematic criterion-based assessment; (6) **Model Ranking**: comparative ranking of all models with written justifications to capture qualitative insights.

Expert Recruitment and Annotation Protocol. For this study, we evaluate 140 unique questions, resulting in 280 completed questionnaires. We recruit 28 human experts through a rigorous vetting process targeting researchers with established expertise in time-series analysis and machine learning. All annotators hold advanced degrees (Ph.D. candidates or post-doctoral fellows) and are recruited from our institution and collaborating academic research laboratories. Experts are compensated at standard academic research rates for their time and effort. To ensure annotation quality and mitigate individual bias, we implement a double-blind protocol wherein each questionnaire is independently evaluated by three randomly assigned experts from the pool of 28, with neither annotator aware of the other's identity or responses. To quantify inter-rater reliability, we compute Kendall's W (Coefficient of Concordance), which is well-suited for measuring agreement among multiple raters on ordinal data. Our analysis yields Kendall's W = 0.587, indicating a "moderate" to "substantial" level of agreement among our expert panel. This coefficient is notably comparable to inter-rater agreement levels reported in similar expert evaluation studies, confirming that our human-annotated dataset provides a reliable and internally consistent gold standard for model assessment.

Human Expert Evaluation Questionnaire Example

Question 1
Question Type: multiple_choice
Question: Which of the following best describes the likelihood of detecting an anomaly within the time series window from step 491 to 521, given its observed behavior?
A) There is a sudden spike at the start of the window that indicates potential anomalous activity B) The window displays periodic irregularities that suggest subtle anomalies C) The window exhibits consistent and stable behavior, suggesting no detectable anomalies D) The pattern shows gradual drift, implying an evolving anomaly towards the window's end
Expected Answer: C) The window exhibits consistent and stable behavior, suggesting no detectable anomalies. The current_value data fluctuates around zero with both positive and negative values but does not display any abrupt spikes, sustained trends, or irregular periodicities...
Time Series Visualization: [Plot showing full time series with highlighted window]
Model Responses:
Model 1 To determine anomaly likelihood in the window from step 491 to 521, let's examine the values: [0.12, -2.45, -1.15, ...]. Within this window, there is a sharp drop from 0.12 to -2.45 at the very start... **A) There is a sudden spike at the start of the window that indicates potential anomalous activity**
Model 2 Within the specified window (indices 491 to 521), the data shows fluctuating values without any abrupt spikes or consistent irregular patterns...
Model 3 The observed behavior within the time series window from step 491 to 521 indicates no sudden spikes or abrupt changes... **C) The window exhibits consistent and stable behavior, suggesting no f anomalies.**
Model 4 The time series visualization highlights the analysis window from step 491 to 521... **C) The window exhibits consistent and stable behavior, suggesting no detectable anomalies.**
Model 5 {"briefExplanation": {"step1_global": "The overall trend of the latest 31 data points is stable...", "is_anomaly": false, "anomalies": [], ...}
Model 6 **Answer: C) The window exhibits consistent and stable behavior, suggesting no detectable anomalies.** The time series window from step 491 to 521 shows a mix of positive and negative values with moderate fluctuations, but there are no sudden spikes...
Evaluation Criteria:
Correctness (1-5): How accurate is the generated response compared to the expected answer? **Reasoning Quality (1-5):** How well does the response demonstrate logical reasoning and explanation?

Scoring Table:

Model	Correctness	Reasoning Quality	Comments
Model 1	---	---	---
Model 2	---	---	---
Model 3	---	---	---
Model 4	---	---	---
Model 5	---	---	---
Model 6	---	---	---

Model Ranking: Rank all models from best (1) to worst (6) with justifications.

H.2 Detailed results for Human evaluation

Fig. 12 provides a criterion-level breakdown of human evaluation across the three question types. AXIS ranks first consistently. On Multiple Choice questions, it achieves the best scores in both *Correctness* and *Reasoning Quality*, with a clear margin over the next best visual baseline. This indicates that our hint-based conditioning not only selects the right option but also articulates concise, logically grounded rationale.

For Open-Ended questions, AXIS leads on *Relevance*, *Completeness*, and *Accuracy*, reflecting faithful, fully supported explanations rather than surface descriptors. On True/False questions, it also tops both *Correctness* and *Justification Quality*, showing strong calibration and evidence-backed decisions. Overall, these results demonstrate that the proposed three-pathway design (symbolic

numeric grounding, context-integrated local hints, and task-prior hints) confers robust advantages across formats—surpassing specialized TS-LLMs and multimodal vision–language approaches in both accuracy and human-judged explanatory quality.

