TIMEINTERACT: TOWARDS REAL-TIME INTERACTIVE INTELLIGENCE FOR STREAMING TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world time series evolve continuously, with meaningful changes potentially emerging at any moment. However, existing time-series language models (TSLMs) remain inherently static. They either receive complete sequences for offline processing or alternate between streaming input and response generation, which prevents processing of new observations during interaction. We introduce a new regime, **Time-Series Interaction**: a model continuously perceives incoming time-series observations and user intent, autonomously decides when to remain silent or respond, and continues processing new observations during response generation. To realize this, we develop **TIMEINTERACT** with three key designs: a *dual-view streaming TS encoder* that captures local variations and historical dynamics, a *response control mechanism* that learns when to trigger a response, and a *decoupled streaming inference mechanism* that separates control from response generation to avoid blocking subsequent observations. We further formulate a hierarchy of interaction capabilities, progressing from Understanding to Adaptivity. Based on this hierarchy, we construct **STREAMTSI-34K**, a large-scale streaming TS interaction dataset with 34,588 episodes and 77,505 responses across synthetic and real-world time series in single- and multi-turn settings. Across all four interaction levels, TIMEINTERACT consistently outperforms existing LLMs, VLMs, and TSLMs, with gains of up to 23.92 points on challenging tasks. It also improves response triggering while achieving near-zero stream stall and up to $2.15\times$ inference speedup. The code is available at `https://anonymous.4open.science/r/Timeinteract`.

1 INTRODUCTION

Streaming time series are prevalent across a wide range of real-world domains, including healthcare monitoring, industrial systems, financial markets, and the energy sector (Huang et al., 2026; Helwig et al., 2015; Jiang et al., 2024; Dong et al., 2026; Rogers et al., 2026). In these scenarios, observations arrive incrementally as the underlying dynamics evolve (Wei et al., 2026; Li et al., 2026), while users may ask questions, set persistent monitoring objectives, or revise their goals throughout the stream (Yu et al., 2025; Kong et al., 2026). This requires continuously tracking the evolving data stream and providing timely responses as new observations arrive and user intent changes (Lau et al., 2025).

In recent years, large language models (LLMs) have substantially broadened the scope of time-series analysis through their strong capabilities in language understanding, reasoning, and zero-shot generalization (Jin et al., 2024; Wang et al., 2024a; Qiao et al., 2026). Beyond conventional tasks such as forecasting (Zhou et al., 2025) and imputation (Guan et al., 2026a), LLM-based methods have enabled richer forms of time-series analysis, including question answering (Xie et al., 2024) and temporal reasoning (Langer et al., 2025). Despite these advances, unlocking the potential of LLMs for streaming TS interaction remains challenging. ❶ **Offline TS language models** (Fig. 1(a)) operate on fixed input segments and respond only after the full sequence is available (Xie et al., 2024; Langer et al., 2025). This works for low-frequency or post-hoc analysis, but introduces substantial latency on high-frequency streams. ❷ **Interleaved streaming models** (Fig. 1(b)) reduce this latency by processing inputs incrementally and generating responses between successive input segments (Tong et al., 2025; Xie et al., 2026; Wang et al., 2026). However, response generation still blocks subsequent observations, leaving additional latency during interaction. Although both designs can accommodate streaming inputs to some extent, they still fall short of real-time interaction because input reception and response generation remain coupled. ❸ Ideally, **TS interaction models** (Fig. 1(c)) should

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.

continuously track the evolving stream, determine when a response is needed, and generate responses without interrupting subsequent inputs (Lab, 2026; Tong et al., 2026a). This requires decoupling stream processing from response generation, allowing input and output to proceed concurrently.

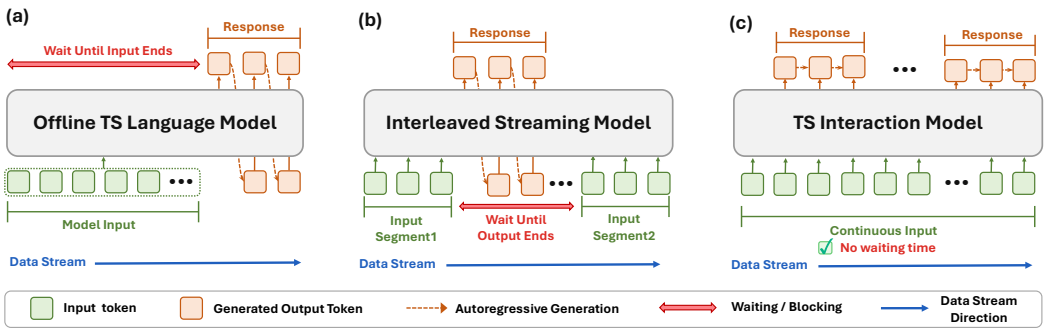


Figure 1: Overview of LLM-based paradigms for streaming time series. (a) Offline TS language models wait for the complete input before responding. (b) Interleaved streaming models alternate between input and output, causing input blocking during generation. (c) TS interaction models process incoming observations continuously while generating responses concurrently.

However, several key gaps remain in the development of TS interaction models. First, **existing models do not naturally support streaming TS interaction** (Xie et al., 2024; Langer et al., 2025). They lack key capabilities such as efficient stream encoding, response triggering, and concurrent generation. Second, **streaming TS interaction tasks lack a systematic formulation**, with different interaction forms and capability levels yet to be clearly defined. Third, **large-scale streaming TS interaction data remain scarce**. Most existing time-series QA datasets focus on isolated tasks or static question answering (Kong et al., 2025; Yu et al., 2026), with limited coverage of user queries, monitoring instructions, and multi-turn interactions over evolving streams.

To bridge these gaps, we develop **TIMEINTERACT** specifically for streaming time-series interaction. It combines efficient stream encoding, response triggering, and concurrent generation, enabling the model to continuously follow incoming observations, determine when interaction is needed, and generate responses without interrupting the ongoing stream. We then formulate a hierarchy of streaming time-series interaction tasks that captures diverse interaction forms and difficulty levels, providing a structured basis for defining and evaluating interaction capabilities over evolving streams. Building on this task hierarchy, we construct **STREAMTSI-34K**, a large-scale dataset built from both synthetic and real-world time series, where interactions are grounded in the temporal evidence observed throughout the stream. It encompasses diverse streaming interaction scenarios across domains and difficulty levels, enabling models to learn appropriate interaction behaviors over continuously evolving time-series streams. Our main contributions are summarized as follows:

1. **A large-scale streaming TS interaction dataset.** We construct **STREAMTSI-34K**, a large-scale dataset for streaming TS interaction. It combines controllable synthetic scenarios with diverse real-world time series and includes both **single-turn and multi-turn interactions**, capturing localized interaction behaviors as well as interaction continuity and cross-turn dependencies. These interactions cover **multiple interaction tasks** across the proposed capability hierarchy.
2. **A specialized model for streaming TS interaction.** We propose **TIMEINTERACT** with three key designs: a **dual-view streaming TS encoder** for efficient stream encoding, a **response triggering mechanism** for deciding when to respond, and a **parallel control–response KV cache** that decouples stream processing from response generation. These designs decouple continuous input processing from response generation to support real-time TS interaction.
3. **A systematic formulation of TS interaction tasks.** To capture the capability progression of TS interaction, we formulate a four-level hierarchy of TS interaction tasks with increasing interaction difficulty: **L1: Understanding** $\rightarrow$ **L2: Persistence** $\rightarrow$ **L3: Initiative** $\rightarrow$ **L4: Adaptivity**. This hierarchy offers a unified way to organize and compare TS interaction capabilities at different levels of difficulty and enables models to be evaluated at different capability levels.

2 Related Work

LLM-Based Time-Series Analysis. In recent years, LLMs have been increasingly adopted for time-series analysis (Zhang et al., 2024; Liu et al., 2025; Guan et al., 2026b). Initial efforts focused on transferring the knowledge and generalization ability of pretrained LLMs to conventional tasks such as forecasting and anomaly detection (Jia et al., 2024; Jin et al., 2024; Sun et al., 2024b). The scope has since expanded beyond numerical tasks toward language-based time-series understanding (Cai et al., 2024). Time-series language models (TSLMs) incorporate temporal signals through either textual inputs (Gruver et al., 2023; Zhou et al., 2023) or temporal representations interleaved with language (Xie et al., 2024; Langer et al., 2025; Parker et al., 2025), enabling description, question answering, and temporal reasoning. Recent work has moved toward more complex reasoning settings, including diverse reasoning capabilities and multi-turn analytical workflows (Guan et al., 2026a; Yu et al., 2026; Kong et al., 2026). However, these methods largely operate on time-series context that is available in advance, rather than observations that continuously arrive and evolve during interaction.

Streaming Multimodal Interaction. Real-time capability has become increasingly important for models processing continuously arriving multimodal signals (Lu et al., 2026; Chen & Yu, 2025; Hu et al., 2026). In audio, streaming ASR and spoken-dialogue systems reduce latency through incremental processing (Gao et al., 2022; Xie & Wu, 2024; Défossez et al., 2024). Recent work further explores richer interaction behaviors. Audio Interaction Model (Xie et al., 2026) unifies perception, response decisions, and generation, while JoyAI-VL-Interaction (Yao et al., 2026) extends such interaction to live video by learning when to respond, remain silent, or delegate. These developments shift the focus from simply generating responses to deciding *whether and when* to interact (Wang et al., 2024b; Lab, 2026; Hu et al., 2026; Tong et al., 2026c). However, existing interaction models remain centered on audio and visual modalities. To the best of our knowledge, we introduce the first interaction model specifically designed for streaming time series.

3 TimeInteract

Problem Definition. Offline time-series language models receive a complete multivariate time series $\mathbf{X} \in \mathbb{R}^{L\times M}$ and generate a response r to a given query or instruction $\mathcal{E}$. This can be formulated as $r = f_\theta(\mathbf{X}, \mathcal{E})$, where L and M denote the sequence length and number of variables, respectively. TimeInteract instead operates over continuously evolving time-series stream chunks $\mathbf{C}_{\leq t}$ with optional queries or instructions $\mathcal{E}_{\leq t}$, determining whether a response is required whenever new information becomes available. The overall framework is illustrated in Figure 2.

$$(d_t, r_t) = f_\theta\left(\mathbf{C}_{\leq t}, \mathcal{E}_{\leq t}, d_{<t}, r_{<t}\right), \tag{1}$$

where each incoming chunk is represented as $\mathbf{C}_t \in \mathbb{R}^{L_t\times M}$ and $\mathbf{C}_{\leq t}$ denotes the observations up to step t. $\mathcal{E}_{\leq t}$ denotes the user queries or instructions received by that time, while $d_{<t}$ and $r_{<t}$ record previous decisions and responses. The current decision $d_t \in \{\texttt{silent}, \texttt{respond}\}$ determines the model's behavior: $d_t = \texttt{silent}$ continues stream processing without textual output, whereas $d_t = \texttt{respond}$ initiates response generation while new observations continue to arrive.

3.1 Dual-View Streaming TS Encoder

Dual-Reference Normalization. Offline time-series language models typically normalize a complete input sequence $\mathbf{X} \in \mathbb{R}^{L\times M}$ using statistics derived from the input itself (Langer et al., 2025). In streaming settings, however, the incoming chunk $\mathbf{C}_t \in \mathbb{R}^{L_t\times M}$ contains both local temporal dynamics and changes relative to the accumulated historical stream (Lau et al., 2025; Li et al., 2026). A current view alone is therefore insufficient; thus we construct two complementary views:

$$\hat{\mathbf{C}}_t^F = \frac{\mathbf{C}_t - \boldsymbol{\mu}_t^c}{\boldsymbol{\sigma}_t^c + \epsilon}, \qquad \hat{\mathbf{C}}_t^H = \frac{\mathbf{C}_t - \boldsymbol{\mu}_{t-1}^h}{\boldsymbol{\sigma}_{t-1}^h + \epsilon}, \tag{2}$$

where $\boldsymbol{\mu}_t^c, \boldsymbol{\sigma}_t^c \in \mathbb{R}^M$ are computed from the current chunk, while $\boldsymbol{\mu}_{t-1}^h, \boldsymbol{\sigma}_{t-1}^h \in \mathbb{R}^M$ summarize the historical stream up to step t and ϵ is a small constant. (1) The *Fast View* $\hat{\mathbf{C}}_t^F$ emphasizes local variations within the current chunk. (2) The *Historical View* $\hat{\mathbf{C}}_t^H$ captures how observations differ from the accumulated historical stream. Historical statistics are updated only after processing $\mathbf{C}_t$, preventing the current chunk from affecting its own historical view (See Appendix A for details).

