TIMEOMNI-LIVE: CONTINUOUS & LIVE TIME SERIES UNDERSTANDING WITH LONG-TERM MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing time series language models predominantly follow an offline, one-shot paradigm: they process fixed-length inputs independently and do not retain past states. This static approach fundamentally misaligns with the continuous nature of real-world streaming applications and fails to maintain the historical context necessary for cross-window reasoning. To bridge this gap, we introduce Continuous & Live Time Series Understanding (CL-TSU), a novel paradigm that shifts the focus toward maintaining a persistent, stateful understanding over open-ended data streams. To systematically evaluate this paradigm, we present **LIVETS-SUITE**, the first comprehensive data suite tailored for CL-TSU, supporting data generation, model training, and systematic evaluation across 18 tasks spanning six core capabilities. It comprises over 5K streams totaling 2.2B timesteps, annotated with 15K specific response events. We further design an evaluation protocol that jointly assesses response timeliness, content accuracy, and computational efficiency. Building on this foundation, we propose **TIMEOMNI-LIVE**, the first live time series understanding model. It integrates a hybrid time series encoding for natively temporal input, a stream-persistent hybrid memory mechanism to effectively preserve exact historical evidence and long-term task states, and an elastic stream-inference synchronization strategy to reduce processing lag. Evaluations show that TIMEOMNI-LIVE establishes a strong baseline for CL-TSU, outperforming frontier LLMs (90.0 vs. 79.3 for Opus 5) with only 4B parameters and achieving 206× lower p95 latency than its backbone. Deployment on a Jetson edge device further shows its ability to support real-time continuous temporal understanding locally at the edge. Our code is available at this repository.

1 INTRODUCTION

Time series are innately a continuous modality, consisting of observations collected sequentially over time (Podest et al., 2026). These observations are typically generated by sensors that continuously measure ongoing processes without a definitive endpoint (Hill & Minsker, 2010). Yet, most existing time series research is primarily designed for offline settings, where a model receives a predefined input window in a single model call, produces a single prediction or decision, and then halts. Conventional models typically do not preserve task states across independent windows (Wu et al., 2020). Typical tasks in this category include forecasting and classification under fixed input lengths (Qiao et al., 2026; Xie et al., 2026a). Recent work has broadened the scope of the field to include one-shot time series understanding and reasoning (Kong et al., 2025; Guan et al., 2026a).

However, this offline modeling paradigm has two major limitations. (1) Fixed-window, post-hoc analysis is poorly aligned with applications that require timely decisions as measurements arrive. For instance, modern monitoring systems generate high-frequency observations that necessitate online analysis, ranging from millisecond-level telemetry in IoT scheduling networks (Zanbouri et al., 2025) to minute-level traffic flow data critical for real-time urban dispatching (Zou et al., 2026). (2) This paradigm discards valuable contextual information beyond the predefined temporal boundaries. For example, streamflow forecasting and flood-risk estimation draw on historical streamflow records and previously observed flood events, which may lie far outside the immediate input window (Chandra et al., 2024). Accordingly, an appropriate modeling paradigm should align computation with continuous data arrival while retaining and reasoning over accumulated temporal context.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The broader multimodal learning community has recently begun addressing analogous requirements in continuously generated audio, video, and embodied intelligence (Lin et al., 2026; Zhang et al., 2026; Black et al., 2025; Wang et al., 2026). Audio-Interaction (Xie et al., 2026b) adopts an always-on perceive–decide–respond loop, while AURA (Lu et al., 2026) continuously processes video streams for real-time question answering and proactive assistance. Vinci2 (Sitong et al., 2026) reasons over accumulated egocentric context to determine whether and when assistance is warranted. These studies represent a paradigm shift from one-shot analysis with isolated inputs to continuous understanding supported by persistent state. Yet, an equivalent setting has not been systematically formulated or evaluated for time series intelligence (TSI). We therefore introduce Continuous & Live Time Series Understanding (CL-TSU), a new paradigm where a model operates continuously over open-ended streams and responds dynamically as observations and instructions evolve.

However, two key limitations hinder the development of CL-TSU: **(1) The lack of dedicated data systematically targeting CL-TSU capability.** The vast majority of time series evaluations remain largely confined to the one-shot setting (Aksu et al., 2024; Guan et al., 2026b). Although some studies on real-time systems evaluate models by processing streams continuously, they primarily focus on timeliness and latency (Ren et al., 2019; Alghieth, 2025), largely ignoring whether earlier observations influence later decisions. Consequently, such tasks do not measure a model's ability to maintain context, update internal states, and reason across a continuing stream. **(2) The architectural paradigm required to support CL-TSU remains largely unexplored.** Currently, processing streaming time series primarily relies on executing predefined code over the data stream (Cugola & Margara, 2012; Maler & Nickovic, 2004). These approaches struggle to capture long-term temporal dependencies and lack the flexibility to perform continuous multimodal understanding. Although some recent agent frameworks attempt to introduce capabilities for automatically updating predefined code (Qiao et al., 2024; Gu et al., 2025), these methods fundamentally lack a native understanding of the time series modality. This leaves CL-TSU without a unified modeling framework capable of natively processing continuous time series streams and maintaining persistent memory over time.

Filling these gaps drives us to formulate CL-TSU first. We argue that CL-TSU goes beyond faster one-shot analysis and requires a systematic assessment of a model's ability to retain and utilize historical context over time. Guided by this principle, we introduce **LIVETS-SUITE**, the first data suite for CL-TSU, which covers six core capabilities across 18 tasks. The evaluation begins with *Instantaneous Conditions (C1)* in current observations and extends to *Sequential Conditions (C2)* requiring recent historical reasoning. To assess persistent state, it challenges models with *Event Tracking (C3)* for ongoing records and *Event Reference (C4)* to retrieve distant historical data. Finally, LIVETS-SUITE evaluates dynamic adaptability via *Online Instructions (C5)* for mid-stream queries and rule changes, culminating in *Ultra-Long Reliability (C6)* to test sustained execution. Building on this data foundation, we present **TIMEOMNI-LIVE**, the first streaming foundation model for CL-TSU. We introduce *(a) hybrid time series encoding* to natively support the time series modality. To allow subsequent decisions to be informed by earlier related observations, we design a *(b) stream-persistent hybrid memory* mechanism, which retains memory of both textual and time series modalities. Furthermore, we introduce *(c) elastic stream-inference synchronization* to reduce processing lag. During inference, TIMEOMNI-LIVE generates responses and updates its memory while continuously processing the incoming data stream. Our contributions lie in three aspects:

1. **New Paradigm.** We formulate Continuous & Live Time Series Understanding (CL-TSU), a novel setting that shifts the paradigm from conventional offline, one-shot analysis to maintaining a persistent, stateful understanding over continuous time series streams.
2. **New Datasets and Testbed.** We introduce LIVETS-SUITE, the first data suite supporting both the training and evaluation of CL-TSU models. It covers 18 tasks across six capabilities, with over 5K streams totaling 2.2B timesteps and 15K annotated response events.
3. **New Model and Empirical Findings.** We present TIMEOMNI-LIVE, the first native foundation model for CL-TSU, which integrates hybrid time series encoding, stream-persistent hybrid memory, and elastic stream-inference synchronization. Evaluations show that TIMEOMNI-LIVE outperforms frontier LLMs on the in-distribution test set (90.0 vs. 79.3 for Opus 5). Notably, TIMEOMNI-LIVE is highly robust to instruction paraphrasing (a minimal score drop of 1.0, compared to 17.6 for Opus 5). Additionally, we validate its practical efficiency by evaluating deployment performance on a real edge device (RTF 0.358 on NVIDIA Jetson AGX Orin).

2 Related Work

Large Time Series Models. Time series research has increasingly shifted from task-specific architectures toward general-purpose pre-trained models. **Time series foundation models (TSFMs)** use large and diverse corpora for zero-shot forecasting (Jin et al., 2024; Shi et al., 2025; Zhou et al., 2026) and, more recently, general-purpose classification (Goswami et al., 2024; Feofanov et al., 2026; Xie et al., 2026a). Recognizing that other modalities can provide complementary information, the community subsequently developed **time series language models (TSLMs)** and multimodal architectures that align temporal data with LLMs for understanding and complex reasoning (Kong et al., 2025; Guan et al., 2026a). Despite these impressive capabilities, the underlying modeling paradigm remains fundamentally one-shot. Although recent efforts such as ARGOS (Gu et al., 2025) attempt to address continuous scenarios by utilizing LLM agents to generate executable code, this generation process is fundamentally offline. At runtime, the system relies on fixed code execution, making online user adaptation difficult (Mohammed & Mohsen, 2026). Furthermore, such approaches remain confined to local analysis around the current timestep. They do not accumulate historical evidence across sequential inputs and are incapable of handling cross-window comparative reasoning.

Streaming Multimodal Models. To process continuously generated data modalities, such as video and audio, the broader multimodal community has begun moving from static clips and utterances toward continuously operating models. In the vision domain, streaming video models incrementally process incoming frames to support online understanding, event tracking, and proactive assistance (Chen et al., 2024; Lu et al., 2026; Zhang et al., 2026; Wang et al., 2026). In audio, real-time and always-on models continuously perceive acoustic streams and interact without predefined segment boundaries (Défossez et al., 2024; Xie et al., 2026b; Lin et al., 2026). To support these continuous streams without exhausting GPU memory, various system-level optimizations have emerged. Most notably, approaches like StreamingLLM (Xiao et al., 2024) employ KV cache eviction strategies and attention sinks to maintain stable generation in theoretically infinite contexts. However, these optimizations primarily focus on maintaining decoding efficiency and low latency by systematically discarding historical tokens (Tong et al., 2026). They largely neglect the necessity of a persistent memory mechanism capable of preserving exact historical evidence and long-term task states. Furthermore, these general-purpose multimodal models fundamentally lack a native understanding of time series waveforms. Consequently, maintaining a persistent memory over continuous time series streams remains an unexplored frontier.

3 Methodology

Problem Definition. We define continuous & live time series understanding as stateful interaction over an open-ended stream. In conventional one-shot tasks, a model receives a bounded sequence $X_{1:T} = (x_1, \ldots, x_T)$ and an instruction q, produces a single output $y = f(X_{1:T}, q)$, and then terminates. In contrast, a CL-TSU system processes an open-ended stream $X = (x_1, x_2, \ldots)$ under an initial instruction q, maintains persistent state, and determines whether and what to respond at each new observation. Formally: $(s_t, a_t) = H(s_{t-1}, x_t, q, u_t)$, where $a_t \in \mathcal{Y} \cup \{\emptyset\}$. Here, s_t represents the persistent task-relevant state; x_t is the current chunk of observations; q is the standing instruction; u_t is an optional mid-stream query or rule update, with $u_t = \emptyset$ when no update is received. The action a_t is either a response in the response space $\mathcal{Y}$, or $\emptyset$, indicating that the system remains silent at step t. Unlike static models, a CL-TSU system is characterized by three properties: **(1) Open-ended processing**, continuously handling incoming observations without a predefined termination point; **(2) Stateful historical reasoning**, maintaining and updating the internal state s_t so that later decisions can use information accumulated across earlier windows; and **(3) Dynamic adaptability**, incorporating asynchronous mid-stream queries or rule changes u_t on the fly.

3.1 LiveTS-Suite

To address the scarcity of data for CL-TSU, we construct LiveTS-Suite, the first comprehensive data suite tailored to this setting and designed to support both model training and evaluation. Its construction is guided by the three defining properties introduced in Section 3. As shown in Figure 1, LiveTS-Suite contains 18 tasks spanning six core capabilities, organized along two axes. The vertical axis, **stream length**, dictates the required duration of continuous operation, assessing

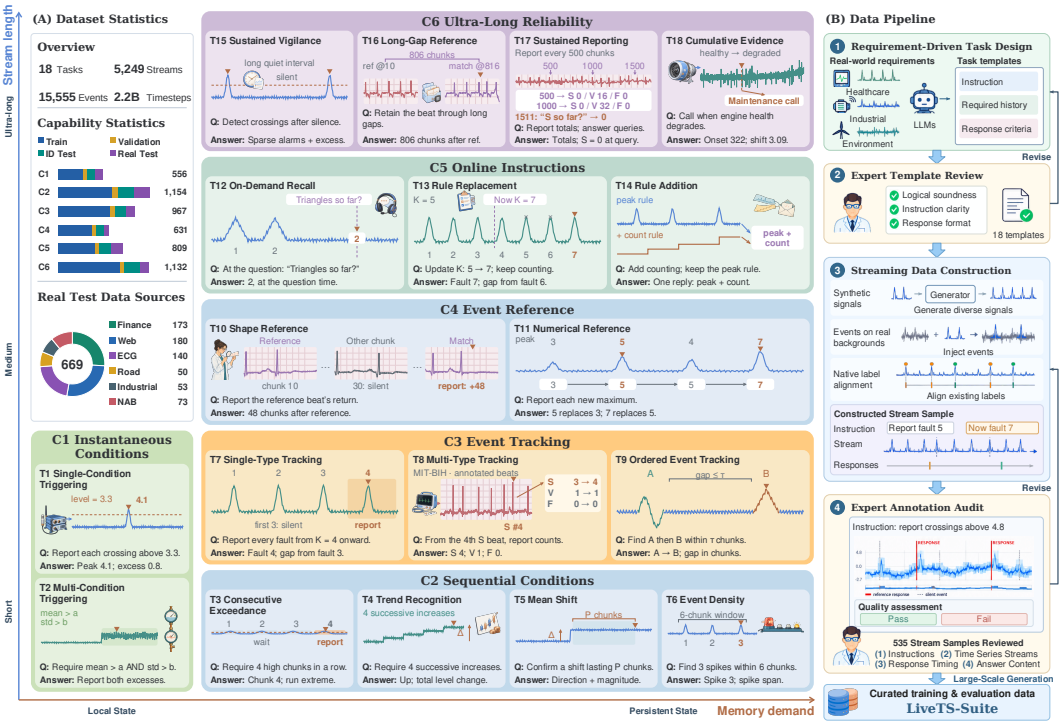


Figure 1: **Overview of LIVETS-SUITE.** The 18 tasks span six capabilities, organized by stream length (vertical) and memory demand (horizontal). Each task illustrates the instruction, the open-ended stream, and example responses. **(a)** Dataset statistics summarize stream counts by capability and Real Test data sources (see Appendix B.2 for details). **(b)** Data pipeline details the four-stage data construction process used to generate large-scale training and evaluation data for CL-TSU.

the capacity for *open-ended processing*. More importantly, distinguishing our paradigm from existing offline tasks, the horizontal axis, **memory demand**, dictates what historical information must be retained and how it supports current decisions, directly evaluating *stateful historical reasoning*. Starting with conditions in current observations (**C1: Instantaneous Conditions**), the suite extends to conditions involving recent history (**C2: Sequential Conditions**), persistent event records (**C3: Event Tracking**), and historical references for subsequent judgments (**C4: Event Reference**). It subsequently introduces dynamic queries and rule changes during the stream (**C5: Online Instructions**) to explicitly test *dynamic adaptability*, culminating in tests of sustained reliability over ultra-long streams (**C6: Ultra-Long Reliability**). LIVETS-SUITE comprises 5,249 streams across the Train, Validation, ID, and Real Test splits, with 15,555 annotated events and 2.2B timesteps. Experts reviewed all 18 task templates and audited 535 stream instances via a dedicated interface, including the numerical-shift and instruction-paraphrasing test sets designed to evaluate model generalization (Figure 4).

3.1.1 CAPABILITIES AND TASKS

LIVETS-SUITE decomposes CL-TSU into 18 specific tasks to systematically evaluate a model's ability to maintain persistent states over data streams. Dataset statistics are provided in Figure 1(a). Below, we detail the tasks. **Instantaneous Conditions (C1).** This capability establishes condition recognition within the current chunk, from a single criterion to a conjunction of criteria. *Single-Condition Triggering **(T1)*** reports a signal crossing a specified threshold, while *Multi-Condition Triggering **(T2)*** requires multiple properties (e.g., mean and variability) to satisfy joint constraints.

Sequential Conditions (C2). Extending local judgments to recent history, C2 covers four conditions involving persistence, direction, mean-level change, and event density. *Consecutive Exceedance **(T3)*** recognizes a sustained run above a threshold; *Trend Recognition **(T4)*** identifies successive changes in one direction; *Mean Shift **(T5)*** recognizes a sustained departure from the preceding mean; and *Event Density **(T6)*** identifies a concentration of events within a given interval.

Event Tracking (C3). Once events are detected, their records must remain available to inform later decisions. *Single-Type Tracking **(T7)*** counts recurring events of one type and reports their count and inter-event interval when the instructed threshold is reached. *Multi-Type Tracking **(T8)*** maintains separate counts for multiple event types, such as ECG beat classes, and reports them when the designated class triggers a response. *Ordered Event Tracking **(T9)*** captures their temporal relationships by identifying a prescribed sequence of event types within an allowed interval.

Event Reference (C4). Historical observations also serve as references for interpreting new inputs. C4 distinguishes shape and numerical references: *Shape Reference **(T10)*** retains an early waveform example to recognize its recurrence, whereas *Numerical Reference **(T11)*** compares incoming values with the historical extremes and updates the record when a new high or low is reached.

Online Instructions (C5). Beyond the initial instruction q, C5 introduces mid-stream requests u_t. *On-Demand Recall **(T12)*** responds to asynchronous queries, such as requests for the count of a specified event type; *Rule Replacement **(T13)*** applies a revised rule to the existing event state; and *Rule Addition **(T14)*** introduces a new rule while preserving the previously active rule. These tasks cover querying, revising, and extending the requirements governing a continuous stream.

Ultra-Long Reliability (C6). C6 extends the capabilities along the stream-length axis to test reliability over prolonged operation. *Sustained Vigilance **(T15)*** tests responsiveness to sparse events amid long quiet periods; *Long-Gap Reference **(T16)*** evaluates reference retention across distant recurrences; *Sustained Reporting **(T17)*** reports cumulative counts at scheduled intervals while answering intervening queries; and *Cumulative Evidence **(T18)*** accumulates subtle deviations to capture gradual shifts that remain hidden within individual chunks. Unlike the localized mean shift in T5, this trigger relies strictly on long-term accumulated evidence. Collectively, by tightly coupling stream length with memory demand, LIVETS-SUITE operationalizes the CL-TSU formulation, requiring models to maintain and dynamically update states over time.

3.1.2 STREAMING DATA CONSTRUCTION AND EVALUATION

Streaming Data Pipeline. Our data construction comprises four stages (Figure 1(b); details in Appendix B.1). **(1) Requirement-Driven Task Design.** LLMs survey real-world continuous time series understanding requirements and define task templates specifying instructions, required history, and response criteria. **(2) Expert Template Review.** Human experts review all 18 task templates for logical soundness, instruction clarity, and response format, revising them as needed. **(3) Streaming Data Construction.** We construct streams through *(a) synthetic signal generation*, *(b) event injection into real-world backgrounds*, and *(c) native label alignment*. **(4) Expert Annotation Audit.** PhD-level annotators use a review interface to audit 535 sampled stream instances, verifying consistency among the instructions, time series stream, response timing, and answer content. Failed cases are revised before the validated pipeline is used to generate training and evaluation data at scale.

Streaming Evaluation Protocol. We assess models along two dimensions: **response quality** and **real-time efficiency**. Quality demands temporal precision and exact information, measured by multiplying *Timing F1* and *Content Accuracy* into a unified ***Joint Score***. Efficiency tracks operational overhead (including *real-time factor*, *chunk latency*, *input tokens / 1k chunks*) and is evaluated exclusively on locally deployable models. Appendix B.3 details all matching and scoring mechanics.

3.2 TIMEOMNI-LIVE

TIMEOMNI-LIVE realizes continuous time series understanding through the three-step **Read-Decide-Repeat** loop illustrated in Figure 2(a). At each iteration, TIMEOMNI-LIVE reads an incoming time series chunk together with the existing context and then decides whether to remain silent or initiate a response. If a response is triggered, the same backbone generates it while subsequently arriving chunks are buffered; the loop then processes the buffered and newly arriving chunks in order. Although the architecture is backbone-agnostic, we instantiate it with Qwen3.5-4B (Qwen Team, 2026); Figure 3 provides a detailed backbone comparison. The following three core mechanisms reliably and efficiently support operation over open-ended streams.