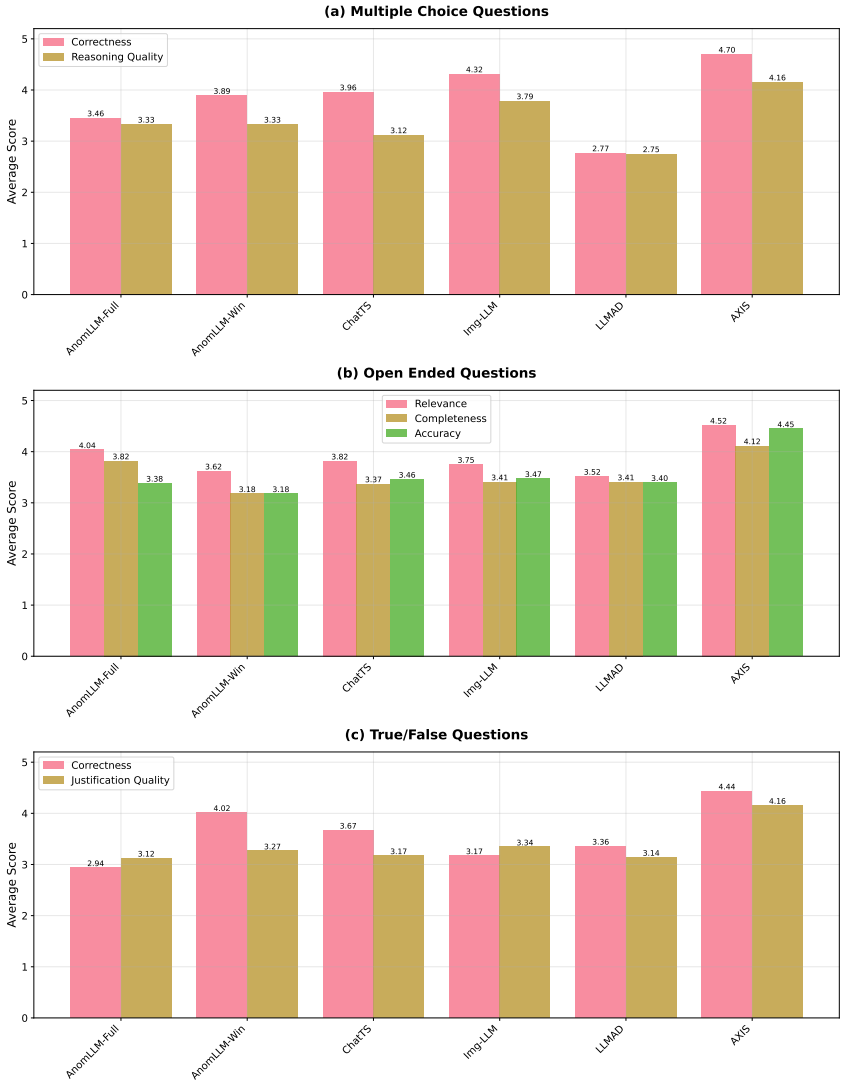

Figure 12: The mean scores by human evaluation for (a) multiple choice question, (b) open ended questions and (c) true/false questions

H.3 Benchmark Construction & Evaluation for Public Datasets

To evaluate the generalization capabilities of our model on real-world data, we construct a comprehensive human evaluation benchmark using 108 time series from three public datasets: YAHOO, TODS, and NEK. These datasets cover diverse domains including web traffic, distributed system metrics, and sensor data. For each time series, we design two distinct question formats: a True/False (TF) question and a Multiple-Choice Question (MCQ), to assess the quality of generated explanations from different perspectives.

The benchmark construction process begins with a deterministic, seed-based sampling procedure to select two representative windows from each of the 108 time series, ensuring reproducibility. For each series, we sample:

- **An Anomaly Window**: A segment of 15 to 40 time steps that contains at least one point labeled as an anomaly.
- **A Normal Window**: A segment of the same length range (15-40 steps) containing exclusively normal data points.

Each selected window is visualized by highlighting it on a plot of the entire time series, providing essential visual context for the evaluation. These selected windows and their corresponding visualizations form the basis for both the TF and MCQ tasks.