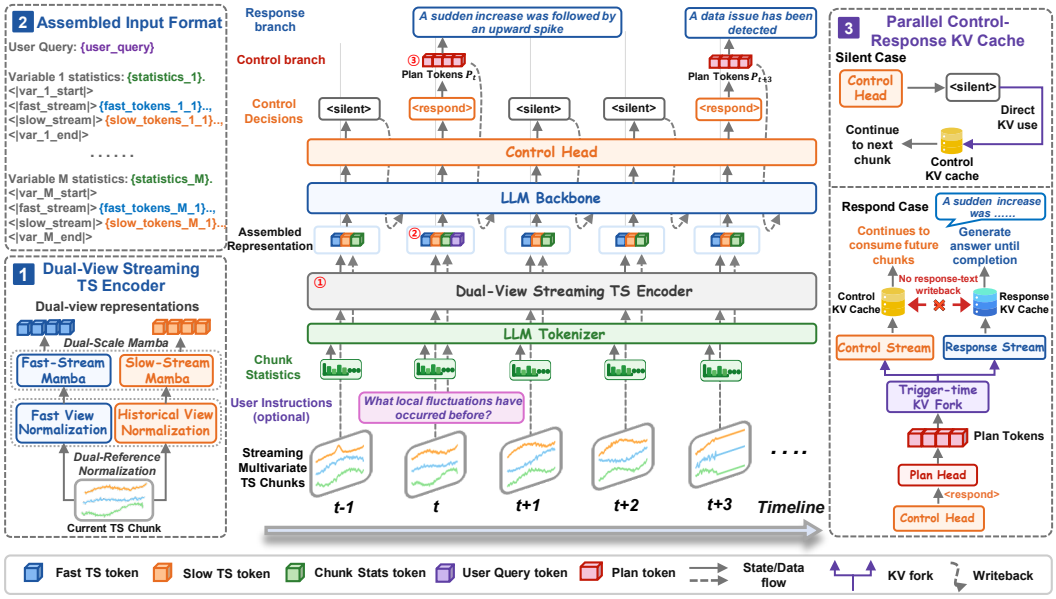


Figure 2: Overview of TIMEINTERACT. (1) The dual-view streaming TS encoder captures local variations and historical dynamics. (2) Input assembly combines temporal tokens, chunk statistics, and user instructions. (3) KV forking supports concurrent stream processing and response generation.

Fast–Slow Stream Encoding. The normalized views $\hat{\mathbf{C}}_t^F$ and $\hat{\mathbf{C}}_t^H$ are first transformed into patch-level representations. Specifically, each variable is segmented along the temporal dimension into J ordered patches and projected into an H-dimensional hidden space, yielding $\mathbf{E}_t^F, \mathbf{E}_t^H \in \mathbb{R}^{M\times J\times H}$. These representations are then processed by independent Fast and Slow Mamba encoders:

$$(\mathbf{Z}_t^F, \mathbf{S}_t^F) = \mathcal{M}_F(\mathbf{E}_t^F, \mathbf{S}_{t-1}^F; \alpha_F), \qquad (\mathbf{Z}_t^S, \mathbf{S}_t^S) = \mathcal{M}_S(\mathbf{E}_t^H, \mathbf{S}_{t-1}^S; \alpha_S). \tag{3}$$

where $\mathcal{M}_F$ and $\mathcal{M}_S$ are independent Mamba encoders (Gu & Dao, 2023), and $\mathbf{S}_t^F$ and $\mathbf{S}_t^S$ are persistent SSM states carried across chunks. Their update rates satisfy $\alpha_F > \alpha_S$, allowing the Slow Stream to retain history longer while the Fast Stream responds more rapidly to recent changes. This persistent state enables incremental encoding without revisiting the full history. The resulting trajectories $\mathbf{Z}_t^F, \mathbf{Z}_t^S \in \mathbb{R}^{M\times J\times H}$ are further projected into the LLM hidden space:

$$\mathbf{T}_t^F = \mathrm{RMSNorm}\big(\Pi_F(\mathbf{Z}_t^F)\big), \qquad \mathbf{T}_t^S = \mathrm{RMSNorm}\big(\Pi_S(\mathbf{Z}_t^S)\big), \tag{4}$$

where $\Pi_F, \Pi_S : \mathbb{R}^H \to \mathbb{R}^D$ are learned projections and $\mathbf{T}_t^F, \mathbf{T}_t^S \in \mathbb{R}^{M\times J\times D}$ denote the ordered Fast and Slow soft tokens. RMSNorm stabilizes temporal token scales via root mean square normalization and learned scaling. Each valid patch contributes one token per stream, and the tokens are inserted into the LLM context, where cross-variable dependencies are modeled through self-attention.

3.2 STREAMING INTERACTION TRAINING

Response Control and Latent Plans. At each streaming step, TIMEINTERACT forms a causal context $\mathcal{H}_t^P$ from the Fast and Slow tokens $\mathbf{T}_{\le t}^F, \mathbf{T}_{\le t}^S$, user queries or instructions $\mathcal{E}_{\le t}$, previous decisions $d_{<t}$, and Plans $\mathbf{P}_{<t}$ representing previous responses $r_{<t}$. A control head G_{ctrl} predicts $d_t \in \{\texttt{silent}, \texttt{respond}\}$. Silent decisions continue stream processing without text, while response decisions activate a Plan head G_{plan} to produce continuous Plan tokens $\mathbf{P}_t$ before decoding r_t:

$$p_\theta(d_t \mid \mathcal{H}_t^P) = \mathrm{Softmax}\big(G_{\mathrm{ctrl}}(\mathbf{h}_t^d)\big), \tag{5}$$

$$\mathbf{P}_t = G_{\mathrm{plan}}(\mathbf{h}_t^p) \in \mathbb{R}^{K\times D}, \qquad \text{if } d_t = \texttt{respond}, \tag{6}$$

where $\mathbf{h}_t^d$ and $\mathbf{h}_t^p$ are the LLM hidden states used by the control and Plan heads, respectively, K is the number of Plan tokens, and D is the LLM hidden dimension. We supervise response decisions with weighted cross-entropy $\mathcal{L}_{\mathrm{control}}$ to account for sparse response events. The Plan tokens $\mathbf{P}_t$ provide a compact representation of the intended response and remain in the control history to inform future decisions, allowing stream processing to continue without waiting for the complete response r_t.

Full-to-Plan Distillation. To maintain a compact response history during inference, Plan tokens should preserve the information needed for subsequent interactions. We train the same language model under two aligned views: the *Full View* $\mathcal{H}_t^F$ augments $\mathcal{H}_t^P$ with previous ground-truth responses $r^*_{<t}$, while the *Plan View* retains only $\mathcal{H}_t^P$, excluding previous response text. Both views condition on the current Plan tokens $\mathbf{P}_t$ and the same target response prefix $r^*_{t,<j}$. We supervise Full-View response generation and distill its token distributions into the Plan View at every target response position:

$$\mathcal{L}_{\text{full}} = -\frac{1}{N_r} \sum_{t\in\mathcal{R}} \sum_{j=1}^{|r_t^*|} \log p_\theta\big(r^*_{t,j} \mid \mathcal{H}_t^F, \mathbf{P}_t, r^*_{t,<j}\big)\,, \tag{7}$$

$$\mathcal{L}_{\text{distill}} = \frac{\tau^2}{N_r} \sum_{t\in\mathcal{R}} \sum_{j=1}^{|r_t^*|} D_{\text{KL}}\big[\text{sg}(q^F_{t,j}) \,\|\, q^P_{t,j}\big]\,, \tag{8}$$

where $\mathcal{R}$ indexes response steps and $N_r = \sum_{t\in\mathcal{R}} |r_t^*|$ is the total number of supervised response tokens. For view $v \in \{F, P\}$, $q^v_{t,j} = \text{Softmax}(\mathbf{z}^v_{t,j}/\tau)$ denotes the vocabulary distribution obtained from logits $\mathbf{z}^v_{t,j}$ at temperature τ. The operator $\text{sg}(\cdot)$ stops gradients through the Full-View targets in distillation, while $\mathcal{L}_{\text{full}}$ trains the shared model and Plan head. The distillation loss encourages the Plan tokens to preserve information required by future responses without retaining past replies.

Three-Stage Training Pipeline. We adopt a three-stage training pipeline to learn the target sequence format, acquire the interaction mechanism, and scale training to diverse streaming settings. Training progressively introduces more complex interaction scenarios, in which previous Plan tokens inform subsequent response decisions and generation. At each stage, we optimize a weighted combination of the Response Control, Full-View generation, and Full-to-Plan distillation losses:

$$\mathcal{L}^{(s)} = \lambda_c^{(s)}\mathcal{L}_{\text{control}} + \lambda_f^{(s)}\mathcal{L}_{\text{full}} + \lambda_d^{(s)}\mathcal{L}_{\text{distill}}, \qquad s \in \{1, 2, 3\}, \tag{9}$$

where s indexes the training stage and $\lambda_c^{(s)}, \lambda_f^{(s)}, \lambda_d^{(s)}$ are nonnegative loss weights. (1) *Stage I: Temporal-Language Alignment.* We use simple interaction tasks in single-turn scenarios to learn the interaction format, training the streaming TS encoder and control and Plan heads with the LLM backbone frozen and distillation disabled ($\lambda_d^{(1)} = 0$). (2) *Stage II: Interaction Mechanism Learning.* We train on all interaction tasks within single-turn scenarios, unfreeze the language model, and jointly optimize all three losses to learn response timing and how previous Plan tokens inform subsequent decisions. (3) *Stage III: Large-Scale Streaming Interaction Training.* We extend joint training to the full data covering all interaction tasks in complex scenarios, including multi-turn sessions with mixed interaction forms and dependencies on earlier instructions, responses, and user feedback.

3.3 Decoupled Streaming Inference via KV Forking

Autoregressive response generation may span multiple time-series chunks $\{\mathbf{C}_t\}$. In an interleaved setting, the model must finish generating the current response before processing subsequent input, causing response generation to block the incoming stream. To remove this dependency, TimeInteract separates inference into a persistent *Control Stream* and temporary *Response Streams* with distinct KV caches (Tong et al., 2026a; Zhang et al., 2026). The control cache $\mathbf{KV}^{\text{ctrl}}$ follows the Plan View $\mathcal{H}_t^P$: it accumulates temporal tokens, user instructions, decisions, and Plan tokens, while the TS encoder carries its states across input chunks.

At streaming step t, the Encode operation encodes the incoming chunk $\mathbf{C}_t$ into the temporal block $\mathbf{B}_t$, and Control predicts and commits the binary response d_t. User input is optional:

Algorithm 1 Decoupled Streaming Inference

Input: TS stream $\{\mathbf{C}_t\}$, optional user inputs $\{\mathcal{E}_t\}$
Initialize $\mathbf{KV}^{\text{ctrl}}$ and encoder states
$\mathcal{Q} \leftarrow \emptyset$; no active response
while stream open or $\mathcal{Q} \neq \emptyset$ **do**
  if a new chunk $\mathbf{C}_t$ is available **then**
    $\mathbf{B}_t \leftarrow \text{Encode}(\mathbf{C}_t)$
    $(d_t, \mathbf{KV}^{\text{ctrl}}) \leftarrow \text{Control}(\mathbf{B}_t, \mathcal{E}_t, \mathbf{KV}^{\text{ctrl}})$
    if $d_t =$ `respond` **then**
      $(\mathbf{P}_t, \mathbf{KV}^*_t, \mathbf{h}^r_t) \leftarrow \text{Plan}(\mathbf{KV}^{\text{ctrl}})$
      $(\mathbf{KV}^{\text{ctrl}}, \mathbf{KV}^{\text{resp}}_t) \leftarrow \text{Fork}(\mathbf{KV}^*_t)$
      $\mathbf{KV}^{\text{ctrl}} \leftarrow \text{FinishStep}(\mathbf{KV}^{\text{ctrl}})$
      $\text{LaunchResponse}(\mathcal{Q}, \mathbf{KV}^{\text{resp}}_t, \mathbf{h}^r_t)$
    else `// silent`
      $\mathbf{KV}^{\text{ctrl}} \leftarrow \text{FinishStep}(\mathbf{KV}^{\text{ctrl}})$
    end if
  end if
  $\text{CollectFinished}(\mathcal{Q})$
end while

$\mathcal{E}_t = \emptyset$ when no new query or instruction arrives. A `silent` decision closes the control step without creating a response request. For `respond`, PLAN appends $\mathbf{P}_t$ and returns the cache $\mathbf{KV}_t^*$ and hidden state $\mathbf{h}_t^r$ used to predict the first response token. The cache is then forked into the continuing control branch $\mathbf{KV}^{\text{ctrl}}$ and a response branch $\mathbf{KV}_t^{\text{resp}}$. FINISHSTEP closes the current control step without modifying the response branch. Subsequent cache updates remain isolated: new observations extend the control history, while generated text extends only the response branch. Each response stream is launched asynchronously by LAUNCHRESPONSE and tracked in $\mathcal{Q}$. Once a response completes, its cache is released, while the corresponding Plan tokens remain in the control history. Algorithm 1 summarizes the overall inference and response scheduling procedure.