Hybrid Time Series Encoding. As shown in Figure 2(b), each incoming time series chunk is processed through two complementary modalities to capture both natively temporal patterns and exact numerical values. A frozen time series encoder (Feofanov et al., 2026) extracts time series

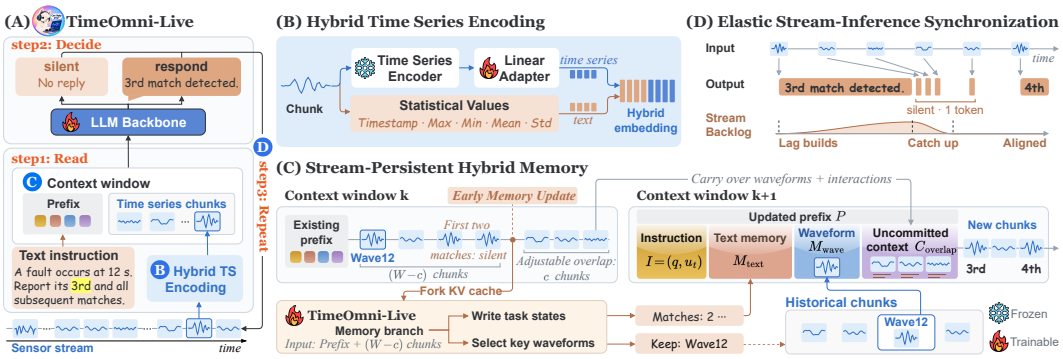


Figure 2: **TIMEOMNI-LIVE architecture. (a)** The continuous Read-Decide-Repeat inference loop. **(b)** Hybrid time series encoding combining continuous waveform embeddings with explicit textual statistics. **(c)** Stream-persistent hybrid memory retaining both text states and selected waveforms, utilizing early KV-cache forking to prevent stalling. **(d)** Elastic stream-inference synchronization that buffers chunks during response generation and realigns via fast single-token silence decisions.

features, which a trainable adapter aligns with the backbone's input dimensions. In parallel, a textual representation provides the chunk's timestamp, maximum, minimum, mean, and standard deviation. The hybrid-modality embedding is then fed into the backbone alongside the existing context.

Stream-Persistent Hybrid Memory. A bounded context cannot retain an entire stream, yet later decisions may depend on earlier events, reference waveforms, or rule updates. As shown in Figure 2(c), our hybrid memory preserves two complementary forms of information in the context prefix. The text modality stores compact task states (e.g., event counts and reference thresholds), while abstracting away signal details. The time series modality retains selected historical waveform embeddings, preserving fine-grained shape information for subsequent comparisons. For each context window **k** containing W new chunks, the memory branch forks the KV cache after the first $W{-}c$ chunks, rather than waiting until the window ends. The forking provides a stable context snapshot from which the model asynchronously generates an updated textual state M_{text} and selected waveform embeddings M_{wave}. Meanwhile, the main branch continues processing the remaining c chunks. The adjustable overlap length c therefore controls how early the memory update begins: a larger c gives slower hardware more time to complete the update, at the cost of carrying more overlap into the next window. Because the memory branch forks before the window ends, its updated state does not include the remaining c uncommitted chunks and their interactions. We thus carry these chunks forward as an overlap C_{overlap}, containing their raw waveforms and interaction history. The next context window **k+1** begins with the updated prefix $P = [I, M_{\text{text}}, M_{\text{wave}}, C_{\text{overlap}}]$, where $I = (q, u_t)$ denotes the standing instruction and optional rule update. Newly arriving chunks are then appended to P in order. By preserving the post-fork context explicitly, this asynchronous memory mechanism allows stream processing to continue without waiting for the memory update to finish.

Elastic Stream-Inference Synchronization. Continuous chunks arrive at fixed intervals, but inference latency varies depending on whether the model is silent or generates a response. As shown in Figure 2(d), chunks arriving during response generation are buffered and subsequently processed in order, preventing observation loss. Silence decisions require only one output token, allowing the model to clear the backlog and catch up with the stream when processing is faster than arrival.

Training. We train only the time series adapter and LoRA parameters of the LLM backbone. **Training samples and objectives.** To mirror continuous inference, each training window contains the prefix P followed by up to W incoming chunks. We use two forms of supervision: ***(a) Response samples*** supervise chunk-wise silence or response decisions and reference responses; ***(b) Memory samples*** supervise the updated text state and indices of historical waveforms to retain. Memory targets are derived deterministically from reference events. Token-level cross-entropy applies only to these targets; input and context tokens are masked from the loss. **Tailored Training Designs.** We use three training strategies for CL-TSU: **(1) Silence down-weighting.** To prevent the $> 95\%$ of non-response chunks from biasing the model toward permanent silence, we down-weight the loss on silence tokens by a factor of 0.2 (versus 1.0 for other target tokens). **(2) Memory-prefix perturbation.** To prevent self-written memory errors from compounding across continuous windows, we

artificially perturb the memory prefix with detectable defects in 30% of training samples, forcing the model to repair imperfect states rather than blindly trusting the prefix. **(3) Variable overlap c.** To avoid tying the model to a fixed hardware assumption, we dynamically sample the overlap length $c \in \{10, 15, 20\}$ per stream during training. This enables flexible adjustment of the overlap length c at deployment, giving memory updates more time to finish before the next window and reducing stalls. Table 3 shows the effect of c on a real edge device. Appendix C provides full training details.

4 EXPERIMENTS

Baselines. We compare TIMEOMNI-LIVE against leading models in two categories: **(1) General-purpose LLMs**, including proprietary models (Opus 5, Sonnet 5) and five open-source LLMs (DeepSeek-V4.1-Flash, MiniMax-M3, Qwen3.8-27B, Qwen3.5-9B, Qwen3.5-4B); and **(2) Time series language models**, TimeOmni-1 (Guan et al., 2026a) and TS-Reasoner (Yu et al., 2025).

Metrics. For **response quality**, we report the Joint Score $S = \frac{100}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} F_j C_j$ to capture both response timing and content accuracy over the evaluated task subset $\mathcal{J}$. For each task $j \in \mathcal{J}$, F_j is the mean per-stream timing F1, and C_j is the mean per-stream content accuracy over streams with matched responses ($C_j = 0$ if no responses match). For **real-time efficiency**, we evaluate locally deployable models using three metrics: (1) Processed input tokens per 1,000 chunks; (2) Real-time factor (RTF), the active processing time divided by stream replay duration; and (3) p95 chunk latency, the wall-clock delay from data arrival to completed processing, including queueing. Appendix B.3 details these metrics, and Appendix E illustrates the task-specific scoring criteria.

4.1 MAIN RESULTS

Setup. To ensure a fair comparison in Table 1, we evaluate the baselines under two specific settings: **(1) Code execution.** Unless explicitly marked as streaming, all baseline models generate executable programs that run in a sandbox. We adopt this protocol for two reasons: ***(a) serializing numerical chunks as text is costly*** (averaging 2.56M *tokens/stream* with the Qwen3.5 tokenizer), and ***(b) direct numerical prompting yields far weaker results*** (Qwen3.5-4B scores only 0.2/0.4 on ID/Real Test in streaming mode, versus 2.5/2.6 when writing code). **(2) Feedback-Driven Curriculum.** Because baseline performance improves with execution feedback and additional examples, a conventional zero-shot evaluation may underestimate these models. We therefore allow baselines to iteratively revise their programs using execution feedback. Figure 7 shows the resulting gains, which vary across models. All code baselines follow the same schedule, and we report their final converged results. **Results. Strong CL-TSU performance.** With only 4B parameters and purely streaming inference, TIMEOMNI-LIVE achieves an ID Test score of 90.0, surpassing the strongest code baseline (Opus 5) by **10.7** points. On the entirely unseen, out-of-distribution (OOD) Real Test, it achieves 56.1, clearly outperforming DeepSeek-V4.1-Flash (50.9) while remaining substantially stronger than other streaming baselines such as Qwen3.5-4B (streaming) and TS-Reasoner-7B. **Robust instruction following.** Paraphrasing reduces its score by only 1.0 point (89.5 to 88.5; Figure 4(a)), demonstrating the robustness to variations in instruction wording.

Table 1: Joint Score S on in-distribution (ID) and out-of-distribution Real Testbeds, with results aggregated from 18 tasks into six capabilities. Detailed per-task Timing F1 F_j and Content Accuracy C_j are provided in Tables 6 and 7 of Appendix D. **Red**: the best, Blue: the 2nd best.

Method	Base LLMs	C1: Instantaneous Conditions		C2: Sequential Conditions		C3: Event Tracking		C4: Event Reference		C5: Online Instructions		C6: Ultra-Long Reliability		Overall	
		ID	Real	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real
Proprietary Models															
Opus 5		98.0	98.0	**96.1**	**95.0**	86.6	**71.4**	74.4	**62.1**	49.0	**56.7**	72.8	28.3	79.3	**66.5**
Sonnet 5		**100.0**	**100.0**	88.3	90.6	39.4	33.9	75.6	41.4	50.7	50.4	71.5	35.3	70.0	57.7
Open-Source Models															
DeepSeek-V4.1-Flash		**100.0**	**100.0**	87.8	89.0	64.9	62.1	0.0	4.5	0.0	0.0	62.7	**41.5**	55.4	50.9
MiniMax-M3		99.9	**100.0**	82.7	87.3	13.6	16.8	22.0	0.0	47.3	50.6	66.6	25.9	56.9	47.5
Qwen3.8-27B		54.2	46.0	56.4	58.4	34.9	33.2	17.8	25.7	38.1	41.4	50.3	27.3	43.9	39.4
Qwen3.5-9B		53.4	49.4	8.2	8.4	0.0	0.0	0.0	0.0	11.6	17.8	0.0	2.6	9.7	10.9
Qwen3.5-4B		9.3	7.8	6.7	7.1	0.0	0.0	0.0	1.4	0.1	0.1	0.0	0.0	2.5	2.6
Qwen3.5-4B (streaming)		2.1	0.7	0.0	0.8	0.0	0.4	0.0	0.0	0.0	0.3	0.0	0.0	0.2	0.4
Time Series Language Models															
TimeOmni-1-4B	Qwen3.5-4B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.3	5.4	0.0	0.0	0.7	0.9
TS-Reasoner-7B (streaming)	Qwen2.5-7B	1.0	0.8	1.1	1.5	0.1	0.1	0.0	0.1	0.5	0.4	0.0	0.0	0.5	0.5
Ours															
TIMEOMNI-LIVE	Qwen3.5-4B	87.6	83.9	92.6	80.4	**95.2**	41.9	**93.8**	54.7	**84.2**	43.7	**86.9**	38.3	**90.0**	56.1

4.2 More Analysis

Ablation Studies. Setup. Table 2 ablates two core components. We uniformly sample five complete streams per task per split (180 streams in total, 11.2% of the full ID/Real evaluation set), using the same samples for all variants. ***(a) Hybrid Time Series Encoding:*** Variants are trained under a strictly fixed 2,500-step budget to isolate modality contributions without confounding by training length. ***(b) Stream-Persistent Hybrid Memory:*** Interventions are applied directly to the main checkpoint without retraining. **Results. Complementary modalities.** The hybrid encoding of both time series and textual modalities achieves the highest overall scores. The closeness of the statistics-only variant is expected: Some tasks can be effectively solved from numerical summaries alone, whereas waveform embeddings remain essential for shape-sensitive tasks in C4. **Memory ablations validate the capability organization.** The full memory mechanism consistently outperforms its ablated variants. For C1–C2, the required evidence remains within the backbone's local context window, limiting the need for cross-window memory, so all variants perform similarly. The benefits become pronounced for C3–C6: removing text memory reduces the C3 score from 95.6 to 30.5. Removing historical waveforms has only a small effect on overall performance because relatively few tasks are shape-sensitive, but the substantial decrease on C4 confirms their importance for cross-window waveform comparisons. These results support the suite's capability organization, where memory demand increases from local condition recognition (C1–C2) to persistent state tracking and historical references (C3–C6). Detailed scores are provided in Appendix D.2.

Backbone Analysis. Setup. We compare four language backbones of comparable scale (3B–4B) under identical training settings, evaluated on the same 180-stream subset described above. **Results. Stronger LLMs yield stronger CL-TSU.** Figure 3 shows that newer backbones consistently improve overall scores on both ID and Real Test, suggesting that our framework can readily benefit from future advances in language backbones. **Lightweight design enables low-cost upgrades.** This trend highlights a key advantage of our 4B-parameter architecture: it allows the framework to be easily and cost-effectively upgraded as new open-source models emerge. Detailed capability scores and training convergence curves are provided in Appendix D.3.

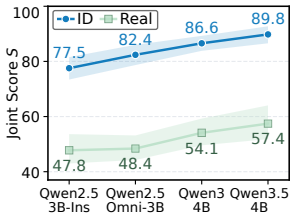
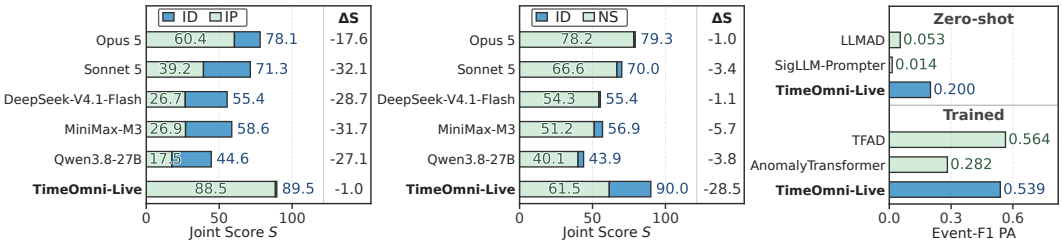


Figure 3: Backbone scores on the 180-stream test subset, with 95% confidence bands.

Generalization. Setup. We evaluate three forms of generalization: ***(a) Instruction paraphrasing***, which rewrites instruction wording while preserving task requirements; ***(b) Numerical shifts***, which introduce new numerical ranges and waveform families; and ***(c) Third-party evaluation*** on KPI (Li et al., 2022), a real-world anomaly detection dataset with manually labeled incidents. **Results.** Figure 4 summarizes the outcomes. **Strong instruction generalization.** TimeOmni-Live exhibits exceptional robustness to paraphrased instructions compared to brittle code execution baselines, suffering a minimal performance drop ($\Delta S = -1.0$). **Competitive numerical generalization.** Under numerical shifts, TimeOmni-Live achieves a Joint Score of 61.5 and outperforms DeepSeek-V4.1-Flash (54.3). This larger ΔS is expected: native language models like TimeOmni-Live rely on learned parametric priors for numerical reasoning, making them less robust to out-of-distribution numerical ranges than deterministic sandbox code execution. **Effective third-party transfer.** On KPI, TimeOmni-Live surpasses zero-shot LLM baselines (Event-F1 PA 0.200 vs. 0.053 for LLM-AD (Liu et al., 2025)) and achieves competitive performance against specialized deep learning methods when adapted (0.539 vs. 0.282 for AnomalyTransformer (Xu et al., 2022)).

Table 2: Encoding and memory ablations. Encoding variants are each trained separately for exactly 2,500 steps; memory variants use the main checkpoint without retraining. **Bold**: the best.

Variant	C1: Instantaneous Conditions		C2: Sequential Conditions		C3: Event Tracking		C4: Event Reference		C5: Online Instructions		C6: Ultra-Long Reliability		Overall	
	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real
(a) Hybrid Time Series Encoding														
TimeOmni-Live (trained for 2,500 steps)	83.1	**84.9**	89.3	**77.2**	97.9	**45.3**	**93.8**	53.9	76.8	54.8	**76.3**	43.9	**85.6**	**59.0**
Time Series Embeddings Only	11.7	15.0	76.5	20.5	**98.7**	19.2	72.3	25.0	74.1	21.6	48.9	13.2	66.0	18.7
Statistical Values Only	**85.6**	74.9	**95.5**	73.4	84.5	38.8	76.0	**56.8**	**83.3**	**59.4**	65.1	**47.1**	81.6	57.8
(b) Stream-Persistent Hybrid Memory														
TimeOmni-Live	**88.1**	**84.9**	93.2	**77.1**	**95.6**	**45.0**	**93.8**	**52.2**	**79.4**	**52.8**	**88.7**	39.5	**89.8**	**57.4**
w/o Text Memory	**88.1**	**84.9**	93.2	**77.1**	30.5	27.8	56.0	28.6	32.8	24.8	53.1	37.6	59.1	46.9
w/o Waveform Memory	**88.1**	**84.9**	**93.4**	**77.1**	**95.6**	**45.0**	65.7	48.1	**79.4**	**52.8**	68.9	40.2	82.3	57.1
w/o Uncommitted Context	**88.1**	**84.9**	93.2	**77.1**	54.6	25.4	74.5	48.0	48.5	26.1	71.5	**40.7**	71.9	49.5
Memory Delayed by One Window	**88.1**	**84.9**	93.2	**77.1**	42.6	35.9	68.6	35.2	39.7	32.6	58.1	38.4	64.8	50.4



(a) Instruction paraphrasing (IP) (b) Numerical shift (NS) (c) Third-party evaluation

Figure 4: Generalization evaluations. (a) Instruction paraphrasing on 180 paired ID test streams. (b) Numerical shift on the entire ID and NS test sets, where ΔS is the Joint Score change. (c) Third-party evaluation on KPI using mean per-series Event-F1 PA. Details are provided in Appendix D.4.

Elastic Scheduling. **Setup.** We test ***(a) elastic stream-inference synchronization*** and ***(b) early memory update***. Figure 5 ablates *w/o Elastic Synchronization* (fixed-rate processing), *w/o Early Memory Update* (window-end updates), and *w/o Both*. All replay the same complete stream at 1 chunk/s. **Results. Elasticity ensures rapid recovery.** TIMEOMNI-LIVE maintains low latency and quickly recovers after response peaks; without it, delay gradually accumulates. **Early memory update prevents boundary stalls.** Without it, window boundaries cause temporary stalls, and removing both mechanisms produces severe increases in delay. Full-stream statistics and backlog curves appear in Appendix D.5.

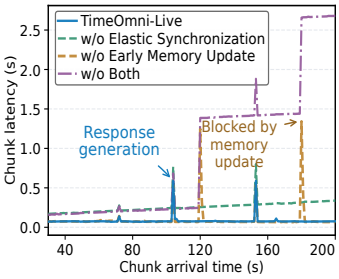


Figure 5: Scheduling ablations.

Efficiency. **Setup.** We deploy three streaming models on dedicated H100 GPUs, feeding time series at 1 chunk/s. We evaluate them on 54 uniformly sampled Real Test streams using the three real-time efficiency metrics, where RTF below 1 indicates sufficient average processing speed. **Results. Efficient streaming inference.** Compared with the Qwen3.5-4B streaming baseline, TIMEOMNI-LIVE reduces processed input tokens by 99.3% and p95 latency by a factor of 206, while achieving a higher Joint Score (Table 3).

Table 3: Efficiency and edge deployment. Details in Appendix D.5.

Model	S ↑	Input tokens / 1k chunks↓	RTF↓	p95 latency (s)↓
(a) Streaming efficiency on H100				
Qwen3.5-4B (streaming)	2.0	8,558,122	0.749	16.615 (206×)
TS-Reasoner-7B (streaming)	0.3	594,623	0.276	0.507 (6.3×)
TIMEOMNI-LIVE ($c = 15$)	**53.8**	**56,541**	**0.117**	**0.081** (1×)
(b) Jetson AGX Orin				
TIMEOMNI-LIVE ($c = 10$)	53.7	53,164	0.375	2.246 (3.8×)
TIMEOMNI-LIVE ($c = 15$)	54.3	56,511	0.358	0.649 (1.1×)
TIMEOMNI-LIVE ($c = 20$)	55.3	59,890	0.360	0.592 (1×)

Edge Deployment. **Setup.** We deploy the TIMEOMNI-LIVE checkpoint on a Jetson AGX Orin edge device without quantization, replaying the identical 54 streams used above. **Results. Real-time edge capability.** At $c = 15$, the Jetson achieves a comparable S to the H100 while maintaining an RTF of 0.358 (well below 1), achieving sub-second real-time processing (Table 3). **Effectiveness of variable overlap.** Increasing the overlap c from 10 to 20 provides more buffer time for memory updates, which reduces p95 latency by 3.8×. Ultimately, $c = 20$ yields both the highest score and the lowest latency on the Jetson. Appendix D.5 reports deployment details and overlap sensitivity.

5 CONCLUSION

The work introduces continuous & live time series understanding (CL-TSU), a paradigm that moves time series understanding beyond offline, one-shot analysis toward persistent reasoning over open-ended streams. We present **LIVETS-SUITE**, the first data suite for CL-TSU, with 18 streaming tasks spanning six core capabilities and progressively increasing demands on memory and continuous operation. We then develop **TIMEOMNI-LIVE**, the first foundation model for CL-TSU, combining hybrid time series encoding, stream-persistent hybrid memory, and elastic stream-inference synchronization. Experiments demonstrate that TIMEOMNI-LIVE, with only 4B parameters, outperforms frontier models and generalizes to paraphrased instructions, shifted numerical distributions, and third-party data. Ablations further validate the complementary roles of its encoding, memory, and scheduling components. Finally, deployment on a real-world edge device demonstrates its practical efficiency for live, stateful interaction.

