PIFBENCH: EVALUATING AND IMPROVING PROACTIVE INTERACTION IN STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI systems move toward real-time human-AI interaction, evaluating models' ability to understand continuously arriving visual and audio information becomes increasingly important. In this context, streaming video understanding requires models to process incoming content incrementally and decide whether and when to respond. The ability to make this decision without an explicit user query is known as proactive interaction. Existing benchmarks for proactive interaction often omit audio or rely on laborious open-ended evaluation, and effective methods for improving this capability in omni-modal models remain underexplored. To address these gaps, we study proactive interaction from a full-stack perspective, covering evaluation, data construction, and model adaptation. For evaluation, we formalize **Proactive Instruction Following (PIF)**, a task defined by providing an instruction before a video begins and requiring the model to process the video incrementally, remain silent when the specified condition is unmet, and produce the required response when it is met. Based on this task, we introduce **PIFBench**, a comprehensive benchmark for evaluating proactive interaction in streaming video understanding. PIFBench contains 1,993 items, covers both visual and audio information, and supports stable rule-based verification. Our evaluation shows that mainstream models have limited proactive interaction capability. Thus, we build **Proactive-Instruct-8K**, a structured training dataset for omni-modal models. For model adaptation, we make reusable modifications to the training framework and fine-tune Qwen3-Omni-30B-A3B-Instruct on Proactive-Instruct-8K, yielding **Qwen3-Omni-Proactive**, which improves accuracy on PIFBench from 13.79 to 19.42 under the same online evaluation setting.

1 INTRODUCTION

Multimodal Large Language Models (**MLLMs**) have shown strong performance on offline video understanding tasks (Zhao et al., 2025a; Liu et al., 2026; Fu et al., 2025a). However, real-time applications, such as autonomous driving (Gao et al., 2026), intelligent surveillance (Yao et al., 2026), and assistive technologies (Zhang et al., 2025c), require models to understand continuously arriving visual and audio information. This need has shifted attention from offline video understanding to streaming video understanding (Lin et al., 2026; Niu et al., 2025). To enable streaming video understanding, models need two capabilities: long-context management and proactive interaction. The former allows a model to continuously receive inputs without running out of memory (Qian et al., 2024), while the latter requires the model to decide whether to stay silent or respond based on the current video stream (Qian et al., 2025). Although existing work has studied both capabilities (Di et al., 2025; Chen et al., 2024; Wang et al., 2025c), proactive interaction has received less systematic attention. Existing benchmarks often omit audio (Shi et al., 2026) or rely on laborious open-ended evaluation (Wang et al., 2025b;d), and effective methods for improving this capability in omni-modal models remain limited. These gaps motivate us to *study proactive interaction in omni-modal models as a full-stack problem, covering evaluation, data construction, and model adaptation.*

In this work, we formalize **Proactive Instruction Following (PIF)**, a task defined by providing an instruction before a video begins and requiring the model to process the video incrementally, remain silent when the specified condition is unmet, and produce the required response when it

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.