4 STREAMTSI-34K

Overview. STREAMTSI-34K is a large-scale dataset designed for streaming time-series interaction. The dataset comprises 34,588 interaction episodes and 77,505 annotated responses (1–10 per episode), drawing on both controllable synthetic signals and real-world time series from six domains. Across these diverse streams, we annotate four interaction tasks: *Instant Query Answering (IQA)*, *Persistent Instruction Following (PIF)*, *Proactive Temporal Warning (PTW)*, and *User-Guided Adaptation (UGA)* (Figure 3). These tasks are organized into single-turn episodes (56.9%) and composed multi-turn interactions (43.1%). Single-turn tasks ground response timing and content in temporal evidence and user intent, while multi-turn interactions capture cross-turn dependencies and interaction continuity.

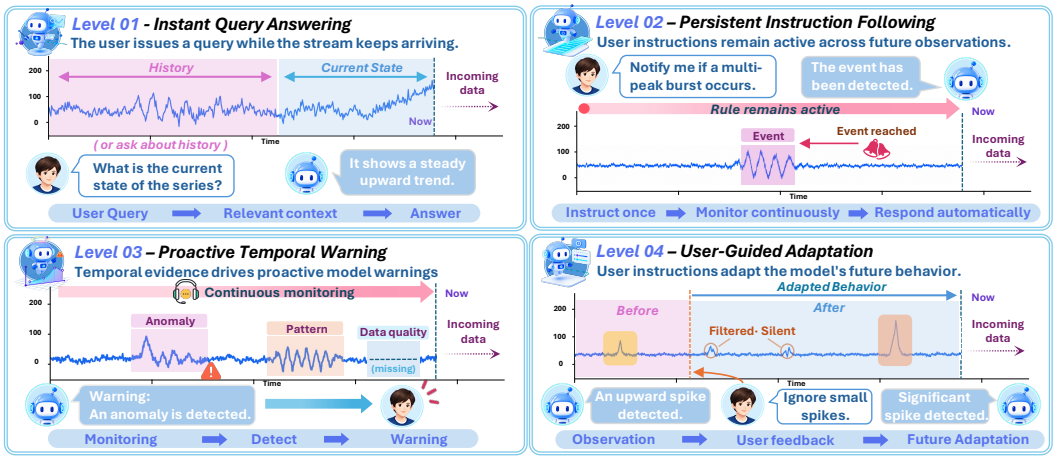


Figure 3: Representative cases of four streaming time-series interaction levels, from IQA to PIF, PTW, and UGA, corresponding to the capabilities of Understanding, Persistence, Initiative, and Adaptivity.

Data Sources. We collect synthetic and real-world time series in both univariate and multivariate settings (Figure 4). Following ChatTS (Xie et al., 2024), synthetic data are generated from configurable trends, seasonal patterns, local events, and noise, with event attributes, temporal locations, and cross-variable relationships retained as metadata. We further manually curate real-world recordings from 12 datasets across six domains: energy, finance, healthcare, human activity, environment, and manufacturing. These data introduce diverse domain-specific dynamics and natural variability.

Component Grounding. We then identify interaction-relevant temporal components spanning local events, interval-level patterns, cross-variable relationships, and data-quality issues. These components range from isolated spikes and sustained upward or downward trends to inter-variable dependencies and corrupted signal segments. ❶ *Synthetic data* use generation metadata to provide known temporal locations, whereas ❷ *real-world data* are processed through LLM-assisted annotation to identify plausible interaction scenarios and localize the corresponding temporal evidence. The resulting components provide explicit temporal evidence for subsequent interaction data construction.

Interaction Construction. The grounded temporal evidence is converted into task-specific interactions for IQA, PIF, PTW, and UGA. Generation combines task-specific templates and rules with LLM-based generation to balance controllability and diversity. Each interaction specifies an optional user query or instruction, a response position, and the corresponding response text, with responses

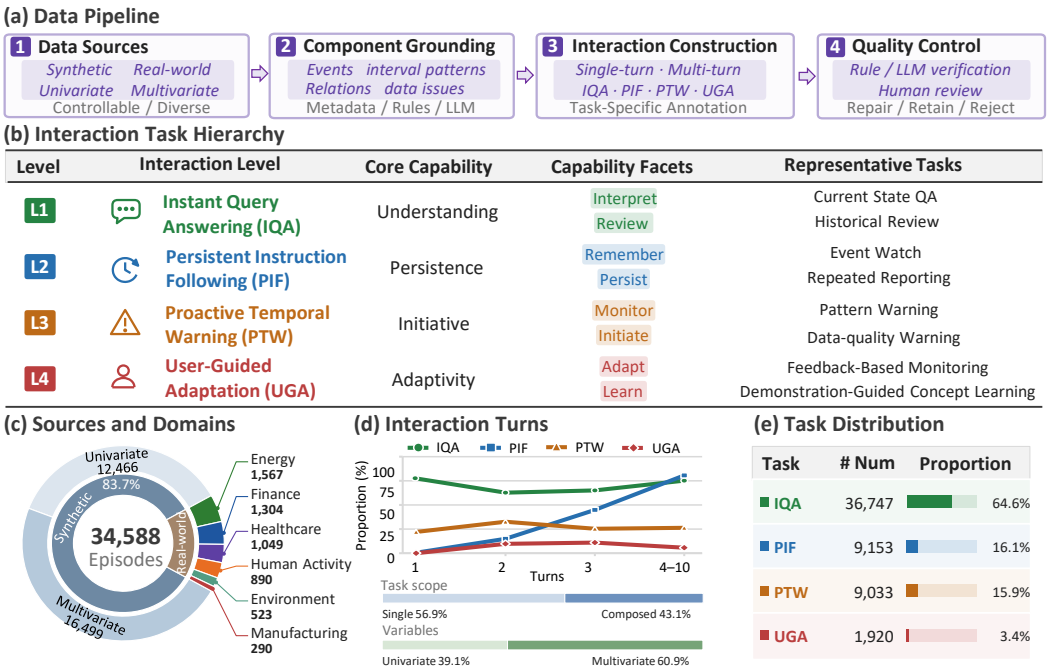


Figure 4: Construction and composition of STREAMTSI-34K. (a) Four-step data construction pipeline. (b) Interaction levels, capabilities, and representative tasks. (c) Source composition and real-world domain breakdown. (d) Task proportions by response count, alongside task scope and variable dimensionality. (e) Distribution of task instances, including tasks within composed episodes.

grounded only in observations available at or before the annotated position. Single-turn episodes are constructed to capture individual interaction behaviors, while multi-turn sessions compose multiple tasks so that earlier instructions, responses, and user feedback can influence subsequent interactions.

Quality Control. The generated interaction data first undergo LLM-based verification to ensure task consistency, appropriate response timing, and that each response relies only on observations available at the annotated position. Data that fail verification are regenerated using verifier feedback and re-evaluated, with this process repeated until they pass. Rule-based filtering then removes samples that violate task-specific requirements. High-confidence samples are retained directly, while low-confidence cases are manually reviewed to determine whether they should be accepted, corrected, or rejected. The complete data collection and construction procedure is provided in Appendix C.

5 EXPERIMENT

Baselines. We compare against three categories of representative models. **General LLMs:** Qwen2.5-7B-Instruct (Yang et al., 2025b), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), and Qwen3-14B (Yang et al., 2025a). **Vision-Language Models:** Qwen2.5-VL-7B (Bai et al., 2025) and InternVL3.5-8B (Wang et al., 2025). **Time-Series Language Models:** ChatTS (Xie et al., 2024) and TimeOmni-1 (Guan et al., 2026a). We report TIMEINTERACT as the streaming time-series interaction model.

Evaluation. The evaluation covers all four interaction levels using task-specific metrics, including ***Semantic Correctness (SC)*** for Understanding and ***Instruction Fulfillment Rate (IFR)*** for Persistence, alongside metrics for Initiative and Adaptivity. Results are reported for both single-turn and multi-turn settings to capture interaction quality under different conversational contexts. Real-time efficiency is assessed using latency and responsiveness measures, including ***Time to First Token (TTFT)*** and ***Completion Latency***, etc. Detailed metric definitions are provided in Appendix B.

5.1 MAIN RESULTS

We summarize the main results in Tables 1 and 2. **[Res.1] Strong interaction capability.** TIMEINTERACT ranks first across all four interaction levels in both settings, with the largest gains on UGA of

+22.22 points in single-turn and +23.92 points in multi-turn interactions. **[Res.2] Superior multi-turn performance.** In multi-turn interactions, TIMEINTERACT outperforms Qwen3-14B by 8.61 points on IQA and InternVL3.5-8B by 10.09 and 12.68 points on PIF and PTW, respectively, exceeding the best baseline per task. **[Res.3] Effective response triggering.** TIMEINTERACT achieves the highest F1 and NQ-F1 in both settings, indicating a better balance between responding and remaining silent. In multi-turn interactions, InternVL3.5-8B tends to over-trigger, reaching 98.24 recall but only 23.66 precision, whereas TimeOmni-1 is overly conservative and tends to suppress necessary responses.

Table 1: Performance on the four interaction levels of STREAMTSI-34K in single-turn and multi-turn settings. **Native Streaming** indicates whether the model natively supports streaming TS data input.

Model	Size	Model Input	Native Streaming	Single-turn				Multi-turn			
				IQA *(SC)*	PIF *(IFR)*	PTW *(CEHR)*	UGA *(ASR)*	IQA *(SC)*	PIF *(IFR)*	PTW *(CEHR)*	UGA *(ASR)*
General LLMs											
Qwen2.5-7B-Instruct	7B	Text	✗	44.14	3.45	2.70	0.00	48.30	2.52	1.41	0.00
Mistral-7B-Instruct-v0.3	7B	Text	✗	38.06	0.00	5.41	0.00	35.13	0.00	1.41	0.00
Qwen3-14B	14B	Text	✗	55.63	36.21	16.22	14.81	55.95	24.37	14.08	13.04
Vision-Language Models											
Qwen2.5-VL-7B	7B	Figure+Text	✗	48.20	5.17	10.81	0.00	44.30	4.20	7.04	0.00
InternVL3.5-8B	8B	Figure+Text	✗	55.18	43.10	35.14	22.22	53.16	33.61	25.35	8.70
Time-Series Language Models											
ChatTS	14B	TS+Text	✗	38.06	5.17	27.03	7.41	43.81	5.88	19.72	2.17
TimeOmni-1	7B	TS+Text	✗	35.81	0.00	0.00	0.00	41.87	0.00	2.82	0.00
TS Interaction Model											
TIMEINTERACT	4B	Streaming TS+Text	✓	**67.34**	**55.17**	**45.95**	**44.44**	**64.56**	**43.70**	**38.03**	**36.96**

Table 2: Response-triggering performance on STREAMTSI-34K in single-turn and multi-turn settings. **P**, **R**, **F1**, and **NQ-F1** denote triggering precision, recall, F1 score, and No-Query F1, respectively.