Ethics Statement

Our work focuses solely on scientific challenges and does not involve human subjects, animals, or environmentally sensitive materials. We foresee no ethical risks or conflicts of interest. We are committed to upholding the highest standards of scientific integrity and ethical conduct to ensure the validity and reliability of our findings.

Reproducibility Statement

To ensure reproducibility, we provide implementation and training details (Appendix C) and examples for all 18 tasks, including task prompts and reference responses (Appendix E). Our code and model checkpoints are publicly available.

References

Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017. ISSN 0925-2312.

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation, 2024. URL https://arxiv.org/abs/2410.10393.

Manal Alghieth. DeepECG-Net: A hybrid transformer-based deep learning model for real-time ECG anomaly detection. *Scientific Reports*, 15(1):20714, 2025. ISSN 2045-2322.

Sarah Alnegheimish, Linh Nguyen, Laure Berti-Equille, and Kalyan Veeramachaneni. Large language models can be zero-shot anomaly detectors for time series? *arXiv preprint arXiv:2405.14755*, 2024.

Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1):5, 2021.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-Time Execution of Action Chunking Flow Policies, 2025.

Rohitash Chandra, Arpit Kapoor, Siddharth Khedkar, Jim Ng, and R. Willem Vervoort. Ensemble quantile-based deep learning framework for streamflow and flood prediction in australian catchments, 2024. URL https://arxiv.org/abs/2407.15882.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online Video Large Language Model for Streaming Video, 2024.

Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3), June 2012. ISSN 0360-0300. doi: 10.1145/2187671.2187677. URL https://doi.org/10.1145/2187671.2187677.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: A speech-text foundation model for real-time dialogue, 2024.

Vasilii Feofanov, Songkang Wen, Jianfeng Zhang, Lujia Pan, and Ievgen Redko. Mantisv2: Closing the zero-shot gap in time series classification with synthetic data and test-time strategies, 2026. URL https://arxiv.org/abs/2602.17868.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: a family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yile Gu, Yifan Xiong, Jonathan Mace, Yuting Jiang, Yigong Hu, Baris Kasikci, and Peng Cheng. Argos: Agentic time-series anomaly detection with autonomous rule generation via large language models. *arXiv preprint arXiv:2501.14170*, 2025.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=kOIclg7muL.

Tong Guan, Sheng Pan, Johan Barthelemy, Zhao Li, Yujun Cai, Cesare Alippi, Ming Jin, and Shirui Pan. Timeomni-VL: Unified models for time series understanding and generation. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=uUnIAtZUfX.

David J. Hill and Barbara S. Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software*, 25(9):1014–1022, 2010. ISSN 1364-8152.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

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. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1437. URL https://aclanthology.org/2025.acl-long.1437/.

Zeyan Li, Nengwen Zhao, Shenglin Zhang, Yongqian Sun, Pengfei Chen, Xidao Wen, Minghua Ma, and Dan Pei. Constructing large-scale real-world benchmark datasets for aiops. *arXiv preprint arXiv:2208.03938*, 2022.

Bin Lin, Bo Zhao, Boyang Zhang, and StepFun-Audio Team. Stepaudio 3 realtime technical report, 2026. URL https://arxiv.org/abs/2609.14005.

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.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. AURA: Always-On Understanding and Real-Time Assistance via Video Streams, 2026.

Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *International symposium on formal techniques in real-time and fault-tolerant systems*, pp. 152–166. Springer, 2004.

Abrar Mohammed and Ayedh Abdulaziz Mohsen. Real-time anomaly detection in cloud data streams: a survey of techniques, challenges, and future directions. In *2026 International Conference on Digital Transformation, Innovation & Sustainable Development (DTISD)*, pp. 1–12, 2026. doi: 10.1109/DTISD69661.2026.11668120.

G.B. Moody and R.G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001. doi: 10.1109/51.932724.

Patrick Podest, Marco Pichler, Elias Bürger, Levente Zólyomi, Bernhard Voggenberger, Wilhelm Berghammer, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex-2: Generalizing tirex to multivariate data and streaming, 2026. URL https://arxiv.org/abs/2607.01204.

Tom Pollard, Benjamin E. Moody, Li-wei H. Lehman, Brian J. Gow, Chrystinne Fernandes, Chen Xie, Alistair Johnson, Roger G. Mark, and Thomas Heldt. PhysioNet as a global platform for biomedical research. *Nature Health*, 1(8):792–795, 2026. ISSN 3005-0693.

Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang, Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, Minghua Ma, Pu Zhao, Si Qin, Xiaoting Qin, Chao Du, Yong Xu, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Taskweaver: A code-first agent framework, 2024. URL https://arxiv.org/abs/2311.17541.

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's TIME: Towards the next generation of time series forecasting benchmarks. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=79TgfXHbsK.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3009–3017. ACM, 2019. doi: 10.1145/3292500.3330680. URL http://dx.doi.org/10.1145/3292500.3330680.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Gong Sitong, Tianyu Yan, Caixin Kang, Bo Zheng, Xiang Ruan, Huchuan Lu, Kaipeng Zhang, Yoichi Sato, and Yifei Huang. Vinci2: Providing Proactive Assistance in Continuous Egocentric Videos, 2026.

Junlong Tong, Yao Zhang, Anhao Zhao, Yingqi Fan, Yunpu Ma, and Xiaoyu Shen. ProactiveLLM: Learning Active Interaction for Streaming Large Language Models. In *Forty-third International Conference on Machine Learning*, number arXiv:2606.00523, 2026. URL https://openreview.net/forum?id=sr1qgowJQ1.

Yifei Wang, Zhenkai Li, Tianwen Qian, Huanran Zheng, Zheng Wang, Yuqian Fu, and Xiaoling Wang. Streameqa: Towards streaming video understanding for embodied scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9422–9432, June 2026.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Shifeng Xie, Vasilii Feofanov, Jianfeng Zhang, Themis Palpanas, and Ievgen Redko. CauKer: Classification time series foundation models can be pretrained on synthetic data. In *The Fourteenth International Conference on Learning Representations*, pp. 68606–68634, 2026a. URL https://openreview.net/forum?id=xBW2FIfswU.

Zhifei Xie, Zihang Liu, Ze An, Xiaobin Hu, Yue Liao, Ziyang Ma, Dongchao Yang, Mingbao Lin, Deheng Ye, Shuicheng Yan, and Chunyan Miao. Audio Interaction Model, 2026b.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Fangxu Yu, Hongyu Zhao, and Tianyi Zhou. Ts-reasoner: Aligning time series foundation models with llm reasoning. *arXiv preprint arXiv:2510.03519*, 2025.

Kouros Zanbouri, Md. Noor-A-Rahim, Jobish John, Cormac J. Sreenan, H. Vincent Poor, and Dirk Pesch. A Comprehensive Survey of Wireless Time-Sensitive Networking (TSN): Architecture, Technologies, Applications, and Open Issues. *IEEE Communications Surveys & Tutorials*, 27(4): 2129–2155, 2025. ISSN 1553-877X.

Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Tfad: A decomposition time series anomaly detection architecture with time-frequency analysis. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 2497–2507, 2022.

Xichen Zhang, Guankai Li, Yinghao Zhu, Shijian Wang, Sitong Wu, Shaozuo Yu, Meng Chu, Yuan Lu, and Jiaya Jia. StreamArena: Toward Continuous, Interactive, and Long-Horizon Agentic Streaming Video Understanding, 2026.

Xinxing Zhou, Qingren Yao, Yiji Zhao, Chenghao Liu, Flora Salim, Xiaojie Yuan, Yanlong Wen, and Ming Jin. Eidos: Latent-space predictive learning for time series foundation models, 2026. URL https://arxiv.org/abs/2602.14024.

Xingchen Zou, Yuhao Yang, Zheng Chen, Xixuan Hao, Yiqi Chen, Chao Huang, and Yuxuan Liang. Traffic-r1: Reinforced LLMs bring human-like reasoning to traffic signal control systems. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21823–21838, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.995. URL https://aclanthology.org/2026.acl-long.995/.

A THE USE OF LARGE LANGUAGE MODELS

During the preparation of this manuscript, we only employed large language models as auxiliary tools. We retain full intellectual ownership of the research design, conceptual framework, experimental analysis, and scientific conclusions. Specifically, we strictly limited LLM assistance to the following specific tasks: polishing text fluency, checking grammatical consistency, and debugging sandbox execution scripts; assisting in literature searches to identify relevant prior work, without generating fictitious citations; aggregating source links for public datasets to accelerate data collection; and paraphrasing task instructions for the instruction generalization evaluation set to ensure diverse linguistic phrasing while preserving strict task requirements. All core methodologies, architectural designs, and insights presented in this work are entirely independent of LLM generative outputs.

B DATA SUITE CONSTRUCTION AND EVALUATION

B.1 FOUR-STAGE DATA CONSTRUCTION

Figure 1(b) illustrates the four stages of data construction for LIVETS-SUITE.

(1) Requirement-Driven Task Design. LLMs survey real-world requirements for continuous time series understanding and organize them into task templates. Each template specifies the standing instruction, the historical information needed for subsequent judgments, and when and what to report. These requirements guide the selection of data sources and the temporal structure of each instance.

(2) Expert Template Review. Human experts inspect the templates for all 18 tasks before stream construction. They check whether the instructions are clear, the required historical information supports the intended task, and the response conditions and answer fields are consistent. Templates that fail these checks are revised before being used to construct instances.

(3) Streaming Data Construction. Approved templates are instantiated through three complementary routes. *(a) Synthetic generation* combines white-noise, AR(1), seasonal, or drift backgrounds with task-specific pulses, level or variance changes, trends, gradual drifts, and event sequences, including distractors that do not satisfy the instruction. *(b) Background-based construction* uses MIT-BIH Arrhythmia signals (Moody & Mark, 2001), NAB series (Ahmad et al., 2017), and Yahoo Finance daily closing prices. Segments are cropped or linearly interpolated to the required length, median-centered, normalized by their standard deviation, and randomly scaled before task events are introduced. *(c) Native-label alignment* uses original beat annotations from MIT-BIH Arrhythmia (Moody & Mark, 2001) and MIT-BIH Long-Term ECG (Pollard et al., 2026), and health-state labels from N-CMAPSS (Arias Chao et al., 2021), an engine simulation dataset driven by real flight conditions. These annotations are aligned to chunks and used to derive task-specific reference responses. Reference event times and answer fields are derived from these construction records, the signals or source annotations, together with the task rules, and checked programmatically. The resulting instance contains a standing instruction, a sequence of 512-sample chunks, optional timed queries or rule updates, and timestamped reference responses. Event occurrence alone does not necessarily require a response: event order, historical gaps and the active instruction determine the response requirements. For example, T13 retains the cumulative event count across rule updates and applies the new rule to subsequent events. T12 answers later queries using past observations.

(4) Expert Annotation Audit. PhD-level human annotators reviewed 535 stream instances across all 18 tasks: 90 from Train and Validation combined, 90 from ID Test, 85 from Real Test, 90 from Numerical Shift, and 180 from Instruction Paraphrasing. Thus, 265 reviewed instances belong to the main suite splits and 270 to the two generalization evaluations in Figure 4. The review interface (Figure 6) displays the exact instruction, an overview and zoomable waveform, reference-event and non-response markers, timed queries or updates, and answer fields. Experts inspect whether these elements agree; identified problems lead to revisions of the affected instructions, signal construction or label derivation. The interface supports annotation quality assurance, not model scoring.



Figure 6: Screenshot of the annotation review interface used in Stage 4 (Expert Annotation Audit).

B.2 Data Statistics

Table 4 breaks down the 5,249 complete streams and 15,555 annotated events in LiveTS-Suite by task and split, totaling 2.2B timesteps. The same table reports maximum and average stream lengths in timesteps, rounded to integers.

Table 4: Stream counts and lengths in LiveTS-Suite. MAX and AVG pool the four splits; AVG is rounded to the nearest timestep. Events counts annotated response occasions (alarms, scheduled reports, and query answers); each response counts once regardless of its fields. †For T2, controlled generation provides precise joint-condition labels for 50 held-out synthetic Real Test streams with independently sampled condition combinations and thresholds.

Task	Train	Validation	ID Test	Real Test	Total	Events	Length (timesteps)	
							MAX	AVG
T1	164	20	50	50	284	780	36,864	30,325
T2	153	19	50	50†	272	674	36,864	30,850
T3	171	20	50	50	291	589	36,864	30,764
T4	167	20	50	50	287	608	36,864	30,647
T5	174	19	50	50	293	377	36,864	30,362
T6	164	19	50	50	283	385	36,864	30,508
T7	160	20	50	50	280	670	409,600	330,373
T8	321	20	50	10	401	1,697	409,600	236,614
T9	166	20	50	50	286	580	409,088	337,557
T10	280	20	50	9	359	1,464	409,600	257,038
T11	153	19	50	50	272	675	409,088	330,671
T12	176	20	50	50	296	895	409,600	332,098
T13	194	20	50	50	314	1,798	409,088	329,368
T14	100	11	39	49	199	1,175	409,600	335,538
T15	100	10	50	50	210	423	1,531,392	1,288,680
T16	208	10	50	11	279	1,060	2,149,376	1,525,665
T17	159	12	50	14	235	1,310	1,529,344	1,148,771
T18	312	20	50	26	408	395	1,536,000	1,091,363
Total	**3,322**	**319**	**889**	**719**	**5,249**	**15,555**	**2,149,376**	**418,831**

B.3 Streaming Evaluation Protocol

We assess models along two dimensions: **response quality** and **real-time efficiency**. Streams and instruction updates are presented in temporal order, and evaluation continues after each response.

B.3.1 Response Quality

Timing F1. For each reference event at chunk a, we match the earliest unused response within $[a, a+2]$ for T1–T17. T18 requires accumulating small deviations over a long history, similar to numerical integration. To tolerate errors in this accumulated estimate, we use the symmetric window $[a-60, a+60]$. Matching is one-to-one: unmatched responses, including duplicates, are false positives (FP), and unmatched reference events are false negatives (FN). With TP denoting matched responses, each stream i receives

$$F_i = \frac{2\mathrm{TP}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}, \tag{1}$$

with zero assigned when the denominator is zero.

Content Accuracy. We compare the requested fields of each matched response with their references. For numerical fields, relative error is $e = |\hat{y}-y|/\max(|y|, d_{\min})$, where $d_{\min}$ is a field-specific floor. The score of an individual numerical field is

$$c(e) = \begin{cases} 1, & e \le 0.05, \\ \dfrac{0.15-e}{0.10}, & 0.05 < e < 0.15, \\ 0, & e \ge 0.15. \end{cases} \tag{2}$$

T18 onset positions instead use absolute errors of 30 and 90 chunks as the full-credit and zero-credit boundaries. Discrete fields require exact agreement after normalization; missing or unparseable fields score zero. Let $\bar{c}_{ik}$ denote the mean field score of matched response k in stream i. With m_i matched responses, its stream-level Content Accuracy is

$$C_i = \frac{1}{m_i}\sum_{k=1}^{m_i} \bar{c}_{ik}, \qquad m_i > 0. \tag{3}$$

Task-specific fields are listed in Appendix E.

Joint Score and aggregation. For task j, F_j averages Timing F1 over all evaluated streams. Let $\mathcal{I}_j^+$ contain its streams with matched responses. Task-level Content Accuracy C_j and Joint Score S are

$$C_j = \frac{1}{|\mathcal{I}_j^+|}\sum_{i\in\mathcal{I}_j^+} C_i, \qquad S = \frac{100}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} F_j C_j. \tag{4}$$

We set $C_j = 0$ when $\mathcal{I}_j^+$ is empty. Capability scores average their constituent tasks; Overall averages all 18 tasks equally. Multiplication precedes averaging across tasks, and rounding is applied only to the displayed results.

B.3.2 Real-Time Efficiency

For locally deployable models, we report the three metrics under the same stream replay rate.

Input tokens / 1k chunks. The total input positions processed by the language model, including text tokens and time series embeddings, divided by the number of chunks and multiplied by 1,000. Reprocessed prefixes count each time; generated output tokens are excluded.

Real-time factor (RTF). Total active processing time divided by stream replay duration. Processing includes encoding, response generation, and memory updates, but excludes waiting for arrivals. RTF below 1 indicates sufficient average processing speed.

p95 chunk latency. The 95th percentile of the delay from chunk arrival to completed processing, including queueing. We pool chunk latencies across the evaluated streams before computing the percentile; lower values indicate faster responses to incoming data.

C Implementation Details

C.1 Training Windows and Supervision

For short tasks (T1–T6), each training window contains the whole stream. For longer streams, windows follow the inference grid with $W = 60$ new chunks each, except for a shorter final window. Candidates cover responses, instruction updates, and annotated signal events; each neighboring window is added with probability 0.35. A uniformly sampled window is added with probability 0.25 (always for T15 and T18). We retain up to six candidates, then additionally include the first and second windows with probabilities 0.30 and 0.5, respectively; the first window is always included for T18. This sampling covers different response positions and both empty memories and memories containing earlier events. We sample the 18 tasks in proportion to the square root of their sample counts, with a minimum sampling weight of $0.6\times$ the mean task weight before normalization.

For each supervised token, cross-entropy receives weight 0.2 for a silence token and 1 for other target tokens; the loss is normalized by the sum of these weights. Context tokens do not contribute to the loss. Memory-prefix perturbation is attempted with probability 0.3 for nonempty memory. Perturbations include a tally inconsistent with the retained event list, shuffled lines, stale reporting positions, or altered whitespace. Historical events are neither inserted nor deleted, and protected memory fields remain unchanged. These perturbations preserve the information needed to recover the reference state. The overlap length is sampled from $\{10, 15, 20\}$ for each stream during training; evaluation uses $c = 15$ with $W = 60$ new chunks per window.

C.2 Training Recipes and Checkpoint Selection

We train the time series adapter and rank-16 LoRA ($\alpha = 32$) on Qwen3.5-4B, keeping the time series encoder and remaining backbone parameters frozen. The adapter and LoRA contain 13.13M and 9.83M trainable parameters, respectively. Their peak learning rates are 10^{-3} and 2×10^{-4}, with an effective batch size of 32, gradient clipping at 1.0, and a sequence limit of 4,608 tokens. Each optimization run uses a fresh optimizer and a cosine schedule with 100 warm-up steps. Table 5 summarizes the main-model and encoding-ablation recipes.

Main checkpoint. Initial training selects step 2,500 using mean validation timing F1 plus $0.1\times$ content token accuracy, and stops at step 3,300 after eight validations without improvement. From the selected checkpoint, we continue training for 1,500 steps with half of samples allocated to T18, and report the final checkpoint. Memory ablations in Table 2(b) and both paraphrasing conditions reuse these weights.

Encoding ablations (Table 2(a)). The three variants are ① TimeOmni-Live (trained for 2,500 steps), ② Time Series Embeddings Only, and ③ Statistical Values Only. Each uses a fixed budget of 1,000 initial-training steps plus 1,500 continuation steps, retaining the original task-sampling distribution and its specified input channels. Only ① applies channel dropout with probability 0.25 during continuation. All three report their final checkpoints. We train ① separately from the main model used in Table 2(b).

Table 5: **Qwen3.5-4B training recipes.** Encoding ablations correspond to Table 2(a); memory ablations in Table 2(b) reuse the main checkpoint.

Setting	Main checkpoint	Encoding ablations
Seed	0	1337
Initial training	Up to 5,000 steps; selected step 2,500	Fixed 1,000 steps
Continuation	Fixed 1,500 steps	Fixed 1,500 steps
Continuation sampling	T18 share 0.5	Original task sampling
Channel dropout	0.25 in training and continuation	0 in initial training; 0.25 for hybrid in continuation; 0 for single-channel inputs
Warm-up	100 steps per optimization run	100 steps per optimization run
Validation interval	100 steps	250 steps
Continuation initialization	Initial training validation best	Initial training final
Reported checkpoint	Continuation final	Continuation final

C.3 Streaming Inference

Inference uses greedy decoding, $W = 60$ new chunks per window, and overlap $c = 15$. An example of the retained memory appears in Appendix C.3.1. The Qwen3.5 runtime permits up to 8,192 tokens to accommodate its numerical-readout tokenization. The same runtime settings are used for the main model and encoding variants, with input channels changed only for the corresponding ablations. Hardware timing measurements are described in Appendix D.5.

C.3.1 Memory

Memory stores task-dependent state and indices of waveform evidence to retain.

Memory Example and Optional Fields

T11: Numerical Reference

```
memory @c345
record:  -24.7 @c133
keep:  c133
last alarm:  @c133
```

`@c345` covers information before chunk 345. `keep: c133` retains chunk 133's waveform representation for the next window.