True/False (TF) Question Generation For the TF task, we formulate a standardized question for every window: *"True or False: The time series window from step {start} to {end} shows only normal behavior, with no evidence of sudden deviations or irregular patterns that would indicate an anomaly."* The ground truth answer is "False" for anomaly windows and "True" for normal windows. The ground truth explanation for an anomaly window is derived from a predefined set of detailed descriptions corresponding to expert-provided labels (e.g., "downward spike," "sudden increase"), while the explanation for a normal window confirms the absence of anomalous patterns.

Multiple-Choice Question (MCQ) Generation Using the same set of windows, we design an MCQ task to assess a model's ability to not only detect but also correctly classify anomalies. The question asks: *"Which of the following best describes the anomaly detection results within the time series window spanning steps {start} to {end}?"* For each question, we generate three distinct options:

- **The Correct Option**: This option accurately describes the window's content. For an anomaly window, it corresponds to the specific, expert-labeled anomaly type (e.g., "upward spike"). For a normal window, it is the "normal" option.
- **Distractor Options**: Two incorrect options are provided to serve as distractors. If the window is anomalous, the options include "normal" and another randomly selected anomaly type from the overall pool of labels. If the window is normal, the distractors are two different, randomly selected anomaly types.

Crucially, each option is presented not just as a short label but as a full descriptive sentence (e.g., *"Upward Spike anomaly: There is a sharp upward jump in the measurements."*) to ensure clarity and prevent ambiguity. This design forces the model to discern nuanced differences between various anomaly types and normal behavior.

Evaluation Criteria To ensure a granular assessment of the models' explanatory power, we establish a detailed scoring rubric for both TF and MCQ tasks, administered by expert human annotators. For the **True/False task**, we use a 6-point scale (0–5) to evaluate both the correctness of the binary answer and the quality of the accompanying justification. A score of 5 represents a correct T/F judgment with a clear, accurate, and comprehensive explanation. Lower scores are assigned for answers with less precise, generic, or irrelevant justifications. A score of 0 indicates an incorrect T/F judgment with a nonsensical explanation. For the **Multiple-Choice task**, scores are assigned based on the model's ability to not only detect but also correctly classify the anomaly:

- **4–5 (Correct Classification):** A score of 5 is awarded for correctly identifying the specific anomaly type (or 'normal') with a clear and accurate explanation. A score of 4 is given for a correct classification with a more generic or slightly flawed explanation.
- **2–3 (Correct Detection, Incorrect Classification):** A score of 3 is assigned if the model correctly identifies the window as anomalous but selects the wrong anomaly type, providing a plausible justification for its incorrect choice. A score of 2 is given for the same error but with a poor or irrelevant explanation.
- **0 (Incorrect Detection):** A score of 0 is given if the model fails to correctly determine whether an anomaly is present (e.g., classifying an anomalous window as 'normal' or vice versa).

This multi-faceted evaluation framework allows us to capture nuanced differences in the quality of explanations beyond simple accuracy metrics.

I Causal Ablation Analysis for Hint Tokens

To analyze the individual contribution of each local hint token to the model's generation quality, we implement a causal ablation study. This method systematically removes or modifies specific hint tokens to quantify their importance in producing accurate explanations.

Given hint embeddings $\tilde{\mathbf{H}}_{s:e} \in \mathbb{R}^{(e-s)\times d_h}$, we measure the contribution of each local hint token at position i within the target window $[s, e)$ through the following procedure:

Baseline Computation. First, we compute the baseline log-likelihood using the complete hint embeddings:

$$\mathcal{L}_{\text{baseline}} = -\log P(\mathbf{y}|\tilde{\mathbf{H}}_{s:e}, q, \tilde{\mathbf{F}}),$$

where $\mathbf{y}$ is the target explanation, q is the query, and $\tilde{\mathbf{F}}$ represents the task-prior hints.

Ablation Methods. For each position $i \in [s, e)$, we create an ablated version of the hint embeddings $\tilde{\mathbf{H}}^{(i)}_{s:e}$ using **Zero Replacement:** $\tilde{\mathbf{H}}^{(i)}_{s:e}[i-s] = \mathbf{0}$.