Model	Size	Model Input	Native Streaming	Single-turn				Multi-turn			
				P	R	F1	NQ-F1	P	R	F1	NQ-F1
General LLMs											
Qwen2.5-7B-Instruct	7B	Text	✗	54.67	53.56	54.11	5.30	83.60	62.87	71.77	7.60
Mistral-7B-Instruct-v0.3	7B	Text	✗	15.54	41.36	22.59	7.09	46.27	36.99	41.11	10.26
Qwen3-14B	14B	Text	✗	15.81	74.92	26.11	10.06	33.88	86.86	48.75	18.08
Vision-Language Models											
Qwen2.5-VL-7B	7B	Figure+Text	✗	14.91	49.49	22.92	5.67	48.55	58.94	53.24	9.80
InternVL3.5-8B	8B	Figure+Text	✗	8.76	**98.98**	16.10	7.65	23.66	**98.24**	38.14	16.91
Time-Series Language Models											
ChatTS	14B	TS+Text	✗	7.83	58.31	13.80	7.43	22.00	44.04	29.35	19.36
TimeOmni-1	7B	TS+Text	✗	45.60	38.64	41.83	8.76	**94.01**	46.75	62.44	8.12
TS Interaction Model											
TIMEINTERACT	4B	Streaming TS+Text	✓	**87.46**	82.71	**85.02**	**63.56**	89.97	83.88	**86.82**	**55.87**

5.2 STREAMING INTERACTION EFFICIENCY

Beyond interaction quality, we assess real-time latency. Table 3 reports response latency, stream processing efficiency, and speedup across response lengths. **(1) Low first-token latency.** TIMEINTERACT exhibits similar TTFT to interleaved inference, with negligible overhead from the Plan and control heads. **(2) Higher efficiency for longer responses.** By avoiding blocking between input processing and output generation, TIMEINTERACT achieves greater speedups for longer responses (offline: $1.80\times \rightarrow 2.15\times$; interleaved: $1.26\times \rightarrow 1.59\times$). **(3) Efficient streaming interaction.** Near-zero stall time and the lowest time per step across all groups enable efficient real-time interaction.

Table 3: Inference efficiency across different response lengths, reported in terms of response latency, stream processing, and overall speedup. **Bold** values indicate the best latency results in each group.

Response Length	Inference Strategy	Response Latency		Stream Processing		Overall
		TTFT ↓ (ms)	Completion Lat. ↓ (ms)	Avg. Stall Time ↓ (ms)	Time / Step ↓ (ms)	Speedup ↑ (×)
<50 tokens	Offline	1368.05	2944.94	—	0.56	1.80
	Interleaved	74.45	2065.39	856.50	0.39	1.26
	TIMEINTERACT (Ours)	**74.07**	**1636.33**	**≈ 0**	**0.31**	—
50–70 tokens	Offline	1578.12	3714.37	—	0.64	1.94
	Interleaved	73.98	2703.60	1286.19	0.47	1.41
	TIMEINTERACT (Ours)	**72.30**	**1913.09**	**≈ 0**	**0.33**	—
>70 tokens	Offline	2041.13	5175.18	—	0.67	2.15
	Interleaved	**72.12**	3829.70	2044.13	0.50	1.59
	TIMEINTERACT (Ours)	73.32	**2403.82**	**≈ 0**	**0.31**	—

5.3 Additional Analysis

Robustness to Longer Interactions. Figure 5 evaluates response triggering as interaction depth and stream length increase. Both metrics exhibit only moderate degradation as the interaction horizon grows, with F1 remaining consistently above 75 across all settings. This indicates that TimeInteract maintains stable response triggering over long interaction histories and streams.

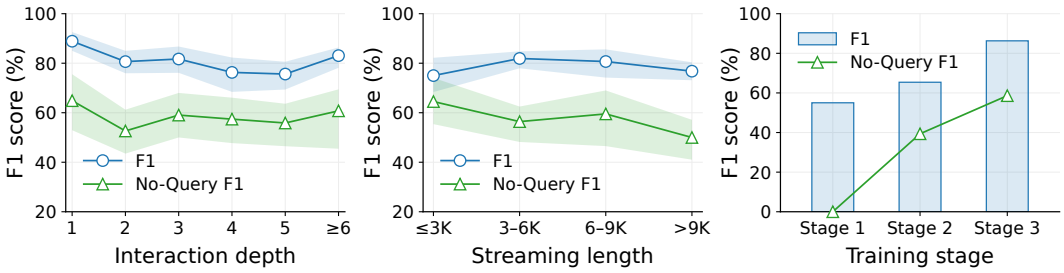


Figure 5: Analysis of interaction robustness and three-stage training. **Left and middle**: performance across interaction depths and stream lengths. **Right**: performance across three training stages.

Effectiveness of Three-Stage Training. Figure 5 shows that the three-stage training progressively improves the model's interaction capability. F1 increases from 55 → 65 → 86, while No-Query F1 rises substantially from 0 to 59. The gains from Stage I to Stage II suggest improved interaction understanding, while Stage III further boosts performance through large-scale streaming training.

Information Retention in Plan Tokens. Figure 6 examines whether Plan Tokens preserve information across interaction turns. After identifying a peak of 3406.9 ppm, TimeInteract correctly answers the follow-up as 3506.9 ppm using the Plan Token, despite the previous response text being absent. Replacing the Plan Token with a random embedding changes the answer to 1100 ppm. The token distribution also assigns lower probability to the correct continuation. These results show that Plan Tokens effectively retain information from previous interactions.

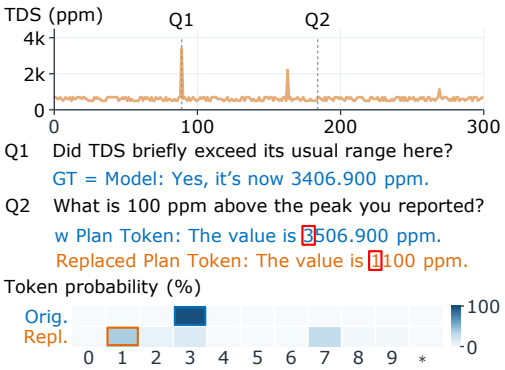


Figure 6: Visualization of Plan Token effectiveness in preserving multi-turn information.

Table 4: Ablation study of temporal representations and interaction mechanisms for triggering.

Variant	F1	NQ-F1
Temporal Representation		
Repl. Slow TS token	62.85	36.02
Repl. Fast TS token	71.16	45.05
Interaction Mechanism		
Repl. Plan token	76.13	48.53
TimeInteract	**86.30**	**58.61**

Ablation Study. We evaluate the contribution of each component by perturbing the embeddings of Slow TS tokens, Fast TS tokens, and Plan tokens. As shown in Table 4, all three perturbations reduce both F1 and NQ-F1. Perturbing Slow TS embeddings causes the largest degradation, indicating their greatest impact on response triggering. The performance drop caused by perturbing Plan embeddings further highlights the importance of retained interaction information for subsequent response decisions.

6 Conclusion

In this work, we introduce **Time-Series Interaction**, where models track evolving observations and user intent, decide when to respond, and continue processing inputs during generation. TimeInteract combines a dual-view streaming TS encoder, a response control mechanism, and decoupled streaming inference. We also establish a four-level interaction hierarchy and construct StreamTSI-34K. Experiments show that TimeInteract achieves strong interaction quality and effective response triggering. Our work marks a step toward interactive intelligence for streaming time series.

AI USE STATEMENT

We used generative AI tools as part of the data construction pipeline. Specifically, GPT-5.6-Terra was used for interaction scenario design, temporal grounding, interaction generation, and reannotation, while DeepSeek-V4-Pro was used for independent interaction verification. All generated annotations were subject to the verification, rule-based filtering, and human-review procedures described in the paper. Generative AI tools were also used to assist with language editing and improving the readability of the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses synthetic time series and publicly available real-world datasets spanning multiple application domains, including healthcare, human activity, finance, energy, environment, and manufacturing. We did not collect any new data from human participants. The provenance and available license information of the real-world datasets are reported in the paper, and the original data sources are appropriately cited.

REPRODUCIBILITY STATEMENT

We provide details to reproduce the proposed method, including its model design and training procedure. We also describe the construction and processing of the datasets, together with the evaluation setup used in our experiments. Additional implementation details and analyses are included in the appendix, and an anonymized link to the implementation is provided with the submission.

REFERENCES

Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav Gulavani, Alexey Tumanov, and Ramachandran Ramjee. Taming Throughput-Latency tradeoff in LLM inference with Sarathi-Serve. In *18th USENIX symposium on operating systems design and implementation (OSDI 24)*, pp. 117–134, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

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

Yuxuan Chen and Haoyuan Yu. From turn-taking to synchronous dialogue: A survey of full-duplex spoken language models. *arXiv preprint arXiv:2509.14515*, 2025.

Sylvain W Combettes, Paul Boniol, Antoine Mazarguil, Danping Wang, Diego Vaquero-Ramos, Marion Chauveau, Laurent Oudre, Nicolas Vayatis, Pierre-Paul Vidal, Alexandra Roren, et al. Armcoda: A data set of upper-limb human movement during routine examination. *Image Processing On Line*, 14:1–13, 2024.

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. *arXiv preprint arXiv:2410.00037*, 2024.

Yawei Dong, He Jiang, Bo Zeng, and Sheng Pan. Reinforcement learning driven periodic kernel fusion for probabilistic forecasting of market dynamics. *Knowledge-Based Systems*, pp. 115770, 2026.

Jiawei Gao, Bochao Chen, and Su-Kit Tang. Water quality monitoring: A water quality dataset from an on-site study in macao. *Applied Sciences*, 15(8):4130, 2025.

Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3, pp. 1–10, 2014.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. *arXiv preprint arXiv:2206.08317*, 2022.

Mohammad M Ghassemi, Benjamin E Moody, Li-Wei H Lehman, Christopher Song, Qiao Li, Haoqi Sun, Roger G Mark, M Brandon Westover, and Gari D Clifford. You snooze, you win: the physionet/computing in cardiology challenge 2018. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pp. 1–4. IEEE, 2018.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in neural information processing systems*, 36:19622–19635, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 152139–152170, 2026a.

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. *arXiv preprint arXiv:2602.17149*, 2026b.

Valerio Guerrini, Thibaut Germain, Charles Truong, Laurent Oudre, and Paul Boniol. Time series motif discovery: a comprehensive evaluation. *Proceedings of the VLDB Endowment (PVLDB)*, 18 (7):2226–2239, 2025.

Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. Condition monitoring of a complex hydraulic system using multivariate statistics. In *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 210–215. IEEE, 2015.

Yibo Hu, Yu Qian, Mao Gu, Yingfan Tao, Yuhao Chen, Yongdong Luo, Zhuoqun Liu, Meiguang Jin, and Junfeng Ma. Tlive-omni: An omni-modal understanding model for e-commerce live streaming. *arXiv preprint arXiv:2608.20958*, 2026.

Bosong Huang, Panzhen Zhao, Zengxiang Li, Patricia Lee, Wei Jin, Alan Wee-Chung Liew, Ming Jin, and Shirui Pan. Learning cardiac latent representations in vectorcardiogram space. *arXiv preprint arXiv:2605.31249*, 2026.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 23343–23351, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

He Jiang, Sheng Pan, Yao Dong, and Jianzhou Wang. Probabilistic electricity price forecasting based on penalized temporal fusion transformer. *Journal of Forecasting*, 43(5):1465–1491, 2024.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *International conference on learning representations*, volume 2024, pp. 23857–23880, 2024.

Iurii D. Katser and Vyacheslav O. Kozitsin. Skoltech anomaly benchmark (skab), 2020.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29736–29753, 2025.

Yaxuan Kong, Qingren Yao, Yuqi Nie, Yichen Li, Yilei Shao, Stefan Zohren, Anna Vettoruzzo, Joaquin Vanschoren, Ming Jin, and Qingsong Wen. Timesage-mt: A multi-turn benchmark for evaluating agentic time series reasoning. *arXiv preprint arXiv:2606.01498*, 2026.

Thinking Machines Lab. Interaction models: A scalable approach to human-ai collaboration. *Thinking Machines Lab: Connectionism*, May 2026. doi: 10.64434/tml.20260511. https://thinkingmachines.ai/blog/interaction-models/.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A Xu, Winnie Chow, Martin Maritsch, Robert Jakob, Ning Wang, Juncheng Liu, Aradhana Verma, et al. Opentslm: Time-series language models for reasoning over multivariate medical text-and time-series data. *arXiv preprint arXiv:2510.02410*, 2025.

Ying-yee Ava Lau, Zhiwen Shao, and Dit-Yan Yeung. Fast and slow streams for online time series forecasting without information leakage. In *International Conference on Learning Representations*, volume 2025, pp. 28736–28762, 2025.

Zechen Li, Keerthana Natarajan, Weizhi Zhang, Menglian Zhou, Simon A Lee, Yuwei Zhang, Maxwell A Xu, Zeinab Esmaeilpour, Flora D Salim, Mark Malhotra, et al. Glucofm: A dual-stream foundation model for continuous glucose monitoring. *arXiv preprint arXiv:2605.30865*, 2026.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18780–18788, 2025.