Optional fields in other tasks

`rule`: active rule and update position
`events`: event types and chunk indices
`tally`: cumulative counts
`due`: next reporting position
`baseline`: reference statistics
`step`: accumulation update parameters
`cusum`: accumulated evidence and onset
`streak`/`trend`: sequential-condition state

D Detailed Experimental Results

The main comparison evaluates code-executing models and TimeOmni-Live on all 18 tasks, using 889 ID and 719 Real Test streams.

D.1 Main Results

Code-generation convergence. Each code baseline iteratively refines its program using execution feedback on up to 20 training examples per task ($k \in \{1, 3, 5, 10, 15, 20\}$). Figure 7 evaluates each stage on the same 180 uniformly sampled ID/Real streams (11.2% of full evaluation). Performance generally stabilizes by $k = 20$, whose programs are used for the main comparison.

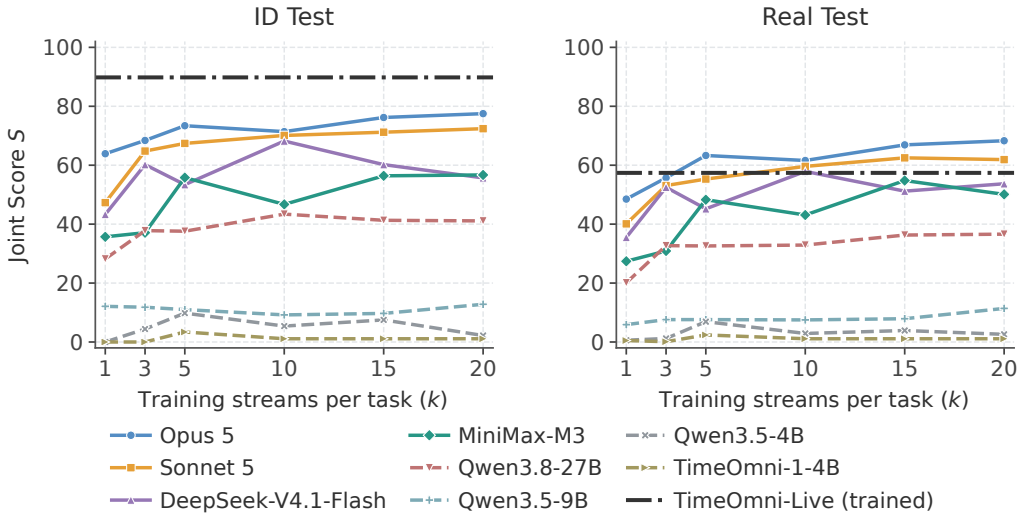


Figure 7: **Code-generation convergence on ID and Real Test.** Each stage uses k training streams per task; later stages show diminishing gains with some fluctuations. Horizontal lines mark the trained TimeOmni-Live reference.

Per-task results. Tables 6 and 7 give per-task Timing F1, Content Accuracy, and Joint Score for the ID and Real results in Table 1. Capability and Overall scores average the corresponding task-level Joint Scores.

Table 6: **ID Test: per-task main results.** Each model reports Timing F1 (F_j), Content Accuracy (C_j), and Joint Score ($S_j = 100F_jC_j$). For readability, F_j and C_j are displayed as percentages; S_j is on a 0–100 scale. Scores use unrounded values and the same evaluation results as Table 1. Overall averages each metric across all 18 tasks; its Joint Score is the mean of task scores, not the product of the aggregate metrics. TIMEOMNI-LIVE is repeated for comparison.

(a) Proprietary Models

Capability	Task	Opus 5			Sonnet 5			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	96.0	100.0	96.0	100.0	100.0	100.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	93.4	80.5	75.2
C2: Sequential Conditions	T3 Consecutive Exceedance	100.0	100.0	100.0	68.9	100.0	68.9	100.0	100.0	100.0
	T4 Trend Recognition	100.0	100.0	100.0	100.0	100.0	100.0	93.5	96.8	90.4
	T5 Mean Shift	100.0	84.2	84.2	100.0	84.2	84.2	100.0	80.7	80.7
	T6 Event Density	100.0	100.0	100.0	100.0	100.0	100.0	99.3	100.0	99.3
C3: Event Tracking	T7 Single-Type Tracking	94.8	100.0	94.8	100.0	100.0	100.0	98.0	99.0	97.0
	T8 Multi-Type Tracking	80.7	88.5	71.4	65.6	28.0	18.3	96.9	98.6	95.6
	T9 Ordered Event Tracking	93.5	100.0	93.5	0.0	0.0	0.0	93.0	100.0	93.0
C4: Event Reference	T10 Shape Reference	74.8	95.2	71.3	81.2	100.0	81.2	100.0	100.0	100.0
	T11 Numerical Reference	100.0	77.6	77.6	93.6	74.9	70.1	98.5	89.0	87.7
C5: Online Instructions	T12 On-Demand Recall	100.0	16.3	16.3	100.0	14.4	14.4	100.0	85.7	85.7
	T13 Rule Replacement	87.9	91.7	80.6	100.0	100.0	100.0	95.0	96.0	91.3
	T14 Rule Addition	88.5	56.6	50.1	78.4	48.1	37.7	99.6	75.9	75.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T16 Long-Gap Reference	73.8	100.0	73.8	73.5	100.0	73.5	100.0	100.0	100.0
	T17 Sustained Reporting	99.8	17.3	17.2	99.8	12.5	12.5	99.8	97.9	97.7
	T18 Cumulative Evidence	100.0	100.0	100.0	100.0	100.0	100.0	89.3	55.9	50.0
Overall		93.9	84.9	79.3	86.7	75.7	70.0	97.6	92.0	90.0

(b) Open-Source Models (1/2)

Capability	Task	DeepSeek-V4.1-Flash			MiniMax-M3			Qwen3.8-27B			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	90.0	100.0	90.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	100.0	100.0	100.0	100.0	99.7	99.7	20.0	92.0	18.4	93.4	80.5	75.2
C2: Sequential Conditions	T3 Consecutive Exceedance	100.0	100.0	100.0	68.9	100.0	68.9	12.7	96.1	12.2	100.0	100.0	100.0
	T4 Trend Recognition	90.0	100.0	90.0	94.5	95.9	90.7	88.0	100.0	88.0	93.5	96.8	90.4
	T5 Mean Shift	92.0	81.7	75.2	100.0	84.2	84.2	34.7	90.8	31.5	100.0	80.7	80.7
	T6 Event Density	86.0	100.0	86.0	88.8	97.8	86.8	94.0	100.0	94.0	99.3	100.0	99.3
C3: Event Tracking	T7 Single-Type Tracking	100.0	100.0	100.0	49.8	69.6	34.7	89.6	100.0	89.6	98.0	99.0	97.0
	T8 Multi-Type Tracking	6.1	37.5	2.3	21.6	28.1	6.1	27.1	52.3	14.2	96.9	98.6	95.6
	T9 Ordered Event Tracking	92.3	100.0	92.3	8.3	0.0	0.0	1.0	100.0	1.0	93.0	100.0	93.0
C4: Event Reference	T10 Shape Reference	0.0	0.0	0.0	43.9	100.0	43.9	0.0	0.0	0.0	100.0	100.0	100.0
	T11 Numerical Reference	61.7	0.0	0.0	38.5	0.0	0.0	38.0	93.5	35.5	98.5	89.0	87.7
C5: Online Instructions	T12 On-Demand Recall	0.0	0.0	0.0	95.0	20.7	19.7	98.6	13.0	12.8	100.0	85.7	85.7
	T13 Rule Replacement	0.0	0.0	0.0	100.0	100.0	100.0	79.6	100.0	79.6	95.0	96.0	91.3
	T14 Rule Addition	0.0	0.0	0.0	58.3	38.2	22.3	49.7	44.1	21.9	99.6	75.9	75.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	100.0	100.0	100.0	100.0	100.0	100.0	85.4	100.0	85.4	100.0	100.0	100.0
	T16 Long-Gap Reference	36.0	100.0	36.0	94.2	100.0	94.2	0.6	12.3	0.1	100.0	100.0	100.0
	T17 Sustained Reporting	99.8	15.0	15.0	99.8	3.0	3.0	99.8	15.6	15.6	99.8	97.9	97.7
	T18 Cumulative Evidence	100.0	100.0	100.0	70.0	98.8	69.2	100.0	100.0	100.0	89.3	55.9	50.0
Overall		64.7	63.0	55.4	74.0	68.7	56.9	56.1	72.8	43.9	97.6	92.0	90.0

Table 6: ID Test per-task main results (continued).

(c) Open-Source Models (2/2)

Capability	Task	Qwen3.5-9B			Qwen3.5-4B			Qwen3.5-4B (streaming)			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	86.7	78.6	68.1	0.0	0.0	0.0	4.3	100.0	4.3	100.0	100.0	100.0
	T2 Multi-Condition Triggering	40.9	94.9	38.8	19.6	94.9	18.6	0.0	0.0	0.0	93.4	80.5	75.2
C2: Sequential Conditions	T3 Consecutive Exceedance	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0
	T4 Trend Recognition	24.0	100.0	24.0	28.5	91.5	26.1	0.0	0.0	0.0	93.5	96.8	90.4
	T5 Mean Shift	8.9	79.7	7.1	0.0	0.0	0.0	0.0	0.0	0.0	100.0	80.7	80.7
	T6 Event Density	18.0	10.0	1.8	4.8	11.0	0.5	0.0	0.0	0.0	99.3	100.0	99.3
C3: Event Tracking	T7 Single-Type Tracking	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0	98.0	99.0	97.0
	T8 Multi-Type Tracking	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.9	98.6	95.6
	T9 Ordered Event Tracking	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	93.0	100.0	93.0
C4: Event Reference	T10 Shape Reference	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	100.0	100.0	100.0
	T11 Numerical Reference	50.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.5	89.0	87.7
C5: Online Instructions	T12 On-Demand Recall	31.7	40.2	12.8	0.0	0.0	0.0	0.0	0.0	0.0	100.0	85.7	85.7
	T13 Rule Replacement	0.0	0.0	0.0	21.7	1.6	0.3	0.8	0.0	0.0	95.0	96.0	91.3
	T14 Rule Addition	42.6	51.6	22.0	0.0	0.0	0.0	0.0	0.0	0.0	99.6	75.9	75.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	68.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	100.0	100.0	100.0
	T16 Long-Gap Reference	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0
	T17 Sustained Reporting	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	97.9	97.7
	T18 Cumulative Evidence	0.2	80.6	0.2	0.0	4.2	0.0	0.0	0.0	0.0	89.3	55.9	50.0
Overall		20.6	29.8	9.7	4.1	11.3	2.5	0.3	11.1	0.2	97.6	92.0	90.0

(d) Time Series Language Models

Capability	Task	TimeOmni-1-4B			TS-Reasoner-7B (streaming)			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	0.0	0.0	0.0	11.0	0.0	0.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	0.0	0.0	0.0	9.8	19.6	1.9	93.4	80.5	75.2
C2: Sequential Conditions	T3 Consecutive Exceedance	0.0	0.0	0.0	7.1	0.0	0.0	100.0	100.0	100.0
	T4 Trend Recognition	0.0	0.0	0.0	6.7	50.0	3.4	93.5	96.8	90.4
	T5 Mean Shift	0.0	0.0	0.0	3.8	30.0	1.1	100.0	80.7	80.7
	T6 Event Density	0.0	0.0	0.0	4.7	0.0	0.0	99.3	100.0	99.3
C3: Event Tracking	T7 Single-Type Tracking	0.0	0.0	0.0	0.9	7.7	0.1	98.0	99.0	97.0
	T8 Multi-Type Tracking	0.0	0.0	0.0	0.8	35.0	0.3	96.9	98.6	95.6
	T9 Ordered Event Tracking	0.0	0.0	0.0	0.6	0.0	0.0	93.0	100.0	93.0
C4: Event Reference	T10 Shape Reference	0.0	0.0	0.0	1.3	0.0	0.0	100.0	100.0	100.0
	T11 Numerical Reference	0.0	0.0	0.0	0.9	0.0	0.0	98.5	89.0	87.7
C5: Online Instructions	T12 On-Demand Recall	3.5	16.7	0.6	6.3	6.7	0.4	100.0	85.7	85.7
	T13 Rule Replacement	0.0	0.0	0.0	1.4	3.4	0.0	95.0	96.0	91.3
	T14 Rule Addition	34.7	35.4	12.3	3.7	24.9	0.9	99.6	75.9	75.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	0.0	0.0	0.0	0.2	13.0	0.0	100.0	100.0	100.0
	T16 Long-Gap Reference	0.1	0.0	0.0	0.2	3.3	0.0	100.0	100.0	100.0
	T17 Sustained Reporting	0.0	0.0	0.0	0.4	16.1	0.1	99.8	97.9	97.7
	T18 Cumulative Evidence	0.0	0.0	0.0	0.1	3.8	0.0	89.3	55.9	50.0
Overall		2.1	2.9	0.7	3.3	11.9	0.5	97.6	92.0	90.0

Table 7: **Real Test: per-task main results.** Each model reports Timing F1 (F_j), Content Accuracy (C_j), and Joint Score ($S_j = 100F_jC_j$). For readability, F_j and C_j are displayed as percentages; S_j is on a 0–100 scale. Scores use unrounded values and the same evaluation results as Table 1. Overall averages each metric across all 18 tasks; its Joint Score is the mean of task scores, not the product of the aggregate metrics. TIMEOMNI-LIVE is repeated for comparison.

(a) Proprietary Models

Capability	Task	Opus 5			Sonnet 5			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	96.0	100.0	96.0	100.0	100.0	100.0	97.5	99.0	96.5
	T2 Multi-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	93.8	76.0	71.4
C2: Sequential Conditions	T3 Consecutive Exceedance	100.0	100.0	100.0	85.1	100.0	85.1	96.4	100.0	96.4
	T4 Trend Recognition	100.0	100.0	100.0	100.0	100.0	100.0	91.6	96.4	88.3
	T5 Mean Shift	100.0	80.1	80.1	100.0	80.1	80.1	85.3	82.7	70.5
	T6 Event Density	100.0	100.0	100.0	97.3	100.0	97.3	68.3	97.4	66.5
C3: Event Tracking	T7 Single-Type Tracking	96.5	100.0	96.5	100.0	100.0	100.0	83.4	74.8	62.4
	T8 Multi-Type Tracking	41.3	57.1	23.6	3.3	50.0	1.7	22.1	40.3	8.9
	T9 Ordered Event Tracking	94.1	100.0	94.1	0.0	0.0	0.0	63.8	85.4	54.4
C4: Event Reference	T10 Shape Reference	47.3	100.0	47.3	4.5	100.0	4.5	38.5	100.0	38.5
	T11 Numerical Reference	92.5	83.1	76.9	93.5	83.8	78.4	85.5	82.8	70.8
C5: Online Instructions	T12 On-Demand Recall	100.0	23.7	23.7	100.0	25.0	25.0	85.9	39.3	33.8
	T13 Rule Replacement	92.4	98.1	90.7	100.0	100.0	100.0	77.9	76.7	59.8
	T14 Rule Addition	70.2	79.4	55.8	40.0	65.6	26.2	66.1	56.9	37.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	100.0	100.0	100.0	100.0	100.0	100.0	99.2	100.0	99.2
	T16 Long-Gap Reference	0.5	100.0	0.5	0.0	0.0	0.0	0.6	100.0	0.6
	T17 Sustained Reporting	99.4	6.0	6.0	99.4	41.3	41.1	99.4	49.0	48.6
	T18 Cumulative Evidence	15.4	43.8	6.7	0.0	0.0	0.0	11.5	39.7	4.6
Overall		80.3	81.7	66.5	68.0	69.2	57.7	70.4	77.6	56.1

(b) Open-Source Models (1/2)

Capability	Task	DeepSeek-V4.1-Flash			MiniMax-M3			Qwen3.8-27B			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	86.0	100.0	86.0	97.5	99.0	96.5
	T2 Multi-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	6.0	98.9	5.9	93.8	76.0	71.4
C2: Sequential Conditions	T3 Consecutive Exceedance	100.0	100.0	100.0	85.1	100.0	85.1	23.8	92.9	22.1	96.4	100.0	96.4
	T4 Trend Recognition	98.0	100.0	98.0	98.0	100.0	98.0	98.0	98.0	96.0	91.6	96.4	88.3
	T5 Mean Shift	91.0	79.0	71.9	100.0	80.1	80.1	24.7	81.1	20.0	85.3	82.7	70.5
	T6 Event Density	86.0	100.0	86.0	86.0	100.0	86.0	95.3	100.0	95.3	68.3	97.4	66.5
C3: Event Tracking	T7 Single-Type Tracking	100.0	100.0	100.0	55.2	87.6	48.4	92.2	100.0	92.2	83.4	74.8	62.4
	T8 Multi-Type Tracking	2.8	25.9	0.7	10.6	20.4	2.2	0.0	0.0	0.0	22.1	40.3	8.9
	T9 Ordered Event Tracking	85.5	100.0	85.5	0.6	0.0	0.0	7.3	100.0	7.3	63.8	85.4	54.4
C4: Event Reference	T10 Shape Reference	8.2	100.0	8.2	0.0	0.0	0.0	0.0	0.0	0.0	38.5	100.0	38.5
	T11 Numerical Reference	66.9	1.1	0.8	52.2	0.0	0.0	52.0	99.0	51.5	85.5	82.8	70.8
C5: Online Instructions	T12 On-Demand Recall	0.0	0.0	0.0	91.9	23.4	21.5	99.3	17.7	17.5	85.9	39.3	33.8
	T13 Rule Replacement	0.0	0.0	0.0	100.0	100.0	100.0	85.1	100.0	85.1	77.9	76.7	59.8
	T14 Rule Addition	0.0	0.0	0.0	56.0	54.2	30.4	34.8	62.0	21.6	66.1	56.9	37.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	100.0	100.0	100.0	100.0	100.0	100.0	78.8	100.0	78.8	99.2	100.0	99.2
	T16 Long-Gap Reference	1.5	100.0	1.5	1.0	100.0	1.0	1.3	0.0	0.0	0.6	100.0	0.6
	T17 Sustained Reporting	99.4	58.4	58.0	82.2	0.0	0.0	99.4	30.6	30.4	99.4	49.0	48.6
	T18 Cumulative Evidence	15.4	41.7	6.4	11.5	21.9	2.5	0.0	0.0	0.0	11.5	39.7	4.6
Overall		58.6	67.0	50.9	62.8	60.4	47.5	49.1	65.6	39.4	70.4	77.6	56.1

Table 7: Real Test per-task main results (continued).