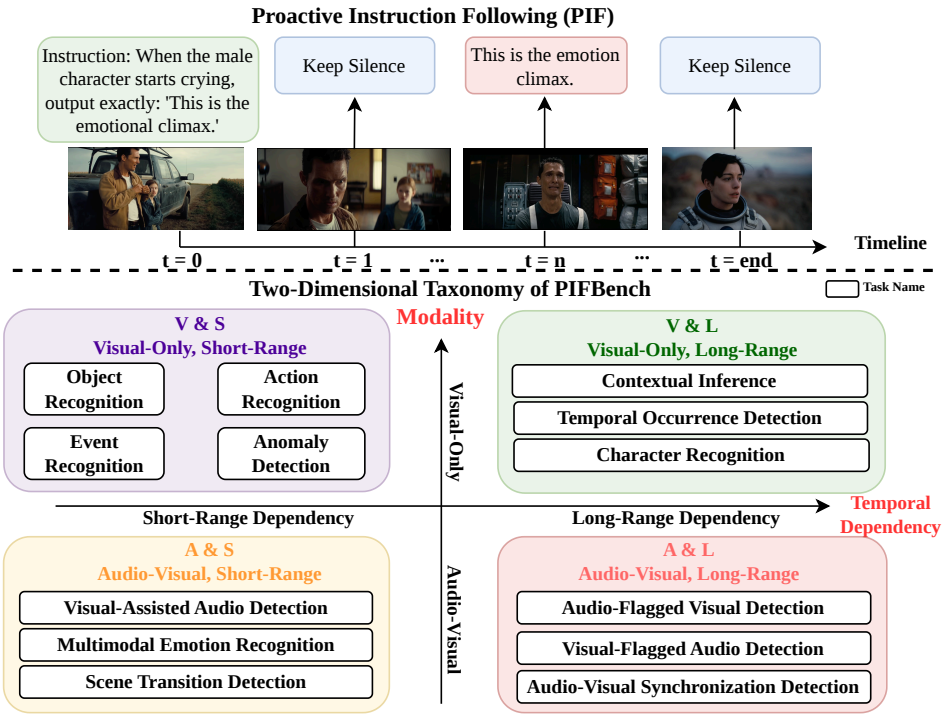

Figure 1: (*Top*): An example of Proactive Instruction Following (PIF), where the model processes the video incrementally, remains silent while the specified condition is unmet, and produces the required response when it is met. (*Bottom*): Taxonomy of PIFBench tasks by temporal dependency and modality.

is met. Examples are shown in Figure 1. PIF formulates proactive interaction as an instruction-following problem that supports rule-based verification and naturally prompts the model to respond at the right moment. Building on this task, we propose Proactive Instruction Following Benchmark (**PIFBench**), a comprehensive benchmark designed to evaluate proactive interaction in streaming video understanding. PIFBench contains 1,993 items with video durations ranging from under 30 seconds to 10 minutes. We construct benchmark samples using an automated annotation pipeline. To better reflect real-world video scenarios, PIFBench covers both visual and audio information while supporting stable rule-based verification. All benchmark samples are further reviewed by human annotators to ensure annotation quality.

We evaluate mainstream MLLMs (Xu et al., 2025; Google DeepMind, 2025a) on PIFBench and find that current models have limited proactive interaction capability. To address this limitation, we construct **Proactive-Instruct-8K**, a structured training dataset for omni-modal models. We select Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025) as the base model because it is open-source and representative of omni-modal models. We then make reusable modifications to the training framework and fine-tune the model on Proactive-Instruct-8K, yielding **Qwen3-Omni-Proactive**, which improves accuracy on PIFBench from 13.79 to 19.42 under the same online evaluation setting.

In summary, our contributions are:

1. **PIFBench**: We introduce a multimodal and rule-verifiable benchmark for proactive interaction in streaming video understanding.
2. **Proactive-Instruct-8K**: We release a structured training dataset for improving proactive interaction in omni-modal models.
3. **Qwen3-Omni-Proactive**: We develop an adapted Qwen3-Omni model with reusable training-framework modifications, demonstrating the effectiveness of task-aligned training for proactive interaction.

2 Related Work

Video Understanding Benchmarks. Video understanding benchmarks have evolved from vision-centric evaluation toward richer multimodal and long-form settings. Early benchmarks typically evaluated general video understanding and often omitted audio (Cao et al., 2026; Fang et al., 2024). More recent benchmarks have incorporated both visual and audio information, such as Video-MME (Fu et al., 2025a) and OmniBench (Li et al., 2026b), or emphasized long-video understanding, such as LVBench (Wang et al., 2025a) and LongVideoBench (Wu et al., 2024). Our work is most related to streaming video benchmarks, where the model receives video over time instead of seeing the full video at once. Existing streaming video benchmarks can be grouped into two types. The first type focuses on real-time video understanding, where models process incoming video content and answer questions asked at specified times. Representative benchmarks include StreamingBench (Lin et al., 2026), OVO-Bench (Niu et al., 2025), LiveSports-3K (Chen et al., 2025), SVBench (Yang et al., 2025), ODV-Bench (Zeng et al., 2026), OVBench (Huang et al., 2025), and RTV-Bench (Xun et al., 2026). The second type focuses on proactive interaction, where models decide whether to remain silent or respond without an explicit user query. Representative benchmarks include OmniMMI (Wang et al., 2025d), ProactiveVideoQA (Wang et al., 2025b), RIVER Bench (Shi et al., 2026), PhoStream (Lu et al., 2026), and SPOT-Bench (Chatterjee et al., 2026). PIFBench belongs to the second type and combines visual and audio coverage with rule-based verification. A comparison is shown in Table 1.