Contribution Score. The contribution score for position i is computed as:

$$C_i = \mathcal{L}_{\text{baseline}} - \mathcal{L}^{(i)}_{\text{ablated}},$$

where $\mathcal{L}^{(i)}_{\text{ablated}} = -\log P(\mathbf{y}|\tilde{\mathbf{H}}^{(i)}_{s:e}, q, \tilde{\mathbf{F}})$. A positive C_i indicates that removing the hint at position i degrades the model's performance, suggesting that this position contributes positively to explanation generation.

Validation of Context-Integrated Hints via Causal Ablation Analysis. As depicted in Fig. 13 and Fig. 14, the causal ablation analysis provides empirical evidence for the efficacy of our proposed hint mechanism. A predominant observation across both examples is that the majority of the contribution scores, denoted by C_i, are positive ($C_i > 0$). According to the formulation presented in Appx. I, a positive C_i indicates that ablating the hint token at position i degrades the model's performance, as measured by an increase in the negative log-likelihood of the target explanation. This finding strongly supports our hypothesis that the context-integrated, step-aligned hints furnish valuable guidance for the model, making a net positive contribution to the generation of high-quality answers.

Furthermore, a more nuanced pattern emerges from the results. We observe that in regions where the time series exhibits significant volatility or sharp fluctuations, the corresponding contribution scores are comparatively lower. This suggests that the inherent complexity and unpredictability of volatile segments in the time series can diminish the marginal utility of individual hint tokens. In essence, while the hints remain beneficial overall, their directional impact is partially mitigated by the increased difficulty of the task in these challenging temporal regions.

J Additional Sensitivity and Robustness Analyses

Analysis of Distractor Design in Multiple Choice Questions. We investigate the hypothesis that the divergent impact of the context-integrated hint across question types stems from the distinct reasoning processes required. While True/False questions demand binary judgments often necessitating global context, Multiple Choice Questions (MCQs) allow for elimination strategies. To validate this, we constructed an evaluation set of 140 MCQs with varying numbers of options ($N \in \{2, 3, 4\}$). Table 12 presents the performance of AXIS and its ablated variants. We observe that with fewer options (2 or 3), where choices are broader, the performance gap between the full model and the 'w/o-context-hint' variant is significant. As the number of options increases to 4, distractors become more specific, allowing the model to rely more on localized features (symbolic numeric hints) for elimination, thereby reducing the relative impact of the global context hint. This confirms that the utility of context hints in MCQs is modulated by distractor complexity.

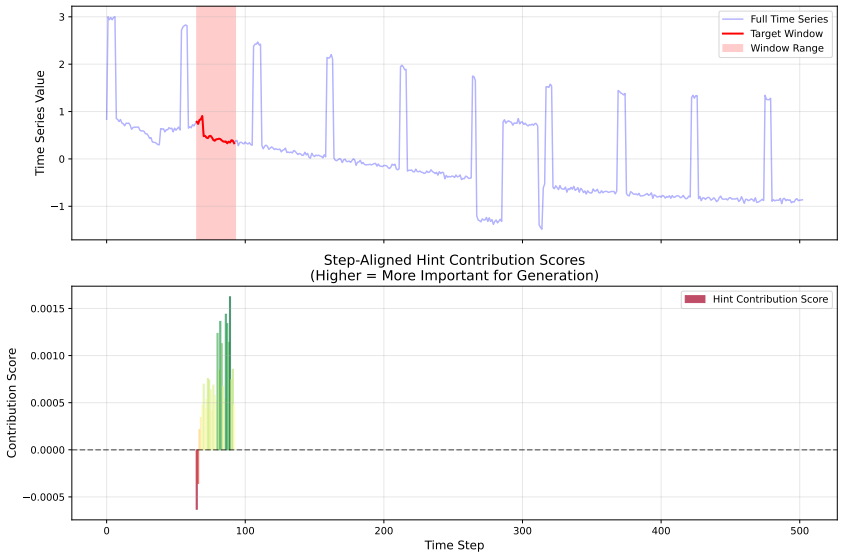

Figure 13: Causal Ablation Analysis of Step-Aligned Hint Tokens. The figure visualizes the contribution score C_i for each hint token, computed via our causal ablation method, alongside the original input time series.