(c) Open-Source Models (2/2)

Capability	Task	Qwen3.5-9B			Qwen3.5-4B			Qwen3.5-4B (streaming)			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	83.1	82.2	68.3	0.0	0.0	0.0	2.7	50.0	1.3	97.5	99.0	96.5
	T2 Multi-Condition Triggering	31.0	98.1	30.5	16.0	98.1	15.7	3.8	0.0	0.0	93.8	76.0	71.4
C2: Sequential Conditions	T3 Consecutive Exceedance	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.4	100.0	96.4
	T4 Trend Recognition	24.0	100.0	24.0	29.9	93.0	27.8	6.7	50.0	3.3	91.6	96.4	88.3
	T5 Mean Shift	10.4	76.4	7.9	0.0	0.0	0.0	0.0	0.0	0.0	85.3	82.7	70.5
	T6 Event Density	10.7	16.7	1.8	4.4	11.0	0.5	0.0	0.0	0.0	68.3	97.4	66.5
C3: Event Tracking	T7 Single-Type Tracking	0.0	0.0	0.0	0.0	0.0	0.0	8.1	0.0	0.0	83.4	74.8	62.4
	T8 Multi-Type Tracking	0.0	0.0	0.0	0.0	0.0	0.0	3.4	33.3	1.1	22.1	40.3	8.9
	T9 Ordered Event Tracking	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	63.8	85.4	54.4
C4: Event Reference	T10 Shape Reference	0.0	0.0	0.0	3.7	76.8	2.8	4.3	0.0	0.0	38.5	100.0	38.5
	T11 Numerical Reference	59.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	85.5	82.8	70.8
C5: Online Instructions	T12 On-Demand Recall	47.4	56.6	26.8	0.0	0.0	0.0	0.0	0.0	0.0	85.9	39.3	33.8
	T13 Rule Replacement	0.0	0.0	0.0	17.9	1.7	0.3	26.9	0.0	0.0	77.9	76.7	59.8
	T14 Rule Addition	38.5	69.1	26.6	0.0	0.0	0.0	0.9	100.0	0.9	66.1	56.9	37.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	56.0	0.0	0.0	0.0	0.0	0.0	0.2	50.0	0.1	99.2	100.0	99.2
	T16 Long-Gap Reference	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.6	100.0	0.6
	T17 Sustained Reporting	14.6	70.0	10.2	0.0	0.0	0.0	7.6	0.0	0.0	99.4	49.0	48.6
	T18 Cumulative Evidence	0.2	2.0	0.0	0.1	9.0	0.0	0.1	0.0	0.0	11.5	39.7	4.6
Overall		20.9	31.7	10.9	4.0	16.1	2.6	3.9	15.7	0.4	70.4	77.6	56.1

(d) Time Series Language Models

Capability	Task	TimeOmni-1-4B			TS-Reasoner-7B (streaming)			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	0.0	0.0	0.0	8.4	5.2	0.4	97.5	99.0	96.5
	T2 Multi-Condition Triggering	0.0	0.0	0.0	7.8	15.0	1.2	93.8	76.0	71.4
C2: Sequential Conditions	T3 Consecutive Exceedance	0.0	0.0	0.0	8.1	6.7	0.5	96.4	100.0	96.4
	T4 Trend Recognition	0.0	0.0	0.0	7.5	46.7	3.5	91.6	96.4	88.3
	T5 Mean Shift	0.0	0.0	0.0	4.8	40.0	1.9	85.3	82.7	70.5
	T6 Event Density	0.0	0.0	0.0	3.6	0.0	0.0	68.3	97.4	66.5
C3: Event Tracking	T7 Single-Type Tracking	0.0	0.0	0.0	1.1	5.0	0.1	83.4	74.8	62.4
	T8 Multi-Type Tracking	0.0	0.0	0.0	3.9	10.0	0.4	22.1	40.3	8.9
	T9 Ordered Event Tracking	0.0	0.0	0.0	0.7	0.0	0.0	63.8	85.4	54.4
C4: Event Reference	T10 Shape Reference	0.0	0.0	0.0	4.3	6.4	0.3	38.5	100.0	38.5
	T11 Numerical Reference	0.0	0.0	0.0	1.5	0.0	0.0	85.5	82.8	70.8
C5: Online Instructions	T12 On-Demand Recall	3.5	0.0	0.0	1.6	0.0	0.0	85.9	39.3	33.8
	T13 Rule Replacement	0.0	0.0	0.0	1.6	0.0	0.0	77.9	76.7	59.8
	T14 Rule Addition	33.8	47.9	16.2	5.3	22.8	1.2	66.1	56.9	37.6
C6: Ultra-Long Reliability	T15 Sustained Vigilance	0.0	0.0	0.0	0.2	7.3	0.0	99.2	100.0	99.2
	T16 Long-Gap Reference	1.3	0.0	0.0	1.2	0.0	0.0	0.6	100.0	0.6
	T17 Sustained Reporting	0.0	0.0	0.0	0.6	20.7	0.1	99.4	49.0	48.6
	T18 Cumulative Evidence	0.0	0.0	0.0	0.2	0.0	0.0	11.5	39.7	4.6
Overall		2.1	2.7	0.9	3.5	10.3	0.5	70.4	77.6	56.1

D.2 Ablation Studies

Setup. All variants use the same 180 uniformly sampled ID/Real streams (11.2% of full evaluation). Encoding variants train for 2,500 steps; memory variants reuse the main checkpoint.

Encoding results. Hybrid encoding scores 85.6/59.0 on ID/Real, compared with 66.0/18.7 for time series embeddings alone and 81.6/57.8 for statistical values alone (Table 8).

Table 8: **Per-task encoding ablations** from Table 2(a). F_j and C_j are percentages; $S_j = 100F_jC_j$. Panels report ID and Real Test; shading marks TIMEOMNI-LIVE. Overall averages task scores. **Bold**: the best.

(a) ID Test

Capability	Task	Time Series Embeddings Only			Statistical Values Only			TIMEOMNI-LIVE (trained for 2,500 steps)		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	82.3	17.1	14.1	100.0	100.0	100.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	73.5	12.8	9.4	95.0	75.0	71.2	88.3	75.0	66.3
C2: Sequential Conditions	T3 Consecutive Exceedance	97.1	72.3	70.2	100.0	100.0	100.0	100.0	100.0	100.0
	T4 Trend Recognition	83.3	98.9	82.4	93.3	100.0	93.3	80.0	100.0	80.0
	T5 Mean Shift	100.0	60.0	60.0	100.0	88.6	88.6	100.0	77.2	77.2
	T6 Event Density	93.3	100.0	93.3	100.0	100.0	100.0	100.0	100.0	100.0
C3: Event Tracking	T7 Single-Type Tracking	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T8 Multi-Type Tracking	96.0	100.0	96.0	78.5	85.0	66.8	93.8	100.0	93.8
	T9 Ordered Event Tracking	100.0	100.0	100.0	86.7	100.0	86.7	100.0	100.0	100.0
C4: Event Reference	T10 Shape Reference	100.0	100.0	100.0	55.1	100.0	55.1	100.0	100.0	100.0
	T11 Numerical Reference	93.1	47.8	44.5	100.0	96.8	96.8	100.0	87.5	87.5
C5: Online Instructions	T12 On-Demand Recall	100.0	93.3	93.3	100.0	85.7	85.7	100.0	84.0	84.0
	T13 Rule Replacement	78.8	100.0	78.8	100.0	100.0	100.0	80.0	100.0	80.0
	T14 Rule Addition	89.7	56.0	50.2	98.7	65.2	64.3	96.0	69.3	66.5
C6: Ultra-Long Reliability	T15 Sustained Vigilance	71.3	1.3	0.9	100.0	100.0	100.0	100.0	100.0	100.0
	T16 Long-Gap Reference	100.0	100.0	100.0	71.6	100.0	71.6	100.0	100.0	100.0
	T17 Sustained Reporting	100.0	94.7	94.7	100.0	80.5	80.5	100.0	97.1	97.1
	T18 Cumulative Evidence	0.0	0.0	0.0	20.0	40.8	8.2	20.0	40.8	8.2
Overall		86.6	69.7	66.0	88.8	89.9	81.6	**92.1**	**90.6**	**85.6**

(b) Real Test

Capability	Task	Time Series Embeddings Only			Statistical Values Only			TIMEOMNI-LIVE (trained for 2,500 steps)		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	51.0	32.7	16.7	100.0	100.0	100.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	93.8	14.1	13.2	67.0	74.3	49.8	91.0	76.8	69.9
C2: Sequential Conditions	T3 Consecutive Exceedance	61.0	10.0	6.1	100.0	100.0	100.0	93.3	100.0	93.3
	T4 Trend Recognition	51.2	86.2	44.1	89.3	100.0	89.3	82.7	100.0	82.7
	T5 Mean Shift	33.3	50.0	16.7	80.0	80.1	64.1	73.3	80.9	59.3
	T6 Event Density	60.0	25.0	15.0	60.0	66.7	40.0	73.3	100.0	73.3
C3: Event Tracking	T7 Single-Type Tracking	8.3	6.7	0.6	53.1	50.0	26.6	73.7	70.0	51.6
	T8 Multi-Type Tracking	31.3	50.0	15.7	22.5	41.7	9.4	34.2	47.9	16.4
	T9 Ordered Event Tracking	47.1	87.5	41.2	80.5	100.0	80.5	71.3	95.3	68.0
C4: Event Reference	T10 Shape Reference	45.6	100.0	45.6	17.1	100.0	17.1	28.8	100.0	28.8
	T11 Numerical Reference	56.7	7.7	4.4	100.0	96.6	96.6	80.2	98.5	79.0
C5: Online Instructions	T12 On-Demand Recall	80.0	50.0	40.0	100.0	87.2	87.2	80.2	66.7	53.5
	T13 Rule Replacement	35.6	42.6	15.2	84.2	57.3	48.3	97.1	86.7	84.2
	T14 Rule Addition	26.7	36.0	9.6	91.0	46.9	42.6	60.4	44.3	26.8
C6: Ultra-Long Reliability	T15 Sustained Vigilance	23.3	10.0	2.3	100.0	100.0	100.0	100.0	100.0	100.0
	T16 Long-Gap Reference	0.0	0.0	0.0	4.7	100.0	4.7	5.1	100.0	5.1
	T17 Sustained Reporting	98.2	51.6	50.7	98.2	60.4	59.3	98.2	61.6	60.4
	T18 Cumulative Evidence	0.0	0.0	0.0	60.0	40.6	24.3	20.0	50.0	10.0
Overall		44.6	36.7	18.7	**72.6**	77.9	57.8	70.2	**82.1**	**59.0**

Memory results. Removing text memory lowers the ID score from 89.8 to 59.1; removing historical waveforms lowers C4 from 93.8 to 65.7. Removing uncommitted context or delaying memory reduces the ID score to 71.9 or 64.8 (Table 9).

Table 9: **Per-task memory ablations** from Table 2(b). F_j and C_j are percentages; $S_j = 100F_jC_j$. Panels report ID and Real Test; shading marks TIMEOMNI-LIVE. Overall averages task scores. **Bold**: the best.

(a) ID Test

Capability	Task	w/o Text Memory			w/o Waveform Memory			w/o Uncommitted Context			Memory Delayed by One Window			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	96.0	79.4	76.2	96.0	79.4	76.2	96.0	79.4	76.2	96.0	79.4	76.2	96.0	79.4	76.2
C2: Sequential Conditions	T3 Consecutive Exceedance	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T4 Trend Recognition	93.3	100.0	93.3	93.3	100.0	93.3	93.3	100.0	93.3	93.3	100.0	93.3	93.3	100.0	93.3
	T5 Mean Shift	100.0	79.5	79.5	100.0	80.2	80.2	100.0	79.5	79.5	100.0	79.5	79.5	100.0	79.5	79.5
	T6 Event Density	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
C3: Event Tracking	T7 Single-Type Tracking	28.0	75.0	21.0	100.0	100.0	100.0	80.5	70.0	56.3	48.0	70.8	34.0	100.0	100.0	100.0
	T8 Multi-Type Tracking	0.0	0.0	0.0	100.0	100.0	100.0	83.8	57.5	48.2	63.8	43.8	27.9	100.0	100.0	100.0
	T9 Ordered Event Tracking	70.5	100.0	70.5	86.7	100.0	86.7	59.3	100.0	59.3	66.0	100.0	66.0	86.7	100.0	86.7
C4: Event Reference	T10 Shape Reference	100.0	100.0	100.0	45.7	100.0	45.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T11 Numerical Reference	16.0	75.0	12.0	100.0	85.6	85.6	90.0	54.5	49.1	85.3	43.7	37.3	100.0	87.7	87.7
C5: Online Instructions	T12 On-Demand Recall	100.0	33.3	33.3	100.0	90.7	90.7	100.0	50.0	50.0	100.0	40.0	40.0	100.0	90.7	90.7
	T13 Rule Replacement	25.7	66.7	17.1	80.0	100.0	80.0	74.7	51.8	38.7	48.8	34.4	16.8	80.0	100.0	80.0
	T14 Rule Addition	88.2	54.5	48.1	98.8	68.3	67.5	96.7	58.9	57.0	96.2	64.9	62.4	98.8	68.3	67.5
C6: Ultra-Long Reliability	T15 Sustained Vigilance	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T16 Long-Gap Reference	100.0	100.0	100.0	21.0	100.0	21.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T17 Sustained Reporting	98.5	12.5	12.3	100.0	94.4	94.4	100.0	40.6	40.6	100.0	29.2	29.2	100.0	94.4	94.4
	T18 Cumulative Evidence	0.0	0.0	0.0	100.0	60.3	60.3	73.3	61.9	45.4	13.3	24.6	3.3	100.0	60.3	60.3
Overall		73.1	70.9	59.1	90.1	92.2	82.3	91.5	78.0	71.9	83.9	72.8	64.8	**97.5**	**92.2**	**89.8**

(b) Real Test

Capability	Task	w/o Text Memory			w/o Waveform Memory			w/o Uncommitted Context			Memory Delayed by One Window			TIMEOMNI-LIVE		
		F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j	F_j	C_j	S_j
C1: Instantaneous Conditions	T1 Single-Condition Triggering	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T2 Multi-Condition Triggering	97.1	71.9	69.9	97.1	71.9	69.9	97.1	71.9	69.9	97.1	71.9	69.9	97.1	71.9	69.9
C2: Sequential Conditions	T3 Consecutive Exceedance	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T4 Trend Recognition	76.0	100.0	76.0	76.0	100.0	76.0	76.0	100.0	76.0	76.0	100.0	76.0	76.0	100.0	76.0
	T5 Mean Shift	80.0	81.9	65.6	80.0	81.9	65.6	80.0	81.9	65.6	80.0	81.9	65.6	80.0	81.9	65.6
	T6 Event Density	66.7	100.0	66.7	66.7	100.0	66.7	66.7	100.0	66.7	66.7	100.0	66.7	66.7	100.0	66.7
C3: Event Tracking	T7 Single-Type Tracking	10.3	25.0	2.6	67.0	73.3	49.1	36.4	54.2	19.7	26.4	50.0	13.2	67.0	73.3	49.1
	T8 Multi-Type Tracking	35.8	48.6	17.4	40.9	39.6	16.2	29.9	44.4	13.3	38.0	61.1	23.2	40.9	39.6	16.2
	T9 Ordered Event Tracking	63.3	100.0	63.3	69.8	100.0	69.8	47.2	91.7	43.3	71.3	100.0	71.3	69.8	100.0	69.8
C4: Event Reference	T10 Shape Reference	23.6	100.0	23.6	28.0	100.0	28.0	36.0	100.0	36.0	28.1	100.0	28.1	36.1	100.0	36.1
	T11 Numerical Reference	58.3	57.8	33.7	80.2	85.1	68.3	69.6	86.3	60.0	68.3	62.1	42.4	80.2	85.1	68.3
C5: Online Instructions	T12 On-Demand Recall	80.4	31.7	25.5	80.4	67.2	54.0	80.4	26.7	21.4	80.4	46.7	37.5	80.4	67.2	54.0
	T13 Rule Replacement	35.0	75.5	26.4	79.0	100.0	79.0	65.3	52.5	34.3	48.9	63.0	30.8	79.0	100.0	79.0
	T14 Rule Addition	73.8	30.5	22.5	55.4	45.6	25.3	63.0	35.8	22.6	64.7	45.6	29.6	55.4	45.6	25.3
C6: Ultra-Long Reliability	T15 Sustained Vigilance	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	T16 Long-Gap Reference	2.6	100.0	2.6	4.0	100.0	4.0	1.0	100.0	1.0	1.9	100.0	1.9	1.3	100.0	1.3
	T17 Sustained Reporting	92.1	52.0	47.9	98.2	47.6	46.7	98.2	48.4	47.6	98.2	48.2	47.3	98.2	47.6	46.7
	T18 Cumulative Evidence	0.0	0.0	0.0	20.0	50.0	10.0	33.3	42.5	14.2	13.3	34.2	4.6	20.0	50.0	10.0
Overall		60.8	70.8	46.9	69.0	**81.2**	57.1	65.6	74.2	49.5	64.4	75.8	50.4	**69.3**	**81.2**	**57.4**

D.3 Backbone Analysis

Setup. Backbones share the training data, encoder, memory settings, and optimization recipe, and use the same 180 ID/Real test streams. Qwen3.5 uses its native tokenizer and 100 warm-up steps; other backbones use 200. Figure 3 shows 95% bootstrap intervals from 2,000 within-task resamples.

Results. Training loss and validation F1 stabilize by the end of training (Figure 8). Qwen3.5-4B achieves the highest overall ID and Real scores (Table 10). The base Qwen2.5-3B model's zero scores reflect tied response-control-token logits in its frozen output head.

Table 10: Backbone comparison by capability on 180 uniformly sampled ID/Real streams (11.2% of full evaluation; Joint Score S). **Bold**: best.

Backbone	C1: Instantaneous Conditions		C2: Sequential Conditions		C3: Event Tracking		C4: Event Reference		C5: Online Instructions		C6: Ultra-Long Reliability		Overall	
	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real	ID	Real
Qwen2.5-3B (base)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen2.5-3B-Instruct	83.2	84.8	86.3	61.4	74.3	33.1	86.8	43.9	65.5	36.1	72.8	37.3	77.5	47.8
Qwen2.5-Omni-3B (Thinker)	85.3	81.8	92.3	69.8	82.6	25.8	**96.2**	49.0	58.3	31.1	82.1	**39.9**	82.4	48.4
Qwen3-4B	86.0	75.4	90.0	71.9	91.6	33.1	91.1	**58.4**	**84.3**	**55.1**	79.1	38.5	86.6	54.1
Qwen3.5-4B	**88.1**	**84.9**	**93.2**	**77.1**	**95.6**	**45.0**	93.8	52.2	79.4	52.8	**88.7**	39.5	**89.8**	**57.4**

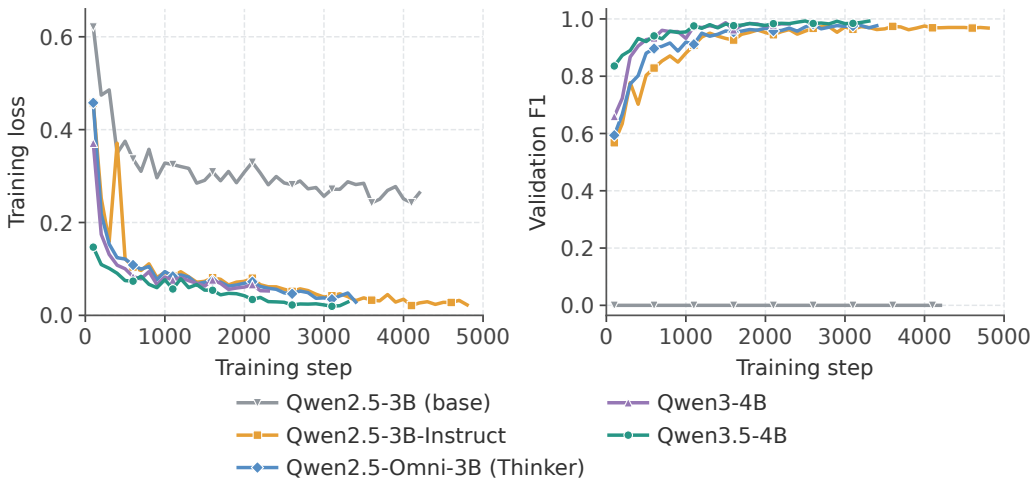


Figure 8: **Backbone convergence.** Training loss (left) and validation response-decision F1 (right) during initial training. Validation uses teacher forcing and is distinct from streaming Timing F1.

D.4 Generalization

D.4.1 Instruction Paraphrasing

Setup. Table 11 evaluates the same 180 ID streams before and after instruction paraphrasing, keeping signals, reference answers, arrival times, programs, and checkpoints fixed. DeepSeek-V4.1-Flash rewrites instructions while preserving numerical values and task requirements, with a separate model call checking equivalence. It rewrites 242 of 243 instructions, updates, and queries; opening-instruction overlap is low (**0.26** BLEU-4 and **0.36** token Jaccard), indicating substantial rewording.

Results. TimeOmni-Live changes from 89.5 to 88.5, while the five code baselines with original scores above 40 lose 17.6–32.1 points. The 95% bootstrap interval for our paired difference is

$[-3.1, +1.1]$ (2,000 within-task resamples). Together with the low lexical overlap, this supports stable generalization to differently worded user instructions.

Table 11: Instruction paraphrasing (IP; Joint Score S) by capability on paired ID streams before (ID) and after instruction paraphrasing (IP). Overall averages all 18 tasks. **Bold**: best.

Method	C1: Instantaneous Conditions		C2: Sequential Conditions		C3: Event Tracking		C4: Event Reference		C5: Online Instructions		C6: Ultra-Long Reliability		Overall	
	ID	IP	ID	IP	ID	IP	ID	IP	ID	IP	ID	IP	ID	IP
Proprietary Models														
Opus 5	95.0	**83.7**	**95.9**	89.6	84.2	44.8	69.2	31.3	50.2	55.0	72.5	49.9	78.1	60.4
Sonnet 5	**100.0**	77.4	88.1	62.3	41.8	1.0	72.2	43.6	54.7	23.8	74.5	35.0	71.3	39.2
Open-Source Models														
DeepSeek-V4.1-Flash	**100.0**	53.7	92.1	59.3	65.3	20.8	0.0	0.5	0.0	0.0	58.4	18.2	55.4	26.7
MiniMax-M3	**100.0**	50.0	85.6	58.1	17.2	2.9	25.3	5.0	45.6	20.9	68.5	17.7	58.6	26.9
Qwen3.8-27B	58.8	46.1	53.5	26.8	41.6	0.6	8.7	8.7	37.3	0.6	54.2	23.4	44.6	17.5
Qwen3.5-9B	60.6	36.9	10.8	5.0	0.0	0.0	0.0	12.1	14.0	0.0	0.0	0.0	11.5	6.6
Qwen3.5-4B	7.8	0.2	5.6	6.5	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	2.1	1.5
Time Series Language Models														
TimeOmni-1-4B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.1	1.4	0.0	0.0	0.5	0.2
Ours														
TIMEOMNI-LIVE	86.1	80.6	94.8	**93.4**	**93.8**	**94.9**	**93.3**	**88.7**	**81.8**	**88.5**	**86.6**	**82.7**	**89.5**	**88.5**

D.4.2 NUMERICAL GENERALIZATION

Setup. Table 12 evaluates numerical generalization on 850 held-out streams across all 18 tasks, with shifted numerical ranges and generation settings. Programs and checkpoints are frozen.