Table 1: Comparison of PIFBench with previous benchmarks. A green dot indicates that proactive interaction is evaluated in only part of the benchmark.

Benchmarks	Multimodal (Audio + Visual)	Automatic Annotation	Accompanying Training Set	Streaming Scenarios	Proactive Interaction	Rule-based Verifiable
MMVU (Zhao et al., 2025a)	✗	✗	✗	✗	✗	✗
Video-MME (Fu et al., 2025a)	✓	✗	✗	✗	✗	✓
OmniVideoBench (Li et al., 2026a)	✓	✗	✗	✗	✗	✓
StreamingBench (Lin et al., 2026)	✓	✓	✗	✓	●	✓
OVO-Bench (Niu et al., 2025)	✗	✗	✗	✓	●	✗
ViSpeak-Bench (Fu et al., 2025b)	✓	✗	✓	✓	✗	✗
LiveSports-3K (Chen et al., 2025)	✗	✓	✓	✓	✗	✗
OVBench (Huang et al., 2025)	✗	✓	✓	✓	●	✓
ODV-Bench (Zeng et al., 2026)	✗	✓	✓	✓	●	✗
OmniMMI (Wang et al., 2025d)	✓	✗	✓	✓	✓	✗
ProactiveVideoQA (Wang et al., 2025b)	✓	✓	✗	✓	✓	✗
RIVER Bench (Shi et al., 2026)	✗	✓	✗	✓	●	✗
PIFBench (Ours)	✓	✓	✓	✓	✓	✓

Optimization Methods for Streaming Scenarios. Optimization methods for streaming video understanding follow two main directions. The first direction improves long-context management, allowing models to process longer video streams with lower memory usage and computational cost. For example, Flash-VStream (Zhang et al., 2025b) and VideoStreaming (Qian et al., 2024) use memory mechanisms to retain and select information from earlier frames, whereas SimpleStream (Shen et al., 2026) uses a sliding window containing only the most recent frames. ReKV (Di et al., 2025) and HERMES (Zhang et al., 2026) further optimize the KV cache for efficient streaming video understanding, while StreamingTOM (Chen et al., 2026b) and STC (Wang et al., 2026a) reduce redundant visual tokens through token compression. The second direction improves proactive interaction, so that models can decide whether to stay silent or respond. Methods for response timing include EOS prediction in VideoLLM-Online (Chen et al., 2024), binary classification in Dispider (Qian et al., 2025), text generation with "NO REPLY" in MMDuet2 (Wang et al., 2026b), and matching video frames with visual proposals in Em-Garde (Zheng et al., 2026). Some works apply agent frameworks to streaming video understanding (Zhang et al., 2025a; Long et al., 2026; Chen et al., 2026a). Different from these methods, our work focuses on data construction and model adaptation for omni-modal proactive interaction, and uses a special token to teach the model when to stay silent.

3 PIFBench

3.1 Proactive Instruction Following

We introduce **Proactive Instruction Following (PIF)**, a task defined by providing an instruction before a video begins and requiring the model to process the video incrementally, remain silent

when the specified condition is unmet, and produce the required response when it is met. This task evaluates three core capabilities. (1) **Multimodal Perception** requires the model to understand relevant visual and audio information over time. (2) **Instruction Following** requires the model to identify the condition and follow the response requirements specified by the instruction. (3) **Proactive Interaction** requires the model to decide whether and when to respond without relying on an explicit user query or a voice activity trigger (Graf et al., 2015). Based on PIF, we build Proactive Instruction Following Benchmark (**PIFBench**).