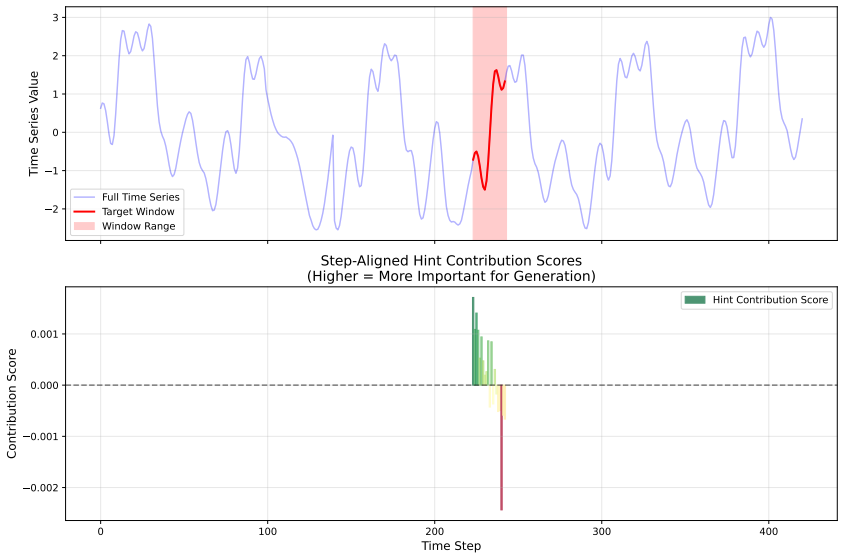

Figure 14: Causal Ablation Analysis of Step-Aligned Hint Tokens. The figure visualizes the contribution score C_i for each hint token, computed via our causal ablation method, alongside the original input time series.

Table 12: Impact of Distractor Options on Model Performance across Ablations

Model	2 Options	3 Options	4 Options
AXIS	3.961 ± 0.304	3.874 ± 0.372	3.983 ± 0.344
w/o-context-hint	3.425 ± 0.336	3.371 ± 0.332	3.791 ± 0.334
w/o-fixed-hint	3.249 ± 0.346	3.369 ± 0.419	3.772 ± 0.381
w/o-windows	3.507 ± 0.299	3.823 ± 0.311	3.734 ± 0.344

Analysis of Window Length Sensitivity. We further analyze the impact of the input question-window length $[s, e)$ on performance by varying it from 10 to 160 time steps. Figure 15 illustrates the sensitivity patterns across question types. For Multiple Choice and Open-Ended questions, performance follows an inverted U-shaped trend. Initial increases in window size improve reasoning by providing more information, but performance declines beyond 40-80 steps, likely due to the processing burden of long numerical sequences on the LLM. In contrast, True/False questions exhibit stability across window sizes. This robustness suggests that binary judgments rely more on the global semantic context provided by the 'context-integrated hint' rather than precise local numerical patterns, making them less sensitive to the granularity of the windowed input.

Figure 15: Performance sensitivity to variations in question-window length across different question types.

Robustness Against Global Noise Injection. To assess model robustness without introducing ambiguity often caused by local perturbations in anomaly detection tasks, we evaluate performance under global Gaussian noise injection. We add zero-mean Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with scales $\sigma \in \{0.1, 0.3, 0.7, 1.5\}$ to the input series. As shown in Figure 16, the model demonstrates strong robustness in discriminative tasks (Multiple Choice and True/False), maintaining stable performance up to noise scales of 0.7. This indicates effective filtering of high-frequency noise to focus on dominant anomalous features. Conversely, the generative Open-Ended task shows higher sensitivity, with performance decreasing monotonically as noise increases, reflecting the challenge of generating detailed semantic descriptions from obscured signal morphologies.

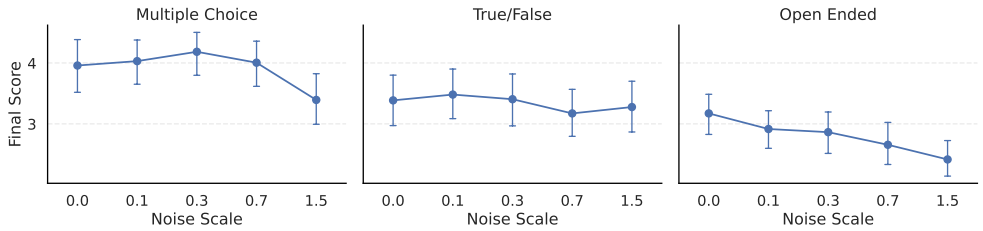

Figure 16: Model robustness evaluation under varying scales of global Gaussian noise injection.