Results. TIMEOMNI-LIVE scores 61.5 under numerical shift, compared with 90.0 on ID Test. Performance varies across capabilities, consistent with their differing demands on learned numerical reasoning: C2 Sequential Conditions declines from 92.6 to 20.4, while C4 Event Reference retains 84.0. Despite the substantial drop, our overall score remains above DeepSeek-V4.1-Flash (54.3).

Table 12: Numerical shifts (NS; Joint Score S) by capability on ID and numerical-shift (NS) testbeds. Code-executing models and TIMEOMNI-LIVE use all 889 ID and 850 NS streams; the two direct streaming baselines use 90 streams per split due to evaluation cost. Overall averages all 18 tasks. **Bold**: best.

Method	C1: Instantaneous Conditions		C2: Sequential Conditions		C3: Event Tracking		C4: Event Reference		C5: Online Instructions		C6: Ultra-Long Reliability		Overall	
	ID	NS	ID	NS	ID	NS	ID	NS	ID	NS	ID	NS	ID	NS
Proprietary Models														
Opus 5	98.0	97.0	**96.1**	**98.0**	86.6	**89.2**	74.4	77.1	49.0	44.2	72.8	**67.0**	79.3	**78.2**
Sonnet 5	**100.0**	**100.0**	88.3	78.5	39.4	39.6	75.6	78.6	50.7	51.3	71.5	63.7	70.0	66.6
Open-Source Models														
DeepSeek-V4.1-Flash	**100.0**	**100.0**	87.8	84.4	64.9	65.5	0.0	0.6	0.0	0.0	62.7	60.5	55.4	54.3
MiniMax-M3	99.9	**100.0**	82.7	71.9	13.6	1.5	22.0	22.1	47.3	42.9	66.6	64.1	56.9	51.2
Qwen3.8-27B	54.2	51.9	56.4	44.5	34.9	31.8	17.8	20.6	38.1	38.9	50.3	46.5	43.9	40.1
Qwen3.5-9B	53.4	52.5	8.2	9.4	0.0	0.0	0.0	0.0	11.6	11.8	0.0	0.0	9.7	9.9
Qwen3.5-4B	9.3	11.6	6.7	5.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	2.5	2.4
Qwen3.5-4B (streaming)	2.1	0.9	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2
Time Series Language Models														
TimeOmni-1-4B	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.3	2.2	0.0	0.0	0.7	0.4
TS-Reasoner-7B (streaming)	1.0	0.7	1.1	0.7	0.1	0.1	0.0	0.1	0.5	0.1	0.0	0.0	0.5	0.3
Ours														
TIMEOMNI-LIVE	87.6	86.2	92.6	20.4	**95.2**	75.1	**93.8**	**84.0**	**84.2**	**70.9**	**86.9**	61.6	**90.0**	61.5

D.4.3 THIRD-PARTY DATASET EVALUATION

Setup. We evaluate domain transfer on KPI (AIOps Challenge 2018), which is absent from our original training data. Each series is split chronologically into 70% training and 30% test data; 18 series with at least three test anomaly segments yield 11,322 chunks and 638 reference segments. One-hour chunks are interpolated to 512 samples. Reports are mapped to original sample positions by their reported values, falling back to the largest deviation from the chunk median. Event-F1 PA counts a segment as detected if it contains a predicted point; predictions outside reference segments are false positives. Precision, Recall, and Event-F1 PA are computed per series and averaged equally. Baseline values come from Table 4 of Gu et al. (2025), which reports the best of ten trials for LLM methods.

Results. In Table 13, TIMEOMNI-LIVE achieves Event-F1 PA of 0.200 without adaptation and 0.539 after training on the KPI training partition. Adaptation raises recall from 0.197 to 0.568. The zero-shot result exceeds the two published prompting baselines; the adapted result exceeds AnomalyTransformer but is slightly below TFAD. The adapted checkpoint is used only for this experiment.

Table 13: KPI evaluation (↑; per-series means). Published baselines from Gu et al. (2025).

Setting	Method	Precision ↑	Recall ↑	Event-F1 PA ↑
Zero-shot	LLMAD (Liu et al., 2025)	0.033	0.940	0.053
	SigLLM-Prompter (Alnegheimish et al., 2024)	0.007	0.498	0.014
	TIMEOMNI-LIVE	0.340	0.197	0.200
KPI-trained	TFAD (Zhang et al., 2022)	0.690	0.670	0.564
	AnomalyTransformer (Xu et al., 2022)	0.482	0.395	0.282
	TIMEOMNI-LIVE	0.602	0.568	0.539

D.5 EFFICIENCY AND EDGE DEPLOYMENT

Table 14: Efficiency by group and pooled over all 54 streams. Each group contains 18 streams replayed at 1 chunk/s. Pooled values match Table 3. Parentheses give ratios to H100 TIMEOMNI-LIVE ($c = 15$, 1×), computed before rounding. Tokens are input tokens per 1k chunks.

(a) Quality and input tokens

Model	Group 1		Group 2		Group 3		Pooled (54 streams)	
	S ↑	Tokens↓	S ↑	Tokens↓	S ↑	Tokens↓	S ↑	Tokens↓
H100								
Qwen3.5-4B (streaming)	0.14	8,691,413	0.05	8,640,881	5.75	8,353,534	2.0	8,558,122 (151.4×)
TS-Reasoner-7B (streaming)	0.37	592,390	0.16	588,440	0.46	602,393	0.3	594,623 (10.5×)
TIMEOMNI-LIVE ($c = 15$)	60.32	56,450	61.85	56,312	44.24	56,837	53.8	56,541 (1×)
Jetson AGX Orin								
TIMEOMNI-LIVE ($c = 10$)	59.13	53,185	61.86	52,987	44.74	53,305	53.7	53,164 (0.94×)
TIMEOMNI-LIVE ($c = 15$)	60.12	56,454	63.16	56,314	44.24	56,744	54.3	56,511 (1.00×)
TIMEOMNI-LIVE ($c = 20$)	58.80	59,888	63.12	59,655	45.76	60,104	55.3	59,890 (1.06×)

(b) Processing speed and latency

Model	Group 1		Group 2		Group 3		Pooled (54 streams)	
	RTF↓	p95 (s)↓	RTF↓	p95 (s)↓	RTF↓	p95 (s)↓	RTF↓	p95 (s)↓
H100								
Qwen3.5-4B (streaming)	0.716	2.165	0.760	13.115	0.772	25.787	0.749 (6.4×)	16.615 (206×)
TS-Reasoner-7B (streaming)	0.275	0.475	0.277	0.527	0.274	0.510	0.276 (2.4×)	0.507 (6.3×)
TIMEOMNI-LIVE ($c = 15$)	0.104	0.078	0.103	0.078	0.143	0.439	0.117 (1×)	0.081 (1×)
Jetson AGX Orin								
TIMEOMNI-LIVE ($c = 10$)	0.394	3.105	0.323	0.276	0.404	4.929	0.375 (3.2×)	2.246 (27.8×)
TIMEOMNI-LIVE ($c = 15$)	0.336	0.288	0.323	0.262	0.412	2.629	0.358 (3.1×)	0.649 (8.0×)
TIMEOMNI-LIVE ($c = 20$)	0.357	0.308	0.320	0.252	0.399	1.994	0.360 (3.1×)	0.592 (7.3×)

Efficiency.
Setup. Table 3 evaluates 54 complete Real Test streams (three per task, 31,033 chunks) at one 512-sample chunk per second. Each model runs on a dedicated H100 with batch size one, a complete first-stream warm-up per session, and one measured repetition per stream. The cohort consists of three disjoint 18-stream groups. Joint Score follows the task-level formula in Appendix B.3, as in Table 1, recomputed across all 54 streams. Input tokens and active processing time are pooled before normalization, and p95 is computed over all chunk latencies rather than averaged across groups. **Results.** Table 14 reports each group and the pooled results from Table 3. Relative to H100 TIMEOMNI-LIVE ($c = 15, 1\times$), Qwen3.5-4B streaming uses $151.4\times$ the input tokens and has $6.4\times$ the RTF and $206\times$ the p95 latency. The corresponding ratios for TS-Reasoner-7B are $10.5\times$, $2.4\times$, and $6.3\times$.

Scheduling ablations.
Setup. Figure 5 uses a separate complete T13 stream (557 chunks), with $W = 60$, $c = 15$, and a 160-chunk warm-up. Each configuration is measured once; the main figure shows seconds 30–200. **Results.** Without elastic synchronization, consecutive chunk starts are spaced by at least one second. Without early memory updates, memory decoding waits until the window end, using the same snapshot and prompt from $W - c$. Removing both combines these restrictions. Responses and memory contents remain identical across configurations. Full-stream p95 latencies are 0.093 s (TIMEOMNI-LIVE), 0.644 s (without elastic synchronization), 0.531 s (without early updates), and 15.380 s (without both). Figure 9 shows the corresponding backlog accumulation and recovery.

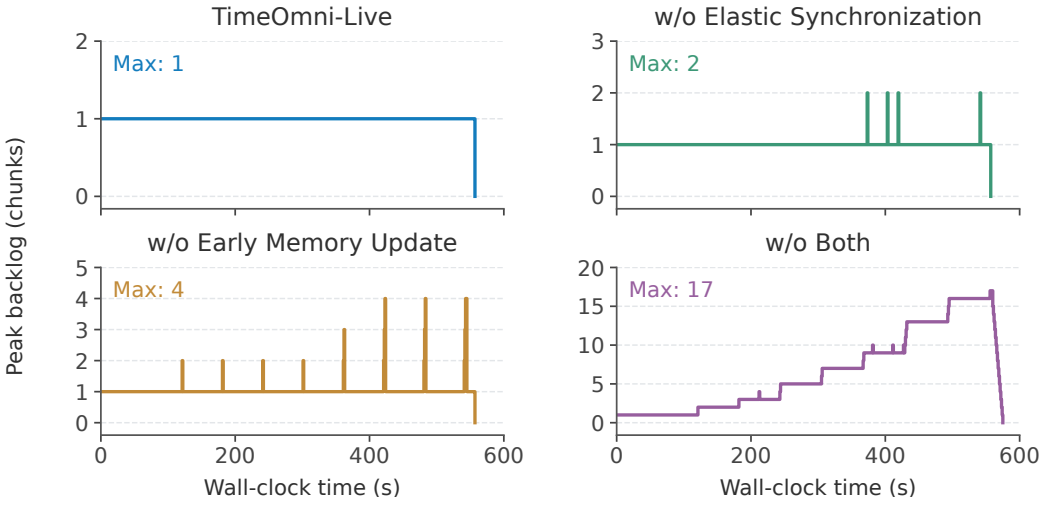


Figure 9: **Streaming backlog at 1 chunk/s.** Each panel shows the maximum number of arrived but unfinished chunks within each one-second interval, including the chunk being processed. Configurations are shown separately to avoid overlapping traces; note the different vertical scales.

Edge deployment.
Setup. Jetson AGX Orin uses the same checkpoint and 54-stream replay as H100, with BF16 backbone weights, FP32 encoder/adapter weights, and no quantization. It runs in MAXN mode with approximately 61 GiB shared memory. **Results.** Across 480 memory updates, deadline misses fall from 117 at $c = 10$ to 59 at $c = 15$ and 14 at $c = 20$ (Table 15). Pooled p95 latency falls from 2.246 s to 0.592 s, a $3.8\times$ reduction, consistent with fewer memory-update stalls. All settings achieve RTF below one. Overlap changes the retained history, so task quality is evaluated separately for each setting.

Table 15: Overlap sensitivity on Jetson.

c	Missed updates↓	p95 latency (s)↓
10	117/480	2.246 (3.8×)
15	59/480	0.649 (1.1×)
20	14/480	0.592 (1×)

E TASK EXAMPLES AND SCORING CRITERIA

The following examples are organized by the six capability groups. Each provides a task instruction, signal evidence, reference responses, and the quantities used to score their content. These are task examples from LIVETS-SUITE with reference answers, not model prediction traces. The shared timing and content scoring rules are defined in Appendix B.3.

E.1 C1: INSTANTANEOUS CONDITIONS

Table 16: Task 1 (Single-Condition Triggering): Example with ground-truth response and scoring criteria.

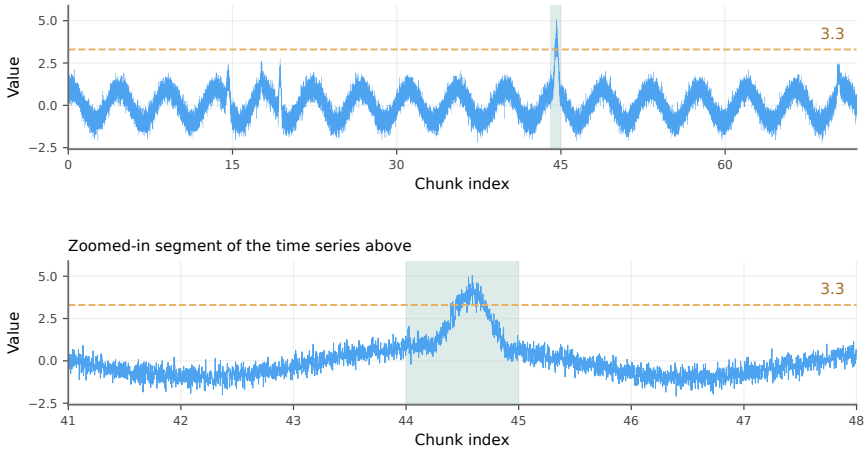
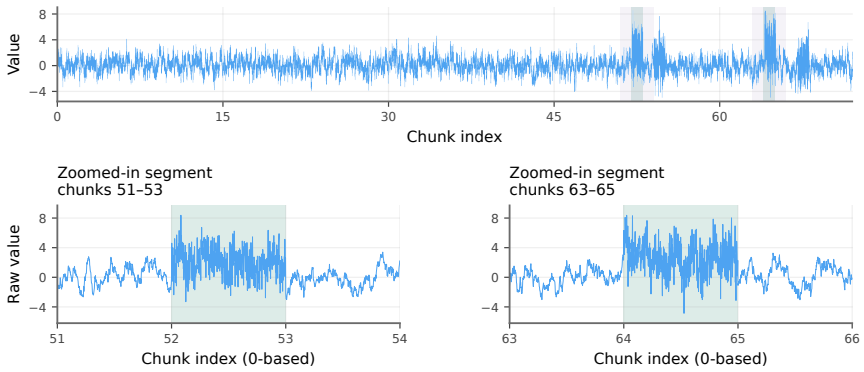
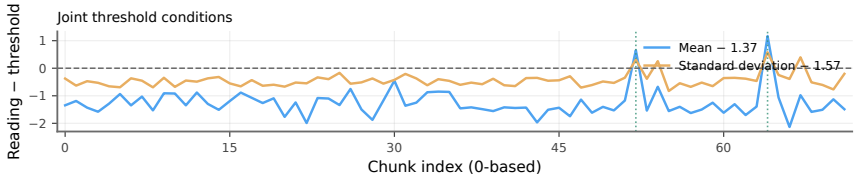
Capability C1: Instantaneous Conditions	
Task	***T1: Single-Condition Triggering***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Alert me whenever the reading rises above 3.3. One alarm per crossing: report the chunk where the reading first goes over the line, not the chunks that stay over. Each time, report exactly: "peaked at <value>", where <value> is the highest reading in that chunk. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 44. **Reported content:** peaked at 5.0391.
Scoring criteria	**Timing:** detect the threshold-crossing event. **Content:** score the reported excess above 3.3 against the ground-truth excess, 1.7391.

Table 17: Task 2 (Multi-Condition Triggering): Example with ground-truth response and scoring criteria.

Capability C1: Instantaneous Conditions	
Task	***T2: Multi-Condition Triggering***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Notify me exactly when the two conditions coincide: the average level is above 1.37, and the fluctuation is above 1.57. Report every chunk in which both conditions hold. Report both readings in one message, exactly: "average at <reading>, fluctuation at <reading>", where the first <reading> is that chunk's mean and the second is that chunk's standard deviation. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 52. **Reported content:** average at 2.02, fluctuation at 1.88. **Response time:** chunk 64. **Reported content:** average at 2.54, fluctuation at 2.15.
Evidence	Average is the mean and fluctuation is the standard deviation of the 512 samples within the current chunk (divisor 512). Each curve subtracts its threshold from that statistic; both must be positive. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond when both conditions hold simultaneously. **Content:** compare both readings as excesses relative to their respective thresholds.

E.2 C2: Sequential Conditions

Table 18: Task 3 (Consecutive Exceedance): Example with ground-truth response and scoring criteria.

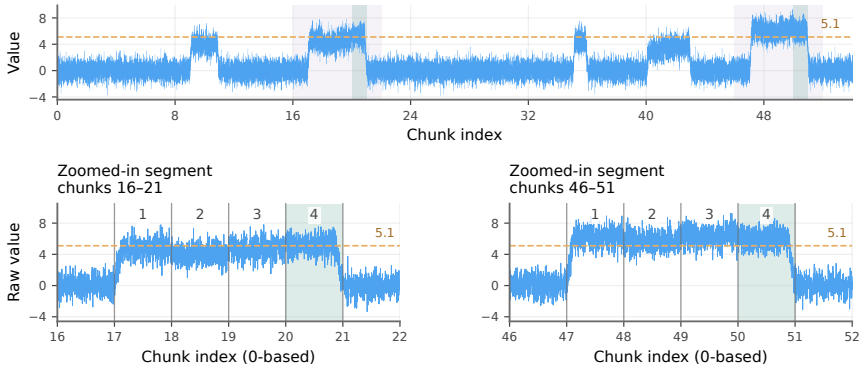
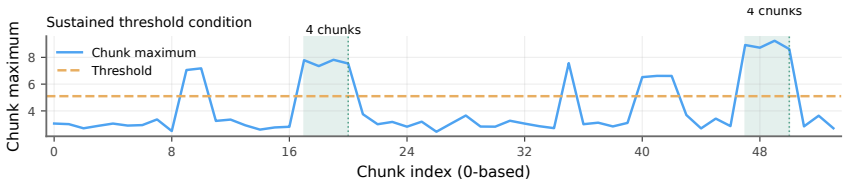
Capability C2: Sequential Conditions	
Task	***T3: Consecutive Exceedance***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). I only care if the level remains above 5.1 for 4 chunks straight – then raise an alarm. Speak at the chunk that completes the run, once per run; the run has to break and build up again before the next alarm. When you call it, report exactly: "peaked at <value>", where <value> is the highest reading across those consecutive chunks. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 20. **Reported content:** peaked at 7.8192. **Response time:** chunk 50. **Reported content:** peaked at 9.2415.
Evidence	Each close-up includes the preceding and following chunks for context. The four consecutive chunks are separated by vertical guides and numbered 1–4. Their chunk maxima exceed 5.1; the response is due at the fourth chunk. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond when the required consecutive run is complete. **Content:** compare the reported run-wide peak as an excess relative to the threshold.

Table 19: Task 4 (Trend Recognition): Example with ground-truth response and scoring criteria.