Jingyu Lu, Yuhan Wang, Jianming Luo, Yifu Chen, Tianle Liang, Shengpeng Ji, Ziyue Jiang, Xiaoda Yang, Yu Zhang, Xize Cheng, et al. A survey of full-duplex spoken dialogue systems: Architectural hierarchy, interaction ontology, and decision state machine. *arXiv preprint arXiv:2606.19453*, 2026.

George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50, 2001.

David Murray, Lina Stankovic, and Vladimir Stankovic. An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. *Scientific data*, 4(1):160122, 2017.

Felix Parker, Nimeesha Chan, Chi Zhang, and Kimia Ghobadi. Tsllm: Augmenting llms for general time series understanding and prediction. *arXiv preprint arXiv:2510.01111*, 2025.

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*, pp. 1–4, 2026.

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. *arXiv preprint arXiv:2602.12147*, 2026.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pp. 108–109. IEEE, 2012.

Stephen Rogers, Huidong Jin, Rose Roche, and Danielle A Way. Improved forecasting of cotton canopy temperature for irrigation scheduling. *Agricultural and Forest Meteorology*, 385:111192, 2026.

Changho Shin, Eunjung Lee, Jeongyun Han, Jaeryun Yim, Wonjong Rhee, and Hyoseop Lee. The enertalk dataset, 15 hz electricity consumption data from 22 houses in korea. *Scientific data*, 6(1): 193, 2019.

Biao Sun, Ziming Huang, Hanyu Zhao, Wencong Xiao, Xinyi Zhang, Yong Li, and Wei Lin. Llumnix: Dynamic scheduling for large language model serving. In *18th USENIX symposium on operating systems design and implementation (OSDI 24)*, pp. 173–191, 2024a.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. In *International Conference on Learning Representations*, volume 2024, pp. 37854–37881, 2024b.

Junlong Tong, Jinlan Fu, Zixuan Lin, Yingqi Fan, Anhao Zhao, Hui Su, and Xiaoyu Shen. Llm as effective streaming processor: Bridging streaming-batch mismatches with group position encoding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 23497–23517, 2025.

Junlong Tong, Yingqi Fan, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. Streamingthinker: Large language models can think while reading. In *International Conference on Learning Representations*, volume 2026, pp. 31470–31498, 2026a.

Junlong Tong, Zilong Wang, YuJie Ren, Peiran Yin, Hao Wu, Wei Zhang, and Xiaoyu Shen. From static inference to dynamic interaction: A survey of streaming large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10237–10263, 2026b.

Junlong Tong, Yao Zhang, Anhao Zhao, Yingqi Fan, Yunpu Ma, and Xiaoyu Shen. Proactivellm: Learning active interaction for streaming large language models. *arXiv preprint arXiv:2606.00523*, 2026c.

Lu Wang, Zhuoran Jin, Yupu Hao, Yubo Chen, Kang Liu, Yulong Ao, and Jun Zhao. Think while watching: Online streaming segment-level memory for multi-turn video reasoning in multimodal large language models. *arXiv preprint arXiv:2603.11896*, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024a.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*, 2024b.

Qing Wei, Hao Miao, Yan Zhao, Kai Zheng, Bin Yang, Volker Markl, and Christian S Jensen. Evolving proxy kills drift: Data-efficient streaming time series anomaly detection. In *Proceedings of the ACM Web Conference 2026*, pp. 7295–7306, 2026.

B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962. doi: 10.1080/00401706.1962.10490022.

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

Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024.

Zhifei Xie, Zihang Liu, Ze An, Xiaobin Hu, Yue Liao, Ziyang Ma, Dongchao Yang, Mingbao Lin, Deheng Ye, Shuicheng Yan, et al. Audio interaction model. *arXiv preprint arXiv:2606.05121*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025b.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Haowen Hou, Zheming Liang, Congcong Wang, Yuhang Cao, Shenglong Ye, Shuai Xie, et al. Joyai-vl-interaction: Real-time vision-language interaction intelligence. *arXiv preprint arXiv:2606.14777*, 2026.

Fangxu Yu, Xingang Guo, Lingzhi Yuan, Haoqiang Kang, Hongyu Zhao, Lianhui Qin, Furong Huang, Bin Hu, and Tianyi Zhou. Tsrbench: A comprehensive multi-task multi-modal time series reasoning benchmark for generalist models. *arXiv preprint arXiv:2601.18744*, 2026.

Xiaofan Yu, Lanxiang Hu, Benjamin Reichman, Dylan Chu, Rushil Chandrupatla, Xiyuan Zhang, Larry Heck, and Tajana S Rosing. Sensorchat: Answering qualitative and quantitative questions during long-term multimodal sensor interactions. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–35, 2025.

Jialiang Zhang, Junlong Tong, Junyan Lin, Hao Wu, Yirong Sun, Yunpu Ma, and Xiaoyu Shen. Think-as-you-see: Streaming chain-of-thought reasoning for large vision-language models. *arXiv preprint arXiv:2603.02872*, 2026.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*, 2024.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX symposium on operating systems design and implementation (OSDI 24)*, pp. 193–210, 2024.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Yitong Zhou, Yucong Luo, Mingyue Cheng, Qi Liu, Jiahao Wang, Daoyu Wang, and Enhong Chen. Time series forecasting as reasoning: A slow-thinking approach with reinforced llms. *arXiv preprint arXiv:2506.10630*, 2025.

A Online Historical Statistics Update

We maintain the historical reference without storing or revisiting past observations. At each streaming step, we construct both normalized views before merging the current chunk into the historical statistics. This ordering avoids leakage from the current chunk into its historical reference.

Reference Statistics and Normalization. Before processing $\mathbf{C}_t$, vectors $\mathbf{n}, \boldsymbol{\mu}, \mathbf{s} \in \mathbb{R}^M$ store historical counts, means, and centered sums of squares. They are initialized to zero. The current chunk yields statistics $\mathbf{n}^c, \boldsymbol{\mu}^c, \mathbf{s}^c$ from valid observations. For positive counts and valid positions, standard deviations and normalized views are computed element-wise with $\epsilon = 10^{-5}$:

$$\boldsymbol{\sigma}^c = \sqrt{\max\left(\frac{\mathbf{s}^c}{\mathbf{n}^c}, \epsilon^2\right)}, \qquad \boldsymbol{\sigma} = \sqrt{\max\left(\frac{\mathbf{s}}{\mathbf{n}}, \epsilon^2\right)}, \tag{10}$$

$$\hat{\mathbf{C}}_t^F = \frac{\mathbf{C}_t - \boldsymbol{\mu}^c}{\boldsymbol{\sigma}^c + \epsilon}, \qquad \hat{\mathbf{C}}_t^H = \frac{\mathbf{C}_t - \boldsymbol{\mu}}{\boldsymbol{\sigma} + \epsilon}. \tag{11}$$

The statistics are broadcast over time for each variable. The Fast View captures local shape within the current chunk, while the Historical View measures deviations from the preceding stream. For $\mathbf{n} = 0$, both views use $\boldsymbol{\mu}^c, \boldsymbol{\sigma}^c$. Padding values are excluded from statistics and zeroed in both views.

Historical State Update. After forming both normalized views, we merge the current-chunk statistics into the historical state using a chunk-wise Welford update (Welford, 1962). Let $\boldsymbol{\delta} = \boldsymbol{\mu}^c - \boldsymbol{\mu}$ denote the mean difference, and let primes indicate updated quantities. For entries with $\mathbf{n}^c > 0$, the historical statistics are updated element-wise to incorporate the current chunk as follows:

$$\begin{aligned} \mathbf{n}' = \mathbf{n} + \mathbf{n}^c, \qquad \boldsymbol{\mu}' = \boldsymbol{\mu} + \frac{\mathbf{n}^c}{\mathbf{n}'} \odot \boldsymbol{\delta}, \\ \mathbf{s}' = \mathbf{s} + \mathbf{s}^c + \frac{\mathbf{n} \odot \mathbf{n}^c}{\mathbf{n}'} \odot \boldsymbol{\delta}^2. \end{aligned} \tag{12}$$

Observation counts determine the relative weights of the historical and current-chunk means, while the correction in $\mathbf{s}'$ accounts for their difference. The merged statistics represent all valid observations accumulated up to the current chunk and provide the reference for normalizing the next chunk, leaving the current normalized views unchanged. Entries with $\mathbf{n}^c = 0$ retain their previous statistics.

B Implementation Details

Training Setup. We initialize TimeInteract with Qwen3-4B-Instruct-2507 and follow the three-stage training pipeline. Stage I optimizes the streaming TS encoder, control and Plan heads while freezing the pretrained language model; Stages II and III jointly fine-tune all modules. Each subsequent stage starts from the preceding stage's model weights. All stages use AdamW with cosine learning-rate decay, 5% warmup, BF16 precision, and DeepSpeed ZeRO-2. Training takes approximately five days on four H100 GPUs (94 GB each). Table 5 summarizes the training configuration.

Interaction Quality. We evaluate the four interaction tasks using metrics aligned with their respective objectives. For IQA, ***Semantic Correctness (SC)*** assigns each response a score $s_i^{\mathrm{SC}} \in [0, 100]$, averaged over N_{IQA} instances. For PIF, ***Instruction Fulfillment Rate (IFR)*** aggregates binary fulfillment indicators $f_i \in \{0, 1\}$ over N_{PIF} instances. For PTW, ***Correct Event Hit Rate (CEHR)*** measures the proportion of correctly warned events, denoted by N_{hit}, among N_{event} annotated events. For UGA, ***Adaptation Success Rate (ASR)*** measures the proportion of successful adaptations, denoted by N_{success}, among N_{UGA} evaluated adaptations. These metrics are defined as follows:

$$\begin{aligned} \mathrm{SC} &= \frac{1}{N_{\mathrm{IQA}}} \sum_{i=1}^{N_{\mathrm{IQA}}} s_i^{\mathrm{SC}}, & \mathrm{IFR} &= \frac{100}{N_{\mathrm{PIF}}} \sum_{i=1}^{N_{\mathrm{PIF}}} f_i, \\ \mathrm{CEHR} &= 100 \frac{N_{\mathrm{hit}}}{N_{\mathrm{event}}}, & \mathrm{ASR} &= 100 \frac{N_{\mathrm{success}}}{N_{\mathrm{UGA}}}. \end{aligned} \tag{13}$$

Interaction Efficiency. We evaluate real-time efficiency through response latency, stream continuity, and processing cost. ***Time to First Token (TTFT)*** (Zhong et al., 2024) and ***Completion Latency***

Table 5: Hyperparameters and computational settings for the three-stage training pipeline.

Parameter	Stage I	Stage II	Stage III
Optimization			
Epochs	1	1	3
LLM learning rate	Frozen	1×10^{-6}	1×10^{-5}
TS encoder and heads learning rate	2×10^{-4}	5×10^{-5}	1×10^{-5}
Optimizer	AdamW	AdamW	AdamW
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Adam ϵ	10^{-8}	10^{-8}	10^{-8}
Learning-rate schedule	Cosine decay	Cosine decay	Cosine decay
Warmup ratio	0.05	0.05	0.05
Weight decay	0.01	0.01	0.01
Maximum gradient norm	1.0	1.0	1.0
Training objectives			
Loss weights $(\lambda_c, \lambda_f, \lambda_d)$	$(0.2, 1, 0)$	$(1, 0.5, 0.5)$	$(1, 0.5, 0.5)$
Control weights (`silent:respond`)	1 : 5	1 : 5	1 : 5
Distillation temperature τ	–	2	2
Execution			
GPUs		4× H100 (94 GB each)	
Precision		BF16	
Sharding		DeepSpeed ZeRO-2	
Total training time (all stages)		5 days (approx.)	

(Agrawal et al., 2024) measure the delays from triggering to response initiation and completion, respectively. We denote the trigger time, first-token generation time, and response completion time by t_{trig}, t_{first}, and t_{end}. ***Average Stall Time*** measures how much response generation blocks incoming observations, averaging the blocked duration $\Delta t_i^{\text{stall}}$ over N_r triggered responses. ***Time per Step*** (Sun et al., 2024a) measures the average processing cost of an incoming step, using the total stream-processing time T_{stream} over N_{step} steps. Their mathematical definitions are as follows:

$$\begin{aligned} \text{TTFT} &= t_{\text{first}} - t_{\text{trig}}, & T_{\text{comp}} &= t_{\text{end}} - t_{\text{trig}}, \\ \bar{T}_{\text{stall}} &= \frac{1}{N_r} \sum_{i=1}^{N_r} \Delta t_i^{\text{stall}}, & T_{\text{step}} &= \frac{T_{\text{stream}}}{N_{\text{step}}}. \end{aligned} \tag{14}$$

All four metrics are reported in milliseconds, with lower values indicating better efficiency. Near-zero stall time indicates that incoming observations can be processed with minimal interruption during response generation. We further report ***Speedup*** to compare inference strategies, computed as $\text{Speedup} = T_{\text{comp}}^{\text{baseline}} / T_{\text{comp}}^{\text{ours}}$, where $T_{\text{comp}}^{\text{ours}}$ denotes the completion latency of TIMEINTERACT.