3.2 Taxonomy

We organize PIFBench along two key dimensions: **Modality** and **Temporal Dependency**, as shown in Figure 1. Modality distinguishes visual-only tasks from audio-visual tasks. The latter evaluate the model's ability to integrate information across modalities. Temporal Dependency distinguishes tasks that rely on short-range context from those that require long-range context. Short-range tasks can be completed using the current or nearby video segments, whereas long-range tasks require information about earlier events, comparisons across time, or evidence accumulated over the video. Combining these dimensions yields four categories: **V&L**, **V&S**, **A&L**, and **A&S**. Here, V and A denote visual-only and audio-visual tasks, while L and S denote long-range and short-range dependencies, respectively. Each category contains three or four tasks. Appendix A provides detailed definitions for all tasks.

3.3 Data Construction

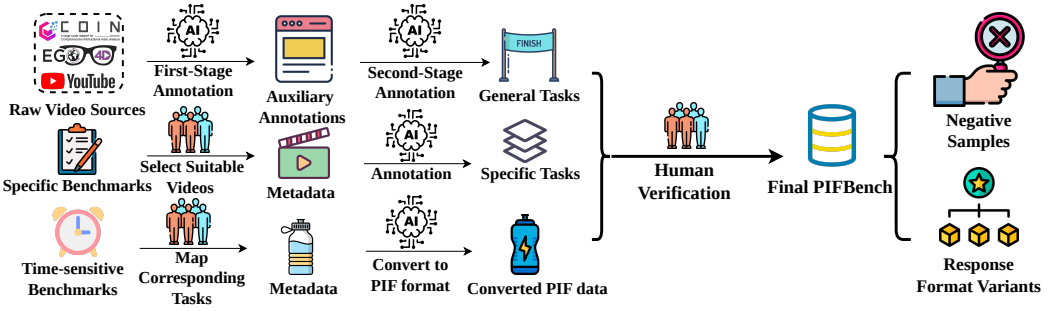

Figure 2: Overview of the PIFBench data construction pipeline. The upper part shows the construction of general-task samples from diverse video sources. The middle part shows the construction of specific-task samples using task-specific videos and annotations. The lower part shows the conversion of time-sensitive benchmarks into the PIF format. Human annotators review all candidate samples to form the final benchmark.

Data Sources. The videos in PIFBench come from two main sources: raw videos and existing time-sensitive video benchmarks. The raw videos are collected from public datasets, including COIN (Tang et al., 2019), EgoSchema (Mangalam et al., 2023), YouTube-8M (Abu-El-Haija et al., 2016), and ActivityNet (Caba Heilbron et al., 2015), as well as manually selected YouTube videos. These sources cover diverse everyday activities, instructional procedures, and web content. We also convert data from time-sensitive video benchmarks (Niu et al., 2025; Fu et al., 2025b) into the PIF format. Figure 2 summarizes the data construction pipeline.

We group tasks as general or specific according to their video requirements. General tasks can be constructed from a wide range of common videos, such as daily activities and instructional procedures. Specific tasks require videos with particular content, such as anomalous events, emotions, or audio-visual patterns. We use specialized datasets to obtain suitable videos for these tasks (Sultani et al., 2018; Pardo et al., 2022; Wu et al., 2025). We classify OR, AR, ER, TOD, CI, VAAD, AFVD, and VFAD as general tasks and the remaining tasks as specific tasks.

General Tasks. For general tasks, we convert raw videos into PIF examples, each containing an instruction, a trigger time range, and an expected response. The construction process has five steps. (1) We select a target task from the general-task set. (2) We select a raw video based on its

duration and audio availability. (3) We use a multimodal model to generate task-specific auxiliary annotations that support the generation of the final instruction and ground truth. (4) Given the video, auxiliary annotations, task definition, metadata, and in-context examples, the model generates the final instruction and ground truth. (5) Human annotators verify the generated data. Details are provided in Appendix B. Rather than generating instructions in a single step, our method first produces task-relevant auxiliary annotations and then uses them to generate the final instruction and ground truth. This decomposition helps verify timestamps and is similar to dense captioning (Zhou et al., 2024), where local annotations support global generation.