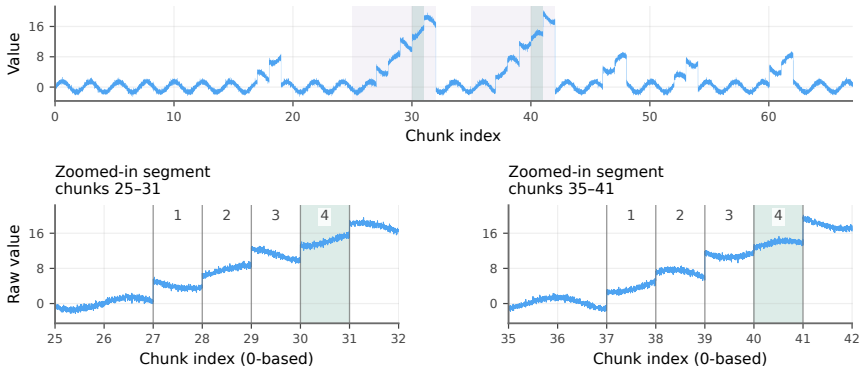
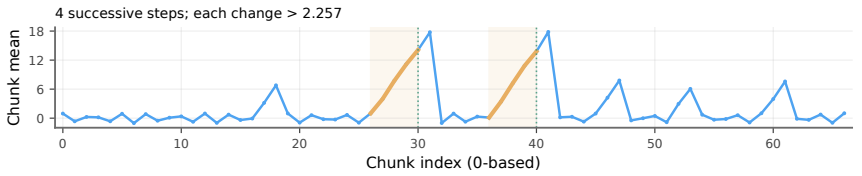
Capability C2: Sequential Conditions	
Task	***T4: Trend Recognition***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Speak up when the value has clearly increased for 4 straight chunks. ”Clearly” means each chunk mean moving more than 2.257 from the previous chunk. Speak at the chunk that completes the run, once per run; a chunk that fails to move clearly resets the count. Report exactly: ”moving <up—down>, covered <amount>”, where <amount> is the total change of the chunk mean across those consecutive chunks. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 30. **Reported content:** moving up, covered 13.0855. **Response time:** chunk 40. **Reported content:** moving up, covered 13.656.
Evidence	Chunks numbered 1–4 each have a higher mean than the preceding chunk, with every increase exceeding the stated margin. The first increase is measured from the preceding, unnumbered chunk; the fourth completes the run and triggers the response. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond at completion of the required consecutive steps. **Content:** check the direction and total movement.

Table 20: Task 5 (Mean Shift): Example with ground-truth response and scoring criteria.

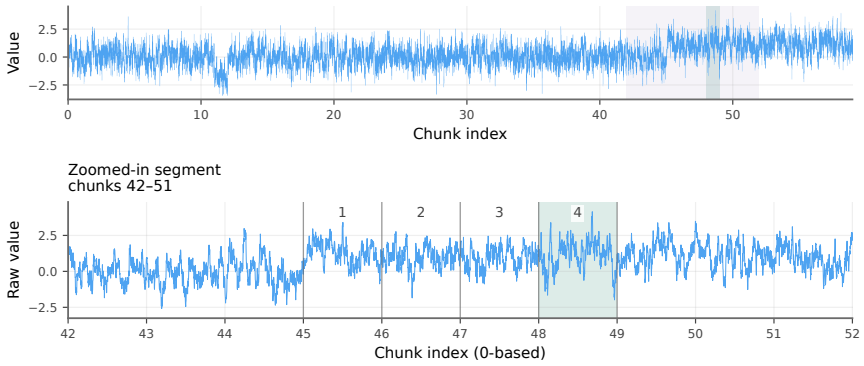
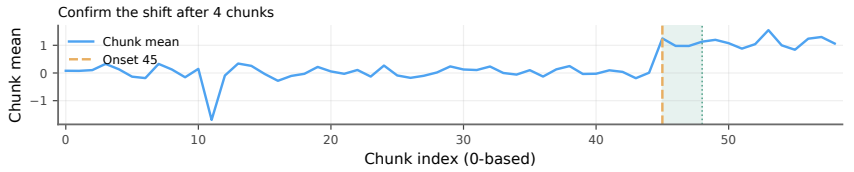
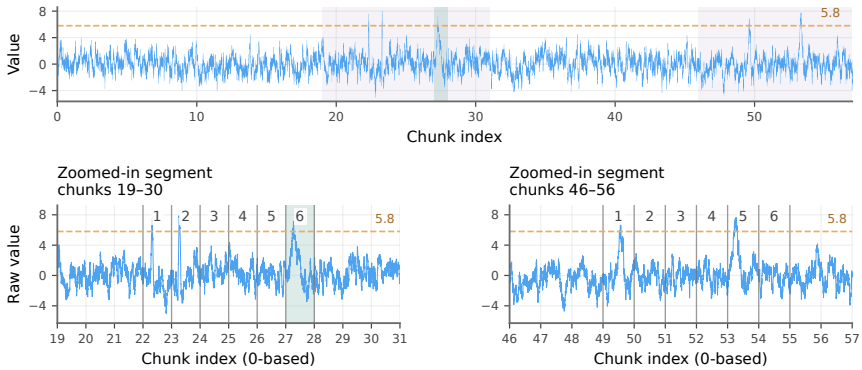
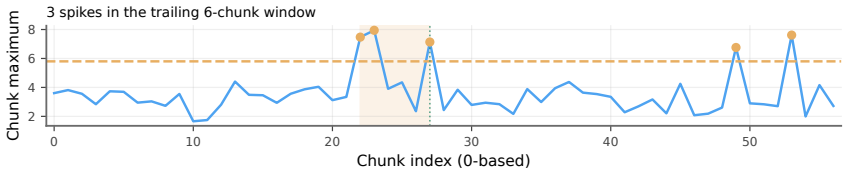
Capability C2: Sequential Conditions	
Task	***T5: Mean Shift***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Tell me the moment the signal moves clearly off its recent norm and keeps that for 4 chunks. ”Clearly” means the chunk mean moving more than 0.319 away from the median of the 8 chunks just before the shift began, and staying that far away for 4 chunks in a row. Speak at the last of those 4 chunks, once; after an alarm, stay silent until a chunk comes along that no longer counts as clearly shifted from the median of the 8 chunks before it. Report exactly: ”settled <higher—lower> by <amount>”, where <amount> is the average chunk mean after the change minus the average chunk mean before it. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 48. **Reported content:** settled higher by 1.1149.
Evidence	Chunks numbered 1–4 show the shifted level persisting for the required four chunks. The response is due at the fourth chunk, rather than at the initial shift. The plot below shows the chunk means and shift onset. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond once the level shift has persisted for the required duration. **Content:** check the shift direction and magnitude.

Table 21: Task 6 (Event Density): Example with ground-truth response and scoring criteria.

Capability C2: Sequential Conditions	
Task	***T6: Event Density***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Raise an alarm once 3 spikes have occurred within the last 6 chunks. A spike means the signal shooting past 5.8; smaller wiggles are not spikes. Speak at the chunk where the count inside the window first reaches 3; after that, stay silent until the count inside the window drops below 3 again. Report exactly: "spanned <n> chunks", where <n> is the chunk index of the last spike minus that of the first, among the 3 spikes that set off the alarm. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 27. **Reported content:** spanned 5 chunks.
Evidence	The six chunks numbered 1–6 contain three spikes, triggering the response in the final chunk. The reported span is the gap between the first and last spike (5 chunks), not the window length (6 chunks). In the other close-up, chunks 49–54 are numbered 1–6, starting at the first spike; only 2 spikes occur in this window (< 3), so no response is required. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond when the trailing window contains the required number of spikes. **Content:** exactly match the span from the first spike to the last.

E.3 C3: Event Tracking

Table 22: Task 7 (Single-Type Tracking): Example with ground-truth response and scoring criteria.

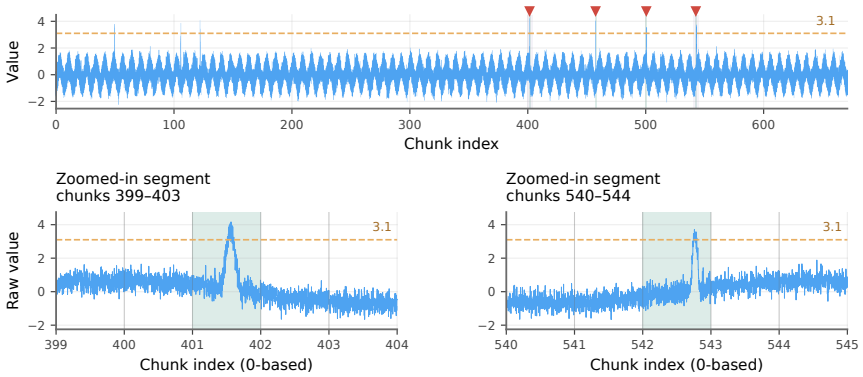
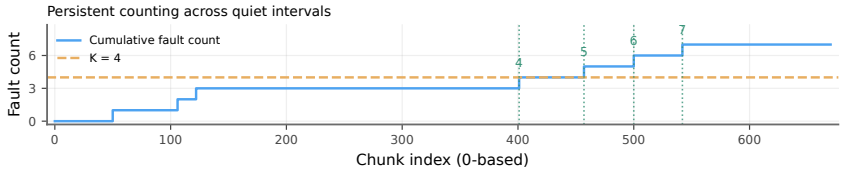
Capability C3: Event Tracking	
Task	***T7: Single-Type Tracking***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Ignore the first 3 upward spikes. Starting with the 4th, report every one. A fault means the signal shooting past 3.1; smaller wiggles are not faults. Report each as: "fault #<n> — <g> chunks since the last", where <n> counts every fault since the start (silent ones included) and <g> is this fault's chunk index minus the previous fault's. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 401. **Reported content:** fault #4 — 279 chunks since the last. **Response time:** chunk 457. **Reported content:** fault #5 — 56 chunks since the last. **Response time:** chunk 500. **Reported content:** fault #6 — 43 chunks since the last. **Response time:** chunk 542. **Reported content:** fault #7 — 42 chunks since the last.
Evidence	The count includes all faults, including the first three silent ones. The close-ups show fault #4 and fault #7; their response gaps are measured from the immediately preceding fault. *For explanation only; not model input.* 
Scoring criteria	**Timing:** report each qualifying fault starting at the specified count. **Content:** exactly match the fault index and score the gap since the previous fault.

Table 23: Task 8 (Multi-Type Tracking): Example with ground-truth response and scoring criteria.

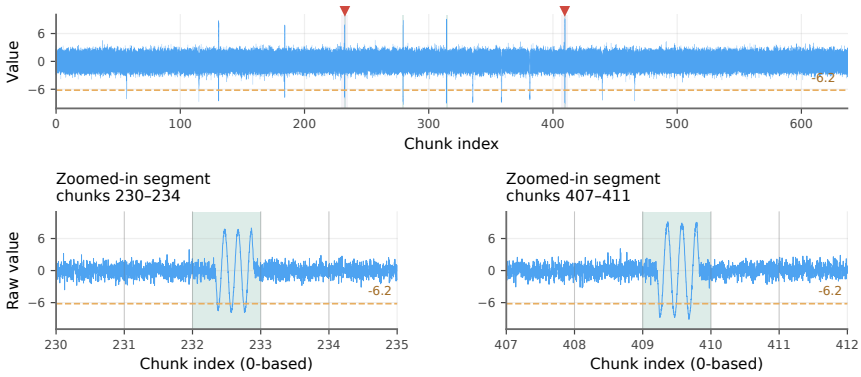
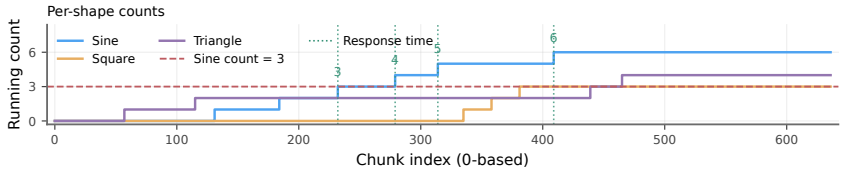
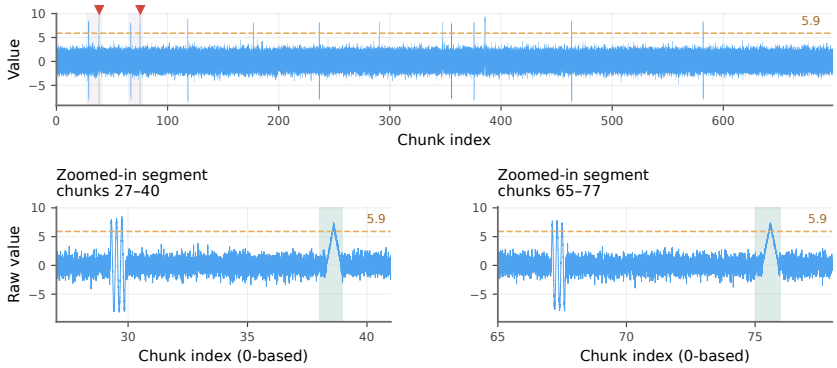
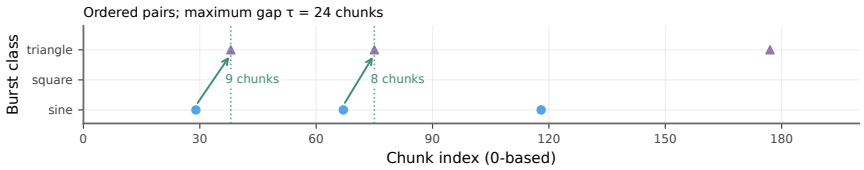
Capability C3: Event Tracking	
Task	***T8: Multi-Type Tracking***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Keep an eye on the sine-shaped bursts for me. 2 of them? Fine. The 3rd and beyond – speak up every time. A burst means a shaped pulse dropping below -6.2; the background's own wiggles never count. Report each as: "sine #<n> — square at <count>, triangle at <count>", where <n> is how many sine bursts have occurred so far including this one (so the 3rd sine is "sine #3"), and each <count> is how many bursts of that other shape have occurred so far, up to and including this chunk. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 232. **Reported content:** sine #3 — square at 0, triangle at 2. **Response time:** chunk 279. **Reported content:** sine #4 — square at 0, triangle at 2. **Response time:** chunk 314. **Reported content:** sine #5 — square at 0, triangle at 2. **Response time:** chunk 409. **Reported content:** sine #6 — square at 3, triangle at 2.
Evidence	The dashed horizontal line marks sine count = 3. Green dotted lines mark responses at chunks 232, 279, 314 and 409; the labels 3, 4, 5 and 6 give the sine count at each response: each new sine burst from the third onward triggers a report. Quiet chunks and other shapes do not trigger a report even while the sine count remains at or above 3. Counts include the current burst. *For explanation only; not model input.* 
Scoring criteria	**Timing:** report target-shape bursts starting at the specified count. **Content:** exactly match the target shape, its index, and the counts of the other shapes.

Table 24: Task 9 (Ordered Event Tracking): Example with ground-truth response and scoring criteria.

Capability C3: Event Tracking	
Task	***T9: Ordered Event Tracking***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Only this sequence matters: first a sine, then a triangle within 24 chunks. Speak at the triangle. Lone bursts and late follow-ups are fine. A burst means a shaped pulse reaching past 5.9; the background's own wiggles never count. Call it exactly as: "<g> chunks apart", where <g> is the chunk index of the triangle minus the chunk index of the sine that came before it. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 38. **Reported content:** 9 chunks apart. **Response time:** chunk 75. **Reported content:** 8 chunks apart.
Evidence	Each close-up contains a qualifying sine-triangle pair with surrounding samples. The evidence shows chunks 0–200: green arrows point from sine to triangle, with gaps of 9 and 8 chunks. Green dotted lines mark the responses at the triangle chunks (38 and 75). *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond at the second burst of a qualifying ordered pair. **Content:** score the gap between the two bursts.

E.4 C4: Event Reference

Table 25: Task 10 (Shape Reference): Example with ground-truth response and scoring criteria.

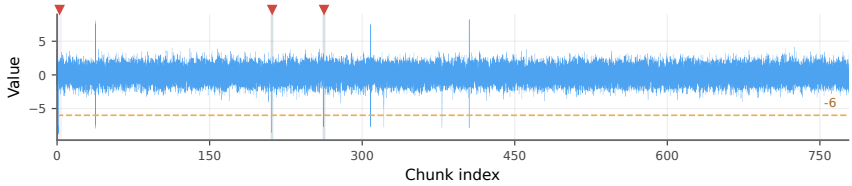
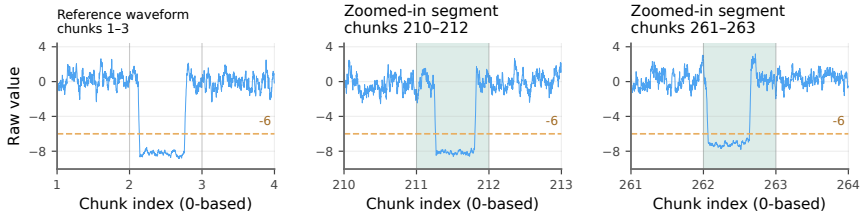
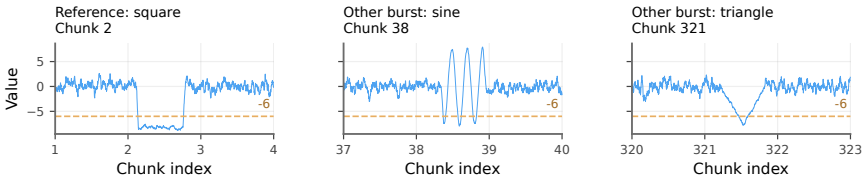
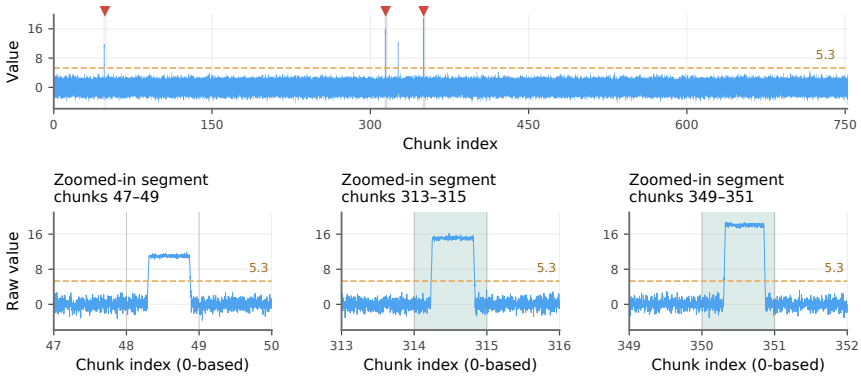
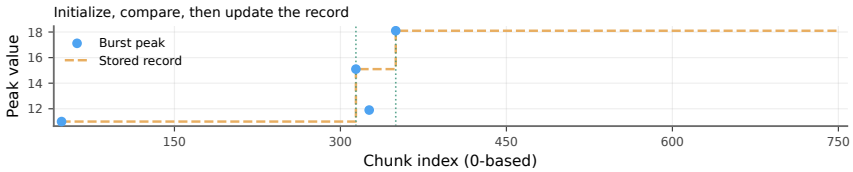
Capability C4: Event Reference	
Task	***T10: Shape Reference***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Right at the start you'll see one burst – keep its picture in mind. I want to know every time that exact look returns. The rest, stay quiet. A burst means a shaped pulse dropping below -6.0; the background's own wiggles never count. Report each return exactly as: "<g> chunks after the opener", where <g> is the chunk index of this burst minus the chunk index of the opening burst. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):   *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 211. **Reported content:** 209 chunks after the opener. **Response time:** chunk 262. **Reported content:** 260 chunks after the opener.
Evidence	The opening burst is square-shaped. Other bursts are sine- or triangle-shaped, so crossing the threshold alone does not trigger a response. Examples of both nonmatching shapes are shown below; only recurrences of the opening waveform are reported. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond when the opening waveform recurs. **Content:** score the elapsed chunks since the reference burst.

Table 26: Task 11 (Numerical Reference): Example with ground-truth response and scoring criteria.

Capability C4: Event Reference	
Task	***T11: Numerical Reference***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Track the highest peak so far. A newcomer that tops it gets reported with both values, new and old. The rest, silence. A burst only counts if it reaches past 5.3; anything tamer never enters the record book. The first burst that reaches past 5.3 sets the initial record and is not reported; from then on, report only bursts that beat the current record. Use exactly: "new record <value>, previous <value>", where the first is the peak (for a below-the-line watch: the lowest point) of this burst and the second is that of the previous record. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 314. **Reported content:** new record 15.1, previous 11. **Response time:** chunk 350. **Reported content:** new record 18.1, previous 15.1.
Evidence	The initial peak at chunk 48 sets the record to 11 without a response. The peaks at chunks 314 and 350 raise it to 15.1 and 18.1, respectively. Their increases over the previous record are 4.1 and 3.0. *For explanation only; not model input.* 
Scoring criteria	**Timing:** the first qualifying burst only initializes the record; respond when a later burst exceeds it. **Content:** score the previous peak and the absolute difference between the reported new and previous peaks, then average the two field scores.

E.5 C5: Online Instructions

Table 27: Task 12 (On-Demand Recall): Example with ground-truth response and scoring criteria.