Evaluation Details. We evaluate on 200 randomly sampled single-turn and 200 multi-turn episodes. The train–test split is performed at the original-sample level, where all episodes derived from the same underlying sample are grouped and assigned exclusively to a single split. This ensures that no source sample contributes to both training and testing, preventing overlap between the two splits.

C Dataset Details

C.1 Real-World Data Sources

The real-world portion of STREAMTSI-34K contains 5,623 interaction episodes constructed from 12 public datasets across six domains: energy, finance, healthcare, human activity, environment, and manufacturing. These sources cover heterogeneous temporal dynamics, sampling frequencies, signal modalities, and annotation schemes in both univariate and multivariate settings. For each source, we select variables suitable for streaming interaction and retain available event information for subsequent component grounding. Table 6 provides detailed information about the datasets.

Energy. We use electricity-consumption signals from ENERTALK (Shin et al., 2019) and REFIT (Murray et al., 2017). For ENERTALK, we retain active-power channels from House 06, align

Table 6: Statistics and composition of the real-world data in STREAMTSI-34K, covering 12 public datasets across six domains. The sources span diverse sampling frequencies, sequence lengths, and signal dimensionalities, providing a heterogeneous basis for constructing time-series interactions

Dataset	Dim	Freq. (Hz)	Length			Episodes		Responses		Source	License
			Min	Max	Avg	Count	Share (%)	Count	Share (%)		
Energy											
ENERTALK	6	15	5,018	7,997	6,665.10	1,467	26.09	4,781	31.63	Figshare	CC0 1.0
REFIT	1	0.13	5,000	7,989	5,763.90	100	1.78	200	1.32	Strathclyde	CC BY 4.0
Finance											
BTCUSDT	4	4	5,000	5,000	5,000.00	1,304	23.19	3,904	25.83	Kaggle	MIT
Healthcare											
JIGSAWMaster	1	30	3,000	3,000	3,000.00	245	4.36	769	5.09	JHU–ISI	Academic
PhysioNet 2018	6	200	8,000	8,000	8,000.00	459	8.16	715	4.73	PhysioNet	ODC-By 1.0
MIT–BIH	2	360	2,920	16,675	6,703.00	345	6.14	1,019	6.74	PhysioNet	ODC-By 1.0
Human Activity											
PAMAP2	6	100	6,000	8,000	7,381.80	598	10.63	1,196	7.91	UCI	CC BY 4.0
Arm-CODA	1	100	6,603	9,420	8,047.40	192	3.41	384	2.54	IPOL	CC BY-NC-SA 3.0
PTT PPG	1	500	1,061	3,296	2,169.30	100	1.78	350	2.32	PhysioNet	ODbL 1.0
Environment											
Macao Water	4	0.20	6,000	6,000	6,000.00	523	9.30	1,109	7.34	GitHub	Not stated
Manufacturing											
SKAB	5	1	745	1,327	1,133.10	30	0.53	30	0.20	GitHub	GPL-3.0
UCI Hydraulic	3	100	6,000	24,000	14,571.40	260	4.62	659	4.36	UCI	CC BY 4.0

independently timestamped measurements to a common temporal grid, and fill missing values. For REFIT, we combine dishwasher, washing-machine, and tumble-dryer loads from selected households into univariate sequences sampled at 8-second intervals (Guerrini et al., 2025).

Finance. Financial sequences are drawn from BTCUSDT, which records the limit order book of Binance's Bitcoin–Tether perpetual contract. The source covers January 9–20, 2023, with approximately 3.73 million observations, providing high-frequency market recordings for interaction construction.

Healthcare. The healthcare sources cover surgical motion, sleep physiology, and cardiac activity. For JIGSAWMaster, we use selected single-channel sequences from JIGSAWS (Gao et al., 2014), with labeled gesture examples and recurrence intervals supporting interactions around demonstrated and recurring motion patterns. From PhysioNet 2018 (Ghassemi et al., 2018), we select intervals from ten records and retain six synchronized channels. For MIT–BIH (Moody & Mark, 2001), we extract ECG segments using cardiologist-reviewed beat annotations and AAMI beat-group labels.

Human Activity. We combine wearable motion measurements, optical motion recordings, and physiological signals collected during physical activity. From PAMAP2 (Reiss & Stricker, 2012), we retain six accelerometer channels and construct contiguous 60–80-second windows. For Arm-CODA (Combettes et al., 2024), we use repeated segments to construct interactions involving movement recurrence. Pulse Transit Time PPG (Pollard et al., 2026) provides physiological and motion recordings collected during sitting, walking, and running, covering different activity conditions.

Environment. Environmental data come from the Macao Water Quality dataset (Gao et al., 2025), which monitors household tap-water conditions. Its four channels measure pH, turbidity, water temperature, and total dissolved solids, capturing changes in water quality.

Manufacturing. We use industrial sensor recordings from SKAB (Katser & Kozitsin, 2020) and the UCI Hydraulic dataset (Helwig et al., 2015). SKAB records a laboratory water-circulation system under normal and abnormal conditions accompanied by anomaly and change-point annotations. The UCI Hydraulic dataset provides pressure, electrical power, flow, temperature, vibration, and efficiency measurements from repeated 60-second operating cycles, together with cycle-level condition labels.

C.2 SYNTHETIC TIME-SERIES GENERATION

Following ChatTS (Xie et al., 2024), we synthesize base signals as $x_i = g(i) + s(i) + \epsilon(i) + \ell(i)$ for $i = 1, \ldots, L$, where L is the sequence length and the four terms represent trend, seasonality, noise, and local events, respectively. Their parameters control component shape, magnitude, and temporal extent. Multivariate and recurrent variants introduce shared shapes, temporally aligned events, and

repeated templates. Table 7 summarizes representative constructions and their controlled parameters; the corresponding metadata supports subsequent interaction annotation (Appendix C.4).

Table 7: Representative mechanisms and controlled parameters for synthetic time-series generation.

Structure	Generation mechanism	Controlled parameters
Component composition		
Trend g	e.g., constant, monotonic, or piecewise curves	Direction, magnitude, turning points
Seasonality s	e.g., harmonic, square, or triangular waves	Waveform, period, amplitude
Noise ϵ	e.g., Gaussian noise with optional oscillations	Noise magnitude relative to signal scale
Local events ℓ	e.g., spikes, dips, excursions, and level changes	Type, onset, duration, magnitude
Multivariate composition		
Shared shape	Shared shape with trend, seasonality, and noise	Prototype perturbation, mixing weights, scale, offset
Aligned events	Aligned events with trend, seasonality, and noise	Timing jitter, event type, width, amplitude
Pattern recurrence		
Repeated template	Shifted and rescaled copies of a fixed waveform	Occurrence count, spacing, amplitude, target channel

C.3 Real-World Interaction Annotation Pipeline

Model Configuration. We use GPT-5.6-Terra (`gpt-5.6-terra`) with `xhigh` reasoning effort as LLM1 for scenario design, temporal grounding, interaction generation, and reannotation. DeepSeek-V4-Pro (`deepseek-v4-pro`) serves as LLM2 for independent interaction verification, with a sampling temperature of 0.7. The two models are assigned distinct roles to separate interaction construction from quality assessment during annotation. Algorithm 2 summarizes the complete annotation workflow, and the corresponding prompt templates are provided in Appendix C.5.

Repeated Verification. We perform $n_v = 3$ independent verification runs for each time series in every round. In each run, LLM2 evaluates the complete interaction set and returns a `PASS` or `FAIL` verdict, a confidence score, and an explanation for each failed interaction. The verification criteria cover temporal grounding, task consistency, response timing, future-information leakage, and unsupported claims. We aggregate the three independent judgments by majority vote to improve the robustness of the verification process. An interaction is selected for reannotation when at least $m = \lfloor n_v/2 \rfloor + 1$ runs return `FAIL`. Under our setting, this corresponds to at least two failed judgments. Only interactions meeting this criterion are passed to the subsequent reannotation stage.

Failed-Interaction Reannotation. LLM1 receives the time series, interaction scenarios and temporal evidence, the interaction set, and verification feedback. It regenerates only the interactions selected for reannotation, while all others remain unchanged. Each revised interaction may update the response position, user input, and response content. The updated interaction set is then evaluated through n_v fresh LLM2 runs, with votes recomputed independently at each round rather than accumulated across rounds. This process continues until no interaction receives a majority of `FAIL` verdicts. With R reannotation rounds, the procedure requires $n_v(R + 1)$ verification calls during the full process.

Final Curation. After LLM-based verification is completed, we first apply rule-based checks to enforce annotation structure and task-specific constraints. Samples that pass these checks are then evaluated using a confidence threshold τ_{conf} based on the verification confidence scores. High-confidence samples are retained automatically, while samples with confidence below τ_{conf} are sent for manual review and are subsequently accepted, corrected, or rejected.

C.4 Synthetic Interaction Annotation

We generate synthetic interactions from temporal metadata using task-specific templates, with the component–task mappings summarized in Table 8. (1) *Task assignment:* we select relevant compo-

Algorithm 2 Streaming Interaction Annotation and Curation

Input: series/metadata $\{(\mathbf{X}_i, \mathcal{M}_i)\}$, templates/rules $\mathcal{T}$, verification count n_v, confidence threshold τ_{conf}
Output: curated dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$; $m \leftarrow \lfloor n_v/2 \rfloor + 1$
for each $(\mathbf{X}_i, \mathcal{M}_i)$ **do**
  ▷ **Phase 1: Interaction Construction** // $\mathcal{S}$: scenarios; $\mathcal{G}$: evidence
  $(\mathcal{S}, \mathcal{G}) \leftarrow$ LLM1-DESIGNANDGROUND$(\mathbf{X}_i, \mathcal{M}_i, \mathcal{T})$
  $\mathcal{A} = (a_j)_j \leftarrow$ LLM1-GENERATE$(\mathbf{X}_i, \mathcal{S}, \mathcal{G}, \mathcal{T})$
  repeat
  ▷ **Phase 2: Repeated Verification** // v: verdict; c: confidence; f: feedback
    for verification run $k = 1, \ldots, n_v$ **do**
      $\mathcal{V}^{(k)} = \{(v_j^{(k)}, c_j^{(k)}, f_j^{(k)})\}_j \leftarrow$ LLM2-VERIFY$(\mathbf{X}_i, \mathcal{S}, \mathcal{G}, \mathcal{A})$
    end for
    $\mathcal{F} \leftarrow \left\{ j : \sum_{k=1}^{n_v} \mathbf{1}[v_j^{(k)} = \texttt{FAIL}] \geq m \right\}$
    if $\mathcal{F} \neq \emptyset$ **then**
  ▷ **Phase 3: Selective Reannotation**
      $\mathcal{A}_{\mathcal{F}} \leftarrow$ LLM1-REANNOTATE$(\mathbf{X}_i, \mathcal{S}, \mathcal{G}, \mathcal{A}, \{\mathcal{V}^{(k)}\}_{k=1}^{n_v}, \mathcal{F})$
    end if
  until $\mathcal{F} = \emptyset$
  ▷ **Phase 4: Final Curation**
  for $j \in$ RULEFILTER$(\mathcal{A}, \mathcal{T})$ **do**
    $a'_j \leftarrow a_j$
    if AGGREGATECONFIDENCE$(\{c_j^{(k)}\}_{k=1}^{n_v}) < \tau_{\text{conf}}$ **then**
      $a'_j \leftarrow$ HUMANREVIEW$(\mathbf{X}_i, \mathcal{S}, \mathcal{G}, a_j)$ // $\emptyset$ if rejected
    end if
    $\mathcal{D} \leftarrow \mathcal{D} \cup \left(\{a'_j\} \setminus \{\emptyset\}\right)$
  end for
end for
return $\mathcal{D}$

nent attributes and variables to construct current-state queries, historical reviews, and monitoring instructions, while other components remain as background. (2) *Response generation:* templates determine response timing and content. IQA answers describe the requested evidence; PIF notifications follow matching events or reporting schedules; PTW warnings target salient events or injected data corruption; and UGA adapts response behavior according to user instructions or feedback, while also learning user-provided concepts and identifying their subsequent occurrences. (3) *Multi-turn composition:* we construct multi-turn interactions by combining compatible single-turn interactions from the same time series in chronological order, preserving user instructions and responses. Each multi-turn interaction contains at most ten model responses along the time-series stream.