Specific Tasks. For specific tasks, we use task-specific construction strategies because they require videos with particular content that is uncommon in general video sources. For Anomaly Detection (**AD**), Scene Transition Detection (**STD**), and Multimodal Emotion Recognition (**MER**), we use UCF-Crime (Sultani et al., 2018), MovieCuts (Pardo et al., 2022), and eMotions (Wu et al., 2025), respectively. We convert their annotations into the PIF format and concatenate selected video segments to create additional examples with clear trigger conditions and response timestamps. For Character Recognition (**CR**), we manually select YouTube videos, use multimodal models to generate video-level annotations, and manually construct PIF examples. For Audio-Visual Synchronization Detection (**AVSD**), we create asynchronous clips by shifting audio or visual segments in time or swapping segments across clips. Human annotators review all generated samples to ensure annotation accuracy.

Additional Construction Components. We further construct response-format variants and negative samples, and convert existing time-sensitive video benchmarks into the PIF format. These components evaluate format following, the ability to remain silent when no response is needed, and proactive interaction on existing tasks. Detailed construction procedures are provided in Appendix C.

Dataset Statistics. PIFBench contains 1,993 samples across 13 tasks, covering both visual-only and audio-visual settings. Video durations range from under 30 seconds to 10 minutes. Detailed task and duration distributions are provided in Appendix D.

4 QWEN3-OMNI-PROACTIVE

To improve proactive interaction in omni-modal models, we study the problem from the perspectives of data construction and model adaptation. From the data perspective, we use the scalable annotation pipeline described in Section 3.3 to construct **Proactive-Instruct-8K** at a lower annotation cost. From the model perspective, we make reusable modifications to the training framework and adapt Qwen3-Omni-30B-A3B-Instruct into **Qwen3-Omni-Proactive**.

4.1 TRAINING DATA CONSTRUCTION

To construct training data at scale, we reuse the annotation pipeline described in Section 3.3. Because human verification is costly, we instead use multimodal models for both automatic annotation and quality filtering. The resulting dataset, **Proactive-Instruct-8K**, contains 8K synthetic samples.

4.2 MODEL ADAPTATION

To support proactive interaction in Qwen3-Omni-30B-A3B-Instruct, we add a special token, `<Silence>`, to its vocabulary. This token indicates that the model should remain silent at the current time step. The core training objective is to teach the model when to remain silent and when to respond. Specifically, we divide each video into short temporal chunks and associate each chunk with a timestamp, enabling the model to track its position in the stream. We then format each PIF sample as a multi-turn streaming conversation in which the model outputs `<Silence>` until the trigger condition is met and produces the target response when it is satisfied. Table 2 illustrates the streaming format on the left and summarizes the corresponding data-loading algorithm on the right.

A practical challenge is that this supervision is highly imbalanced: most streaming chunks correspond to `<Silence>`, while only a small number require an actual response. Without additional

Table 2: CoT-augmented streaming format and timestamp-aware video-loading algorithm used for Qwen3-Omni-Proactive training.

CoT-augmented streaming format

```
user: instruction
user: <video> <0--1s>
assistant: <cot> reason
</cot> <Silence>
user: <video> <1--2s>
assistant: <cot> reason
</cot> <Silence>
...
user: <video> <12--13s>
assistant: <cot> reason
</cot> react content
```

Algorithm: Timestamp-aware video loading

Input: source video V; segment plan $P = \{\tau_i\}_{i=1}^{K}$, where K is the number of chunks and $\tau_i = [t_i^0, t_i^1)$ is the time window of chunk i.
Output: cached chunks $\mathcal{C} = \{(X_i, A_i, \tau_i)\}_{i=1}^{K}$, where X_i and A_i denote visual frames and audio.
1. Probe V to obtain metadata $m = (F, r, T)$, where F, r, and T denote total frames, FPS, and duration.
2. Compute frame budget $B = \mathrm{smart_nframes}(m)$ and allocate b_i frames to each chunk with $\sum_i b_i = B$.
3. For each τ_i, sample frames $X_i = \mathrm{resize}(\mathrm{decode}(V, \tau_i, b_i))$.
4. Clip audio $A_i = \mathrm{audio}(V, \tau_i)$ and cache (X_i, A_i, τ_i) before dataset loading.