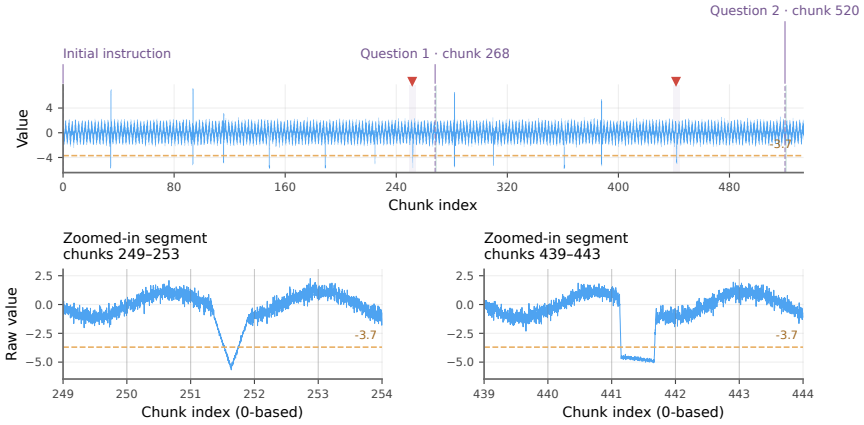
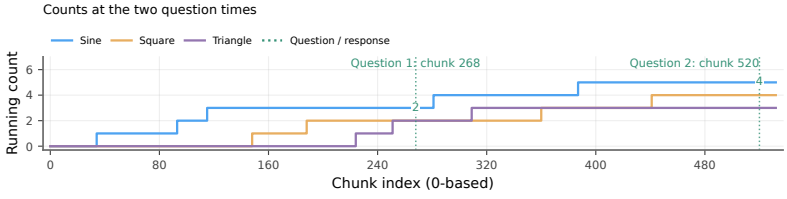
Capability C5: Online Instructions	
Task	***T12: On-Demand Recall***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Just keep watch and keep count of the bursts – I'll ask you along the way. Answer my questions with the number; otherwise stay quiet. A burst means a shaped pulse dropping below -3.7; the background's own wiggles never count. Answer with just the number, or just the shape name, and nothing else. **Question at chunk 268:** What's the count on triangle-shaped bursts up to now? Answer with just the number. **Question at chunk 520:** How many square-shaped bursts so far? Answer with just the number. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 268. **Reported content:** 2. **Response time:** chunk 520. **Reported content:** 4.
Evidence	The initial instruction requests counting but does not specify which shape will be queried. Separate shape counts must therefore be retained. Question 1 arrives at chunk 268, after the triangle burst at 251 shown in the first close-up; Question 2 arrives at 520, after the square burst at 441 in the second. Green dotted lines mark the question/response times; labels 2 and 4 give the answers. Bursts themselves require silence. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond only to the in-stream questions. **Content:** score the requested count from a number-only reply.

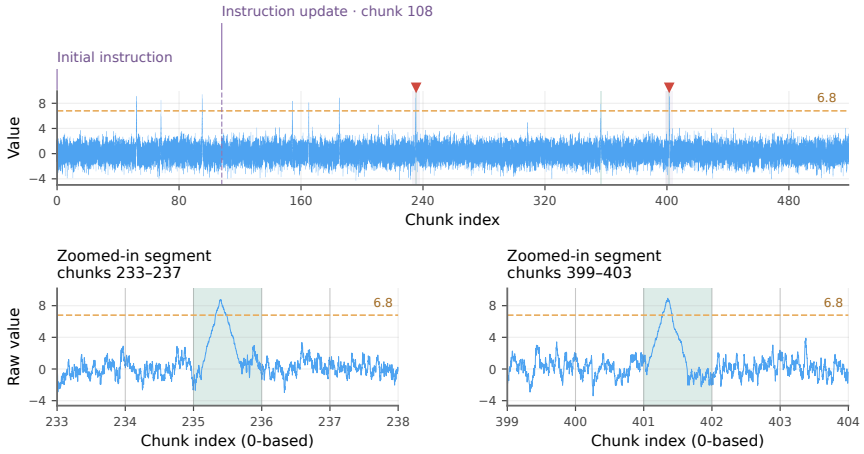
Table 28: Task 13 (Rule Replacement): Example with ground-truth response and scoring criteria.

Capability C5: Online Instructions

Task ***T13: Rule Replacement***

User The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Count the faults quietly. The 5th and every one after it gets reported. A fault means the signal shooting past 6.8; smaller wiggles are not faults. Report each as: "fault #<n> — <g> chunks since the last", where <n> counts every fault since the start (silent ones included) and <g> is this fault's chunk index minus the previous fault's.
Instruction update at chunk 108: New rule: report from the 7th one on. Forget the 5th.

Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):



Note: Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.

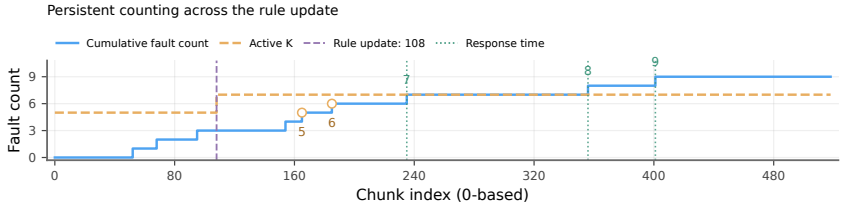
Ground Truth

Response time: chunk 235. **Reported content:** fault #7 — 50 chunks since the last.

Response time: chunk 356. **Reported content:** fault #8 — 121 chunks since the last.

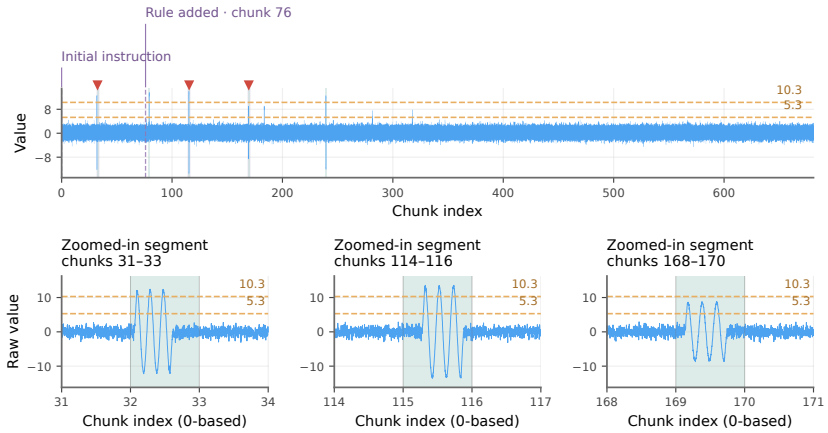
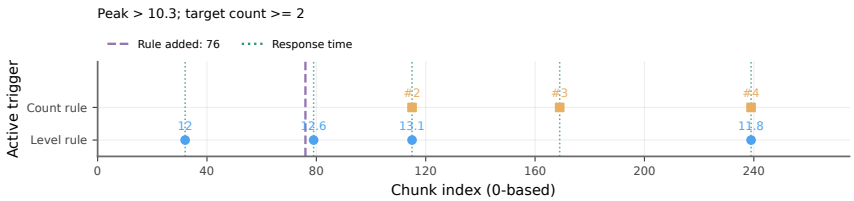
Response time: chunk 401. **Reported content:** fault #9 — 45 chunks since the last.

Evidence At chunk 108, the reporting threshold changes from 5 to 7 without resetting the count. Orange circles mark faults 5 and 6: the original rule would require reports, but the updated rule requires silence. Green lines mark the actual required reports for faults 7, 8 and 9. This sample has no required report before the update. *For explanation only; not model input.*



Scoring criteria **Timing:** apply the new count threshold immediately without resetting the count. **Content:** score the fault index and the gap from the preceding fault, with equal field weights.

Table 29: Task 14 (Rule Addition): Example with ground-truth response and scoring criteria.

Capability C5: Online Instructions	
Task	***T14: Rule Addition***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). The line to watch is 10.3. Any burst going past it gets reported with its peak. Rule: whenever a burst's highest reading rises above 10.3, say "peaked at <reading>", where <reading> is the highest reading in that chunk. A second rule will be added part-way through the stream; when both rules apply to the same burst, put both in one message, in either order. **Instruction update at chunk 76:** One more rule – sine-shaped bursts, the 2nd and beyond (count from the beginning), report them as well. Rule: whenever a sine burst is the 2nd or later, counting from the very start, say "sine #<n>", where <n> is how many sine bursts have occurred so far including this one. When both rules apply to the same burst, put both in one message, in either order. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 32. **Reported content:** peaked at 12. **Response time:** chunk 79. **Reported content:** peaked at 12.6. **Response time:** chunk 115. **Reported content:** sine #2, peaked at 13.1. **Response time:** chunk 169. **Reported content:** sine #3. **Response time:** chunk 239. **Reported content:** sine #4, peaked at 11.8.
Evidence	At chunk 76, a shape-count rule is added while the original peak rule remains active. The new count includes bursts from before the update. The purple line marks this insertion. Blue labels give peak readings; orange labels give sine-burst counts. Green dotted lines mark responses. At chunks 115 and 239 both rules fire, so both values belong in one reply. *For explanation only; not model input.* 
Scoring criteria	**Timing:** combine both facts in one reply when the active rules fire together. **Content:** score the rule set, target count and/or excess above the peak threshold, with equal weights over the applicable fields.

E.6 C6: Ultra-Long Reliability

Table 30: Task 15 (Sustained Vigilance): Example with ground-truth response and scoring criteria.

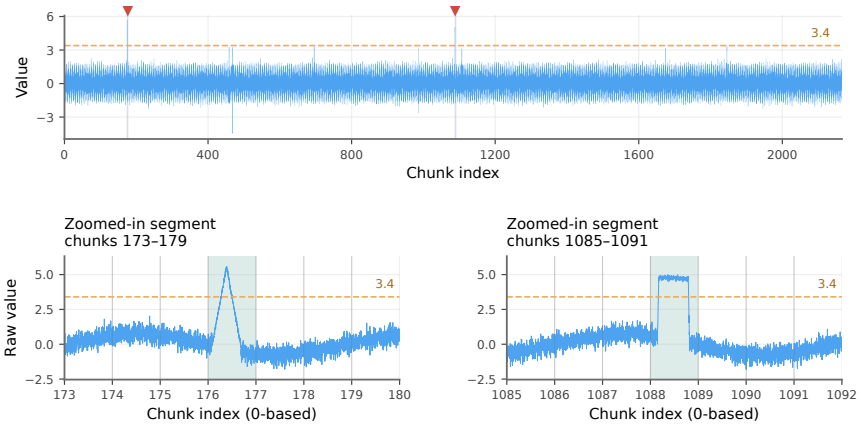
Capability C6: Ultra-Long Reliability	
Task	***T15: Sustained Vigilance***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). The line is 3.4. Report every crossing, however long the quiet stretches get. One alarm per crossing: report the chunk where the reading first goes over the line, not the chunks that stay over. Each time, report exactly: "peaked at <value>", where <value> is the highest reading in that chunk. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 176. **Reported content:** peaked at 5.5813. **Response time:** chunk 1088. **Reported content:** peaked at 4.993.
Scoring criteria	**Timing:** report only the first chunk of each crossing. **Content:** score the reported peak as an excess above the threshold.

Table 31: Task 16 (Long-Gap Reference): Example with ground-truth response and scoring criteria.

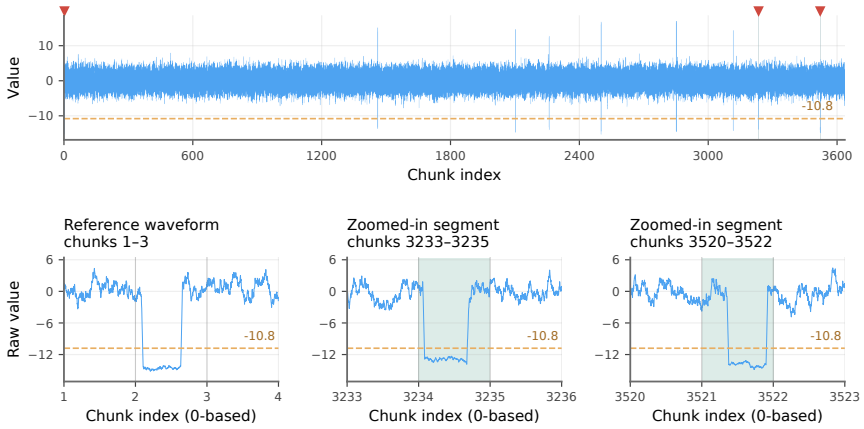
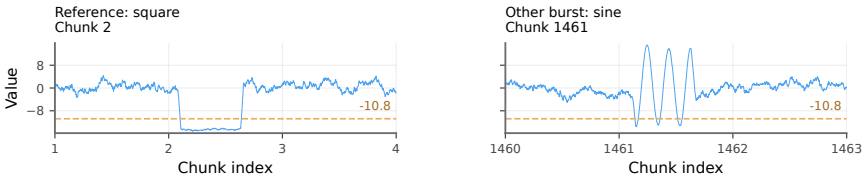
Capability C6: Ultra-Long Reliability	
Task	***T16: Long-Gap Reference***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). The very first thing on this line is a sample burst. Matching bursts after it: report each one. Non-matching: not a word. A burst means a shaped pulse dropping below -10.8; the background's own wiggles never count. Report each return exactly as: "<g> chunks after the opener", where <g> is the chunk index of this burst minus the chunk index of the opening burst. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 3234. **Reported content:** 3232 chunks after the opener. **Response time:** chunk 3521. **Reported content:** 3519 chunks after the opener.
Evidence	The opening waveform is square-shaped. All six intervening bursts are sine-shaped and do not match it; one is shown below. The square waveform returns at chunks 3234 and 3521, giving gaps of 3232 and 3519 from the reference at chunk 2. *For explanation only; not model input.* 
Scoring criteria	**Timing:** keep the opening burst silent and report each later matching burst. **Content:** score the return chunk index minus the reference chunk index.

Table 32: Task 17 (Sustained Reporting): Example with ground-truth response and scoring criteria.

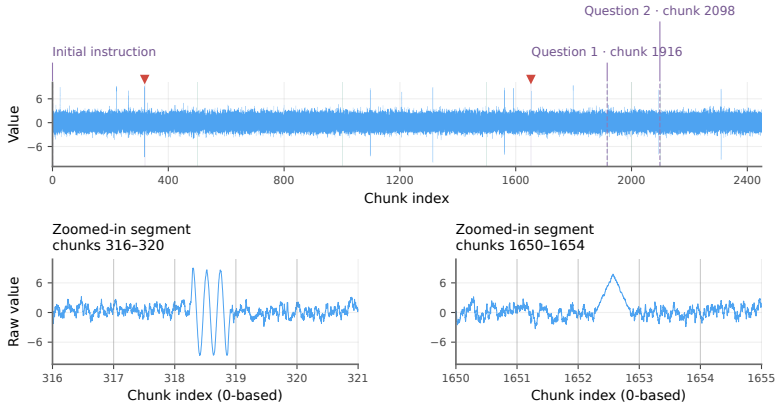
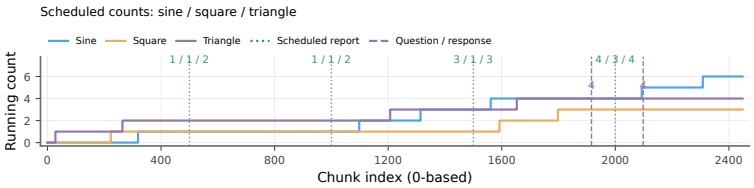
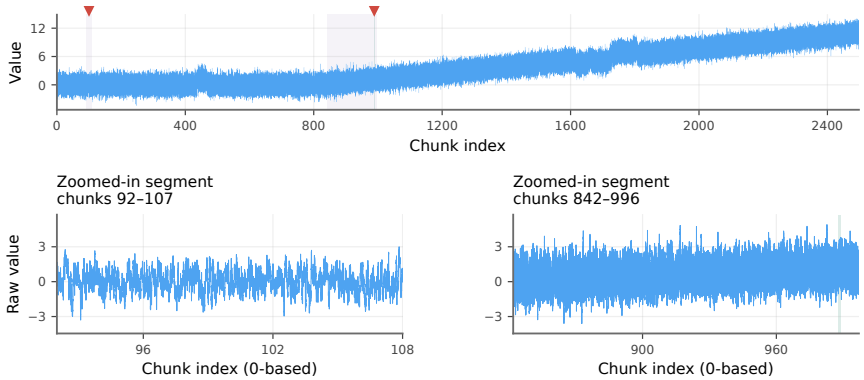
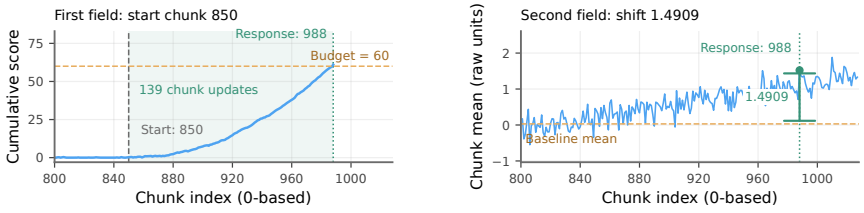
Capability C6: Ultra-Long Reliability	
Task	***T17: Sustained Reporting***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). Long shift: keep count of the bursts by shape. Every 500 chunks – at chunk 500, 1000, and so on – give me the running totals unprompted. I may also ask in between. Bursts themselves: no comment. Two kinds of replies. Scheduled report: when chunk 500, 1000, 1500, ... arrives, say exactly "sine <n>, square <n>, triangle <n>", where each <n> is how many bursts of that shape arrived before this chunk (bursts in the current chunk are not counted yet). Answer to my question: reply with just the number; the number is the count of that shape that arrived before the chunk the question comes in. **Question at chunk 1916:** How many sine-shaped bursts so far? Answer with just the number. **Question at chunk 2098:** How many triangle-shaped bursts so far? Answer with just the number. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 500. **Reported content:** sine 1, square 1, triangle 2. **Response time:** chunk 1000. **Reported content:** sine 1, square 1, triangle 2. **Response time:** chunk 1500. **Reported content:** sine 3, square 1, triangle 3. **Response time:** chunk 1916. **Reported content:** 4. **Response time:** chunk 2000. **Reported content:** sine 4, square 3, triangle 4. **Response time:** chunk 2098. **Reported content:** 4.
Evidence	Green dotted lines mark scheduled reports; their labels list sine / square / triangle counts. Purple lines mark the two questions, and the labels 4 give their answers. Counts exclude the current chunk. The raw close-ups show a sine burst and a triangle burst contributing to later totals. *For explanation only; not model input.* 
Scoring criteria	**Timing:** report at scheduled chunks and answer each intervening question. **Content:** scheduled reports require exact per-shape counts; question replies are scored as numbers, always excluding current-chunk bursts.

Table 33: Task 18 (Cumulative Evidence): Example with ground-truth response and scoring criteria.

Capability C6: Ultra-Long Reliability	
Task	***T18: Cumulative Evidence***
User	The signal arrives one chunk at a time (512 samples per chunk; chunks are numbered from 0). This machine ages slowly. If the signal's level walks away from where it began and the cumulative shift (skip anything under a quarter) tops 60, report it. A single report will do. Treat the first 100 chunks as normal: their average level is the starting level, and the spread (standard deviation) of their samples is one unit. From chunk 100 on, keep a running score: add this chunk's mean minus the starting level, measured in those units, then subtract 0.25; if the score falls below zero, reset it to zero. Keep a second score the same way for downward moves, using the starting level minus this chunk's mean. The moment either score exceeds 60, say so once and then stay silent. Report exactly: "started near chunk <n>, moved <amount>", where <n> is the chunk at which the winning score last rose from zero, and <amount> is how far this chunk's mean is from the starting level, in the original reading units, always given as a positive number. Input time series stream (shown in full; supplied sequentially in chunks of 512 samples):  *Note:* Only raw samples are provided as signal input. Shading, labels and guide lines are for illustration only and are not model inputs.
Ground Truth	**Response time:** chunk 988. **Reported content:** started near chunk 850, moved 1.4909.
Evidence	The first 100 chunks fix the baseline mean and sample standard deviation. Left (first field): normalized mean shifts, minus 0.25 per chunk, accumulate with resets at zero. The winning score first exceeds 60 at chunk 988, after 139 updates since its last departure from zero at chunk 850. The reported 850 is this start time, not the score itself. Right (second field): at chunk 988, the absolute difference between the current chunk mean and the baseline mean is 1.4909 in raw signal units. This is a current-chunk magnitude, not an accumulated sum. *For explanation only; not model input.* 
Scoring criteria	**Timing:** respond once at the first budget crossing. **Content:** score the start chunk (full credit within 30 chunks; zero at 90 or more) and the current mean's shift from baseline, then average both field scores.