C.5 Annotation Prompt Templates

We provide the prompt templates for scenario design and temporal grounding, interaction generation, verification, and regeneration. Instructions and input placeholders are retained; detailed examples and output schemas are abbreviated as {`Example Return`} and {`Output Format`}, respectively.

Prompt 01: Scenario Design and Component Grounding

Role

You are a time-series interaction annotator responsible for designing interactive scenarios and identifying the temporal evidence that supports each scenario.

Task

- From `Instant Query Answering (IQA)`, `Persistent Instruction Following (PIF)`, `Proactive Temporal Warning (PTW)`, and `User-Guided Adaptation (UGA)`, select **the most suitable task types** for the provided time series and design **one or more appropriate interaction scenarios**.
- Multiple scenarios may belong to the same task type.

Table 8: Task-specific temporal evidence and interaction construction patterns for synthetic annotation. ✓ denotes eligible evidence and – unused evidence for each task. Eligible sources may be used independently or jointly, while other components remain as background throughout the sequence.

Level	Task	Temporal Evidence					Interaction Pattern
		Trend	Season-ality	Cross-var.	Event	Corrupt.	
IQA	Current State QA	✓	✓	✓	✓	–	Attribute query → current-state response
	Historical Review	✓	–	✓	✓	–	Past interval → trend/event review
PIF	Event Watch	–	–	✓	✓	–	Monitoring instruction → event-triggered report
	Repeated Reporting	–	–	✓	–	–	Reporting schedule → periodic notification
PTW	Pattern Warning	–	–	✓	✓	–	Salient event → proactive warning
	Data-Quality Warning	–	–	✓	–	✓	Corrupted interval → data-quality warning
UGA	Feedback-Based Monitoring	–	–	✓	✓	–	Feedback-adjusted criterion → selective alert
	Demonstration-Guided Concept Learning	–	–	✓	✓	–	New example → recurrence-triggered report

- For each scenario:
  - Describe the interaction that could naturally occur based on the observed temporal behavior.
  - Identify **temporal evidence instances** in the time series that support the scenario.
- One scenario may contain multiple temporal evidence instances occurring at different intervals.
- Only construct scenarios and temporal evidence that are clearly supported by the data.
- Do not infer unsupported causes or explanations.

Example

{example_return}

Variable Descriptions

{variable_descriptions}

Time-Series Data

{time_series_data}

Output Format

{output_format}

Prompt 02: Scenario-Grounded Interaction Generation

Role

You are a time-series interaction annotator responsible for generating user-assistant interactions based on given interaction scenarios and their temporal evidence.

Scenarios

The following JSON array contains all scenarios for one time-series sample. Process the scenarios in the listed order.

{scenarios}

Task

Generate user-assistant interaction samples that naturally instantiate the given scenarios.

- Generate one interaction for each appropriate **response-triggering temporal evidence**, in the same order as the supplied temporal evidences.
- Select an appropriate `query_point` where sufficient evidence is available to support the response.
- Use only observations at or before `query_point`.
- Ground each interaction in the corresponding temporal evidence and scenario.
- Keep the interactions **concise and consistent** with the corresponding scenario and task type.
- If no user input occurs at the response point, set `question` to an empty string.
- Do not use future observations or infer unsupported causes or explanations.

Example

{example_return}

Variable Descriptions

{variable_descriptions}

Time-Series Data

{time_series_data}

Output Format

{output_format}

Prompt 03: Evidence-Grounded Interaction Verification

Role

You are a time-series interaction verifier responsible for evaluating generated user-assistant interactions against the given scenarios and temporal evidence.

Scenarios

{scenarios}

Generated Interactions

{generated_interactions}

Task

Evaluate whether each generated interaction is valid with respect to its corresponding scenario and temporal evidence.

- Check whether each interaction is **consistent with the corresponding scenario, task type, and temporal evidence**.
- Check whether each `query_point` is appropriate and whether sufficient evidence is available at that point.
- Check whether each question and answer accurately reflects the observed time-series behavior.
- Check whether any interaction uses **future observations** or introduces unsupported claims.
- Return `PASS` only if the interaction is valid.
- Report `confidence` from 0.0 to 1.0 for each interaction verdict.

Example

{example_return}

Variable Descriptions

{variable_descriptions}

Time-Series Data

{time_series_data}

Output Format

{output_format}

Prompt 04: Verification-Guided Interaction Regeneration

Role

You are a time-series interaction annotator responsible for regenerating user-assistant interactions that did not pass verification based on given interaction scenarios and their temporal evidence.

Scenarios

The following JSON array contains the scenarios corresponding to the failed interactions.

{scenarios}

Failed Verification Result

{failed_verification_result}

Task

Regenerate user-assistant interaction samples that instantiate the given scenarios while correcting the problems identified during verification.

- Generate new interactions rather than reproducing the failed interactions.
- Correct all problems identified in the corresponding **verification result**.
- Generate one interaction for each appropriate **response-triggering temporal evidence**, in the same order as the supplied temporal evidences.
- Select an appropriate `query_point` where sufficient evidence is available to support the response.
- Use only observations at or before `query_point`.
- Ground each interaction in the corresponding temporal evidence and scenario.
- Keep the interactions concise and consistent with the corresponding scenario and task type.
- If no user input occurs at the response point, set `question` to an empty string.
- Do not use future observations or infer unsupported causes or explanations.

Variable Descriptions

{variable_descriptions}

Time-Series Data

{time_series_data}

Output Format

{output_format}

D Additional Experiments

Analysis of Plan Tokens. Figure 7 examines the role of Plan Tokens in subsequent response triggering and history compression. On PIF (left), perturbing Plan embeddings widens the F1 gap across the first three interaction rounds, with third-round F1 dropping from 81 to 42. For history compression (right), we compare retaining only Plan Tokens with retaining the full response history. In the pooled ≥ 6 bin, mean history length drops from 190 to 25 tokens, a reduction of 86.8%. These results show that Plan Tokens serve as a compact history representation for subsequent response triggering.

Out-of-Domain Evaluation. We further evaluate response triggering on an additional OOD dataset containing 200 interaction episodes from eight real-world domains, with 100 episodes from each annotated single-turn and multi-turn subset. All eight models are evaluated on the same sampled

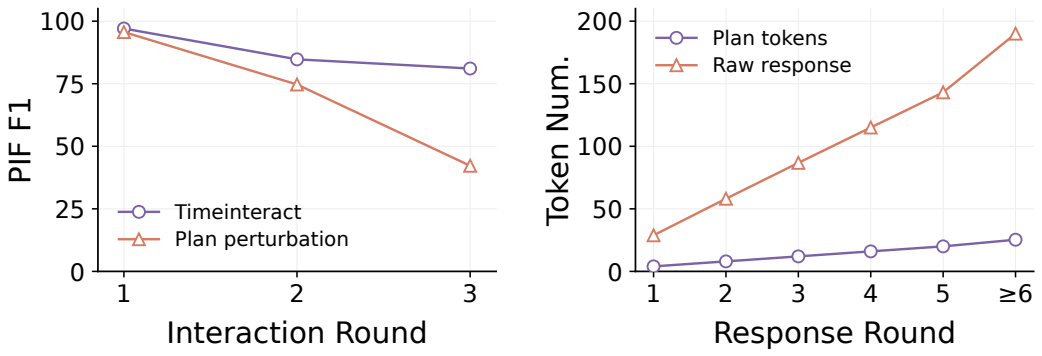


Figure 7: Analysis of Plan Tokens. Left: PIF F1 across rounds with original and perturbed Plan embeddings. Right: mean history token counts for Plan Tokens and full response history.

episodes, each containing at most six variables. We retain the model settings and response-triggering metrics used in the main evaluation and report precision, recall, F1, and NQ-F1.

As shown in Table 9, TIMEINTERACT achieves the highest precision, F1, and NQ-F1 in both single-turn and multi-turn OOD settings. In the multi-turn setting, its No-Query F1 reaches 37.40%, outperforming the strongest baseline by 21.87 percentage points. These results indicate that TIMEINTERACT generalizes well to OOD streams while maintaining effective response triggering.

Table 9: Response-triggering performance on the OOD test set. All scores are percentages. Bold and underlined values indicate the best and second-best results in each column, respectively.

Model	Size	Model Input	Native Streaming	Single-turn				Multi-turn			
				P	R	F1	NQ-F1	P	R	F1	NQ-F1
General LLMs											
Qwen2.5-7B-Instruct	7B	Text	✗	<u>55.86</u>	43.78	<u>49.09</u>	<u>26.00</u>	<u>82.67</u>	65.23	<u>72.93</u>	9.60
Mistral-7B-Instruct-v0.3	7B	Text	✗	16.31	57.30	25.39	13.94	35.94	60.94	45.22	7.96
Qwen3-14B	14B	Text	✗	22.11	<u>71.35</u>	33.76	20.03	36.63	72.27	48.62	<u>15.53</u>
Vision-Language Models											
Qwen2.5-VL-7B	7B	Figure+Text	✗	10.07	23.24	14.05	8.53	17.68	22.66	19.86	9.68
InternVL3.5-8B	8B	Figure+Text	✗	9.92	**100.00**	18.05	11.58	13.95	**100.00**	24.49	9.62
Time-Series Language Models											
ChatTS	14B	TS+Text	✗	10.85	60.00	18.38	13.84	11.07	32.81	16.55	9.77
TimeOmni-1	7B	TS+Text	✗	51.25	22.16	30.94	11.39	80.21	30.08	43.75	5.66
TS Interaction Model											
TIMEINTERACT	4B	Streaming TS+Text	✓	**69.01**	63.78	**66.29**	**41.75**	**92.42**	<u>76.17</u>	**83.51**	**37.40**

Effectiveness of Full-to-Plan Distillation. To isolate the contribution of Full-to-Plan distillation in Stage III, we compare TIMEINTERACT with a variant that sets $\lambda_d^{(3)} = 0$ while retaining the response-control and Full-View generation losses and keeping all other settings unchanged. We focus on multi-turn IQA and PIF, where prior responses support follow-up questions and persistent instruction tracking. As shown in Table 10, removing distillation loss reduces IQA by 14.37% (64.56 → 55.28) and PIF by 15.40% (43.70 → 36.97). Their weighted mean decreases by 14.78% (54.13 → 46.13). These results suggest that Full-to-Plan distillation in Stage III helps maintain interaction quality when previous responses are represented by compact Plan tokens.

Table 10: Effect of Full-to-Plan Distillation.

Model	IQA	PIF	Mean
w/o Stage-III distillation	55.28	36.97	46.13
TIMEINTERACT	**64.56**	**43.70**	**54.13**
Absolute decrease	9.28	6.73	8.00
Relative decrease (%)	**14.37**	**15.40**	**14.78**

E BROADER IMPACT

Toward interactive time-series intelligence. To the best of our knowledge, this work is the first to introduce the concept of *Time-Series Interaction*, extending time-series language models from offline question answering toward real-time interaction over continuously evolving streams. Rather than

treating time series as static inputs that are analyzed only after they are fully observed, our formulation emphasizes continuous perception, response timing, persistent instructions, proactive interaction, and adaptation to user feedback. We hope this perspective can stimulate further exploration of interactive intelligence for time-series data, including new model architectures, learning paradigms, benchmarks, and evaluation protocols for continuously evolving environments.

Potential for Edge Deployment. The streaming design of TIMEINTERACT shows strong potential for deployment on edge devices. Since TIMEINTERACT processes observations incrementally rather than repeatedly re-encoding the entire historical sequence, it is naturally suited to continuously generated sensor streams. With further advances in lightweight model design, model compression, and efficient inference, such systems could potentially operate on wearable devices, mobile platforms, industrial controllers, and embedded monitoring systems, enabling low-latency interaction.

Human-centered Time-Series Interaction. TIMEINTERACT places users at the center of time-series interaction by allowing them to continuously express their needs through queries, instructions, and feedback. Users can decide what information is relevant, how the analysis should be presented, and how the interaction behavior should change as their goals evolve. Rather than forcing users to adapt to a fixed analysis procedure, the system can follow and refine its behavior according to user intent throughout the stream. This provides a more flexible and natural way for people to interact with evolving time-series data and supports more effective human–AI collaboration.