treatment, the model can easily learn a biased default behavior of keeping silent. We explored several strategies to mitigate this issue. One option is to use focal loss (Lin et al., 2017) or reweighting to down-weight silence tokens and up-weight reaction tokens. However, in practice, this objective quickly reaches a very low loss and converges too easily, partly because the target reaction content is often already specified in the instruction, making the response-generation loss less informative. Another option is to add a classification head (Qian et al., 2025) to explicitly predict whether the model should react at each moment. Although this can be effective, it introduces extra trainable components and makes the method less off-the-shelf.

We therefore adopt a chain-of-thought (CoT) augmented training format before the final decision. Specifically, before outputting either `<Silence>` or the reaction content, the assistant first generates a short reasoning trace that summarizes the previous context, analyzes the current video chunk, and determines whether the instruction condition has been satisfied. We use Qwen3-Omni-30B-A3B-Instruct as the annotation model to generate these CoT annotations. This design makes the supervision signal denser: instead of concentrating the loss on one or two decision tokens, the learning signal is distributed across multiple reasoning tokens. As a result, the training data become more learnable, the loss better reflects useful temporal and instruction-following behavior, and the extreme imbalance between silence and reaction labels is alleviated.

We implement model training with ms-swift (Zhao et al., 2025b). The default video-loading workflow expects videos to be pre-segmented before training, so changing the chunk duration would require repeated preprocessing and additional storage. To make training more flexible and efficient, we redesign the loading pipeline as summarized on the right side of Table 2: given a timestamp segment plan, the loader starts from the original video, allocates the frame budget across segments, and generates and caches timestamped video/audio chunks before dataset loading. As a result, the same source video can support different chunking configurations without repeatedly decoding and preprocessing the video during training.

5 Experiments

5.1 Experimental Setup

Models We evaluate representative models with different modality support and proactive interaction capabilities. We divide them into three groups. The first group consists of omni-modal models without native proactive interaction capability: Phi-4-Multimodal (Abouelenin et al., 2025), Gemini-2.5-Flash (Comanici et al., 2025), Gemini-3-Flash (Google DeepMind, 2025b) and Gemini-3-Pro (Google DeepMind, 2025a), Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025), and Qwen3.5-Omni-Plus (Team, 2026). We evaluate these models using the offline protocol, which provides the complete video and asks the model to predict the trigger timestamp and required reaction content. The second group contains representative open-weight models with proactive interaction capability: VideoLLM-Online (Chen et al., 2024), MMDuet2 (Wang et al., 2026b), Dispider (Qian et al., 2025), and Em-Garde (Zheng et al., 2026). Because these models process only visual inputs, we evaluate them using the online protocol on the visual-only tasks in PIFBench. The third group con-

sists of omni-modal models with proactive interaction capability, including MiniCPM-o 4.5 (Cui et al., 2026) and our adapted Qwen3-Omni-Proactive. We evaluate both models using the online protocol on the full PIFBench benchmark.

Evaluation Protocols We adopt two evaluation protocols according to whether a model supports proactive interaction. The online protocol feeds video frames to the model one at a time, with the temporal interval between consecutive frames adapted to each experimental setting. At each step, the model decides whether to respond or remain silent. The offline protocol provides the entire video and asks the model to identify the timestamp at which it should respond, similar to a temporal grounding task. We use the online protocol for models with proactive interaction capability because they provide dedicated signals for remaining silent and responding and are explicitly optimized for proactive responses. For models without this capability, we use the offline protocol instead. We examine the effect of this protocol choice in detail in Section 5.3.1. For generation, we use each model's default decoding settings. If a video exceeds the model's default input limits, we reduce its spatial resolution or frame sampling rate as needed.