F LIMITATIONS AND FUTURE WORK

Although STREAMTSI-34K provides large-scale time-series interaction data spanning both synthetic and real-world time series across multiple domains, the current interaction construction still relies primarily on synthetic scenarios. In practice, real-world settings involve much richer and more diverse interaction scenarios and user behaviors. Future work should therefore extend Time-Series Interaction to a broader range of real-world systems with native interactive capabilities, enabling the collection and study of more naturally occurring interaction scenarios and user behaviors. More broadly, recent work has begun to systematize streaming LLMs through the perspectives of data flow and dynamic interaction (Tong et al., 2026b). Extending these broader perspectives to the time-series setting may help establish more unified problem formulations, learning paradigms, benchmarks, and evaluation protocols for advancing Time-Series Interaction.

G CASE STUDIES

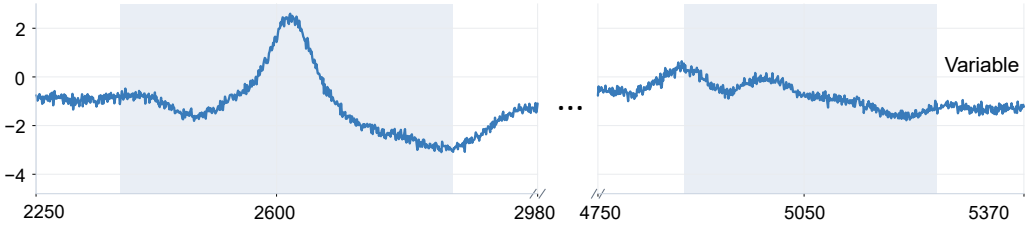
We provide six examples of streaming time-series interaction, comparing the response decisions and generated answers of TIMEINTERACT with representative baselines. Each case includes time-series observations, user queries or instructions when present, and reference responses to contextualize the model outputs. Figures 8–11 illustrate IQA, PIF, PTW, and UGA, respectively, covering query answering, instruction persistence, proactive warnings, and adaptation to user guidance. Figures 12 and 13 present two multi-turn scenarios combining PTW with PIF and IQA with PTW, respectively.



User Query: What physiological pattern is represented by these highlighted signals?
GT: Repeated large transient deflections in C3-M2 and E1-M2, with transient ECG and respiratory changes, are most **consistent with an arousal, such as a periodic-limb-movement-associated arousal.**
Expected Action: Respond

Model	Action	Response
TimeInteract	Respond	The sustained changes in C3-M2 and E1-M2, with transient ECG variation and comparatively continuous respiratory signals, are **consistent with an arousal-associated physiological change.**
Qwen2.5-7B	Respond	… Overall, these patterns suggest a response to a stimulus or an event that causes a **decrease in respiratory and cardiac activity while increasing brain activity.**
ChatTS-14B	Failed	The observed pattern is characterized by a sudden increase in EOG signal between E1 and M2, with an amplitude of 30.31, … **Although no new user request is present**, it is advisable to inform the user about this event due to its notable magnitude.

Figure 8: An example of IQA. TIMEINTERACT interprets the highlighted physiological signals as arousal-related changes, consistent with the reference response.



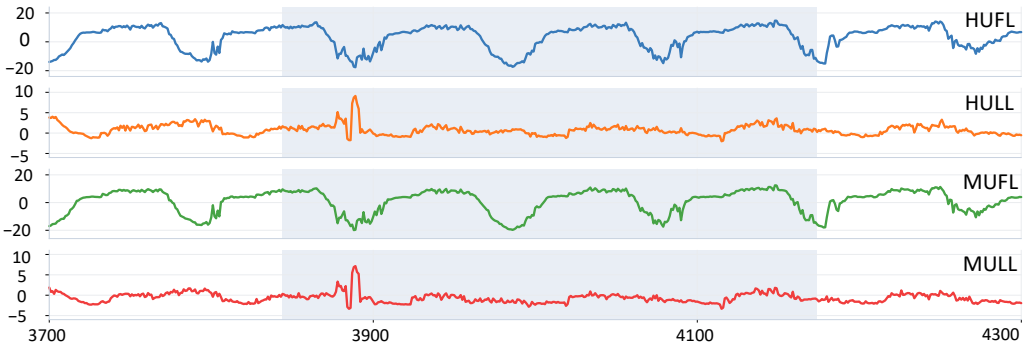
User Instruction: Please watch this time series and report once every 2618 points.

GT: [1] **This is Point 2618: first scheduled report.**
[2] **This is Point 5236: second scheduled report.**

Expected Action: Respond

Model	[1] Response	[2] Response
TimeInteract	Action: Respond **This is Point 2618: first scheduled report.**	Action: Respond **This is Point 5236: second scheduled report.**
Qwen2.5-7B	Action: Silence ***[No response – empty output]***	Action: Silence ***[No response – empty output]***
TimeOmni-1	Action: Silence ***[No response – empty output]***	Action: Silence ***[No response – empty output]***

Figure 9: An example of PIF. TIMEINTERACT follows the user's periodic-reporting instruction and responds at both scheduled points, while the baselines remain silent.



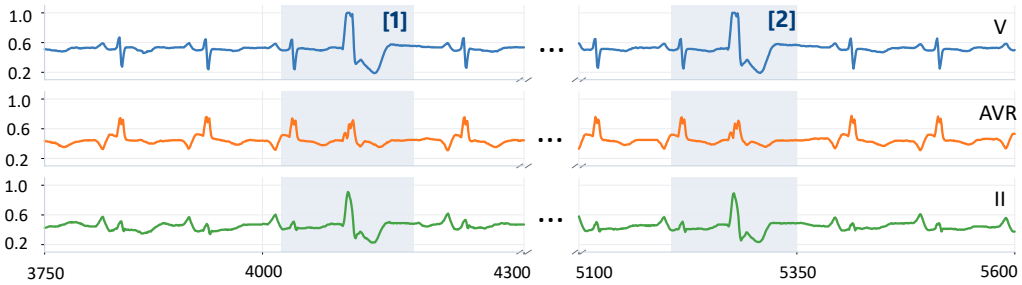
User Query: [None].

GT: Warning: From points 3884 to 3889, HULL and MULL swung from -1.407 and -2.878 to 9.109 and 7.143.

Expected Action: Respond

Model	Action	Response
TimeInteract	Respond	**Warning:** Variable 2 is undergoing **an upward spike.**
Mistral-7B-Instruct-v0.3	Respond	**No significant anomalies or abrupt changes detected** in the time-series data at this moment. Monitoring continues.
TimeOmni-1-7B	Silence	***[No response – empty output]***

Figure 10: An example of PTW. TIMEINTERACT warns of an upward spike without an explicit user query, while the baselines either report no anomaly or remain silent.



User Instruction: Use the multi-channel excursion as my reference pattern and let me know when a similar pattern appears later.

GT: Understood. I will use the pattern as the reference and watch for later similar excursions.

Expected Action: Respond

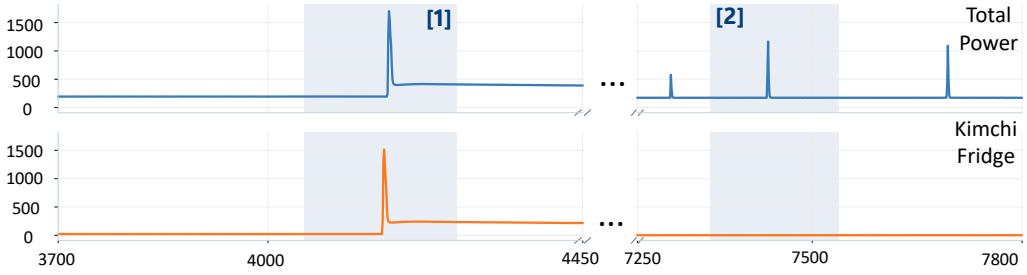
Question: [None]

GT: A similar multi-channel excursion appeared again: V rose to 1.001, AVR to 0.709, and II to 0.890 before the readings declined.

Expected Action: Respond

Model	[1] Response	[2] Response
TimeInteract	Action: Respond Understood. I will **use this pattern as the reference** and **notify you at its later occurrence.**	Action: Respond The same multi-channel excursion as the reference pattern **has appeared again.**
Mistral-7B-Instruct-v0.3	Action: Respond I will monitor the time-series data and **notify you when a similar pattern** to the reference pattern appears later in the data.	Action: Silence ***[No response – empty output]***
ChatTS-14B	Action: Failed *Raw_output:* A similar pattern occurred between points 208–222, where Variable 1 showed a slow rise followed by a rapid decline, similar to the reference pattern.	Action: Failed *Raw_output:* Variable 2's final value is 0.43, which is lower than its starting value of 0.34. This decrease is due to the overall trend being steady …

Figure 11: An example of UGA. TIMEINTERACT adopts the demonstrated multichannel pattern as a reference and reports its later recurrence.



User Query: [None].

GT: Warning: Whole-house and kimchi-fridge power underwent an **extreme abrupt surge.** Total power reached about 1700 W and kimchi-fridge power exceeded 1500 W before both declined.

Expected action: Respond

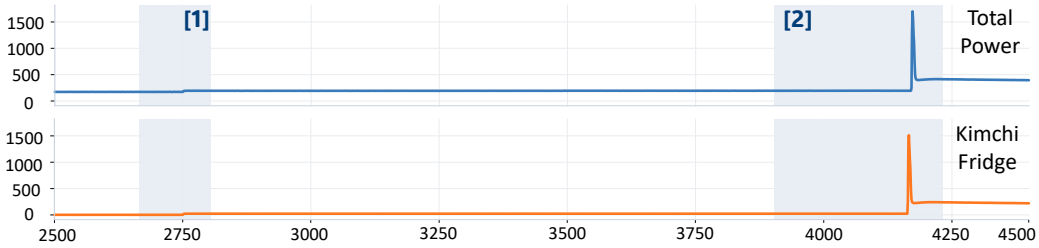
User Instruction: Please notify me whenever whole-house power shows a brief, sharp spike from its nearby baseline.

GT: Notification: Another brief whole-house power **spike occurred, reaching about 1166 W before returning toward baseline.**

Expected action: Respond

Model	[1] Proactive Temporal Warning	[2] Persistent Instruction Following
TimeInteract	Action: Respond Warning: Whole-house power **rose from about 197 W to above 1,700 W** while kimchi-refrigerator power rose from about 23 W to **above 1,500 W** in the **same transient interval.**	Action: Respond Notification: Whole-house power **briefly rose from about 174 W to above 1160 W** and returned near baseline.
Qwen2.5-7B	Action: Respond There is a significant anomaly in Variable 1, where the values suddenly **drop to much lower levels** (e.g., 249.82, 1305.86, 1701.33, etc.) ...	Action: Silence ***[No response – empty output]***
Qwen3-14B	Action: Silence ***[No response – empty output]***	Action: Silence ***[No response – empty output]***

Figure 12: An example of multi-turn interaction combining PTW and PIF. TIMEINTERACT first warns of a power surge, then follows a user instruction to report a later power spike.



Question: When did the sustained higher-power interval begin?

GT: The sustained higher-power interval **began around timestep 2751**, when kimchi-fridge power rose from about 1.6 W toward 23 W and total power **increased into the mid-190 W range.**

Expected action: Respond

User Query: [None]

GT: Warning: Whole-house and kimchi-fridge power underwent an extreme abrupt surge. Total power reached about 1700 W and kimchi-fridge power exceeded 1500 W before both declined.

Expected action: Respond

Model	[1] Instant Query Answering	[2] Proactive Temporal Warning
TimeInteract	Action: Respond Around timestep 2800, total power rose from **about 174 W to roughly 196 W** and remained near that higher level.	Action: Respond Warning: Whole-house power rose from about **197 W to above 1700 W** while kimchi-refrigerator power rose from about **23 W to above 1500 W** in the same transient interval.
Qwen2.5-7B	Action: Respond The sustained higher-power interval **began around the 13th observation**. This interval lasted until the 25th observation.	Action: Silence ***[No response – empty output]***
Mistral-7B-Instruct-v0.3	Action: Respond The sustained higher-power interval appears to have started around the **171st observation**, where Variable 1 value was 172.62.	Action: Silence ***[No response – empty output]***

Figure 13: An example of multi-turn interaction combining IQA and PTW. TIMEINTERACT estimates the onset of a higher-power interval and subsequently warns of a surge without a new query.