Metrics We evaluate models using four metrics. **Total Accuracy (TotalAcc)** measures overall correctness across all samples by jointly considering the reaction decision, response content, and response timing, making it the most comprehensive and intuitive metric. **Reaction Accuracy (ReactAcc)** measures whether the model correctly decides whether a response is required or it should remain silent. **Response Accuracy (RespAcc)** measures how often the model provides the correct content when it chooses to react, reflecting its textual instruction-following and comprehension abilities. **Weighted Time MAE (WT-MAE)** measures how far a reaction falls outside its valid time window, with a larger penalty for early reactions than for late ones. Overall, TotalAcc provides a comprehensive measure of model performance, while the other three metrics help diagnose specific failure modes: whether the model correctly decides to react, whether its response content follows the instruction, and how far its response timestamp deviates from the valid time window. Detailed calculations are provided in Appendix G.

5.2 Main Experiments

Table 3: Main results on PIFBench. The visual-only proactive models are evaluated only on visual-only tasks because they do not support audio inputs.

Model	TotalAcc (↑)	ReactAcc (↑)	RespAcc (↑)	WT-MAE (↓)
Omni-modal models without proactive interaction capability (offline protocol, full PIFBench)				
Phi-4-Multimodal (Abouelenin et al., 2025)	3.80	40.34	29.28	70.51
Gemini-2.5-Flash (Comanici et al., 2025)	44.31	87.18	83.06	18.24
Gemini-3-Flash (Google DeepMind, 2025b)	53.08	89.32	91.65	13.59
Gemini-3-Pro (Google DeepMind, 2025a)	**55.24**	**90.32**	**92.82**	12.43
Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025)	11.31	81.41	40.84	33.70
Qwen3.5-Omni-Plus (Team, 2026)	28.02	75.52	53.19	**9.54**
Visual-only models with proactive interaction capability (online protocol, visual-only tasks)				
VideoLLM-Online (Chen et al., 2024)	4.52	8.19	0	-
MMDuet2 (Wang et al., 2026b)	5.39	30.35	7.61	88.48
Dispider (Qian et al., 2025)	4.76	7.85	10.00	**36.00**
Em-Garde (Zheng et al., 2026)	**8.34**	**90.71**	**46.65**	128.11
Omni-modal models with proactive interaction capability (online protocol, full PIFBench)				
MiniCPM-o 4.5 (Cui et al., 2026)	7.41	48.35	25.16	74.41
Qwen3-Omni-Proactive (Ours)	**19.42**	**64.94**	**57.99**	**56.42**

Overall Results. Table 3 shows that current models remain limited on PIFBench. Under the offline protocol, Gemini-3-Pro achieves the highest TotalAcc, ReactAcc, and RespAcc, while Qwen3.5-Omni-Plus obtains the lowest WT-MAE. The gap between these metrics indicates that accurate timing alone does not guarantee correct trigger detection and response content. Among the visual-only proactive models, Em-Garde achieves the best TotalAcc, ReactAcc, and RespAcc. However, its high WT-MAE suggests that producing responses at the correct time remains challenging.

Under the online protocol on the full benchmark, Qwen3-Omni-Proactive consistently outperforms MiniCPM-o 4.5. It improves TotalAcc from 7.41 to 19.42, ReactAcc from 48.35 to 64.94, and RespAcc from 25.16 to 57.99, while reducing WT-MAE from 74.41 to 56.42. These improvements demonstrate the effectiveness of task-aligned data and model adaptation for proactive interaction. Nevertheless, the overall results show that models still struggle to identify trigger conditions and produce the required content at the correct time. Because the three groups use different protocols and task sets, their scores should not be compared directly. Detailed analyses are provided in Appendix E.

5.3 Ablation Study

5.3.1 Evaluation Protocol

Although the online protocol is designed for models with native proactive interaction, it can also be adapted to models without this capability. We compare the online and offline protocols on Qwen3-Omni-30B-A3B-Instruct and Gemini-2.5-Flash, with the results shown in Table 4. For computational efficiency, we conduct this comparison only on the subset of PIFBench constructed through our annotation pipeline. Overall, the offline protocol yields stronger results for two main reasons. First, these models are not explicitly trained for proactive interaction, and remaining silent is not a behavior commonly required during their standard training and inference. Second, the offline protocol provides access to the complete video and more closely matches the conventional inference setting of these models. As the underlying video-understanding capability improves, offline performance can thus be viewed as an approximate upper bound on what the same model may achieve under the more demanding online protocol.

Table 4: Comparison of offline and online evaluation protocols.

	Qwen3-Omni-30B-A3B-Instruct		**Gemini-2.5-Flash**	
Metric	**Offline**	**Online**	**Offline**	**Online**
TotalAcc ↑	10.20	7.38	43.72	17.92
ReactAcc ↑	78.51	86.92	86.53	89.86
RespAcc ↑	40.66	38.73	79.41	74.77
WT-MAE ↓	31.89	91.70	18.49	91.47

Compared with the offline protocol, the online protocol has two practical limitations. First, it is substantially more expensive: for a 60-second video evaluated once per second, the online protocol processes approximately $30.5\times$ as much video input as a single offline evaluation, even before accounting for repeated inference overhead. Second, applying the online protocol to a model without native proactive interaction requires reformulating the question and defining an artificial output convention for indicating that no response is needed, such as producing "NO" or a designated special token. Such conventions are prompt-dependent and may not generalize across models. The offline protocol also has a limitation: access to the complete video may reveal information that would not yet be available in a true streaming setting. Nevertheless, it matches the standard inference setting of these models while avoiding the additional computational cost and model-specific output conventions of online evaluation. We therefore use the offline protocol to evaluate models without proactive interaction capability.

Notably, TotalAcc remains low even under the offline protocol, indicating limitations in the models' underlying video understanding and temporal localization capabilities. This result also shows that, although the questions have a simple format and are easy for humans to answer, they remain challenging for current models.

5.3.2 Original vs. PIF Format

As described in Section 3.3, we convert samples from several time-sensitive benchmarks into the PIF format. Their original tasks primarily evaluate real-time video understanding by asking a model to answer a multiple-choice question at a specified timestamp. We evaluate these questions in their original format using each dataset's default settings and then evaluate the same underlying questions in the PIF format. This controlled comparison isolates the ef-

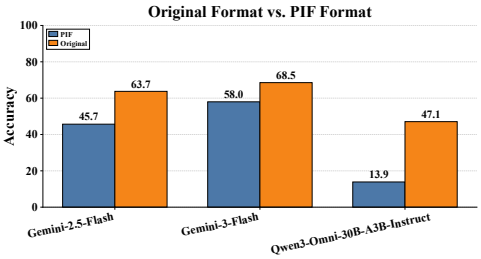

Figure 3: Comparison of accuracy under the original and PIF formats on converted time-sensitive benchmark samples.

fect of requiring a model to determine when to respond. We conduct the comparison using Qwen3-Omni-30B-A3B-Instruct, Gemini-2.5-Flash, and Gemini-3-Flash. As shown in Figure 3, all three models perform substantially worse in the PIF format. The performance drop is largest for Qwen3-Omni and becomes smaller for the stronger Gemini models, although a clear gap remains across all three models. These results show that even when the underlying questions remain unchanged, requiring models to decide when to respond introduces considerable additional difficulty.

5.3.3 Training Ablation

Table 5: Ablation of training strategies for Qwen3-Omni-Proactive. Scores are evaluated on the complete PIFBench.

Setting	SFT	CoT	Focal Loss	TotalAcc (↑)	ReactAcc (↑)	RespAcc (↑)	WT-MAE (↓)
Baseline	✗	✗	✗	13.79	**88.38**	49.6	91.62
PIF data w/o CoT	✓	✗	✗	18.73	50.45	**65.09**	62.14
PIF data w/o CoT + focal loss	✓	✗	✓	10.55	73.71	55.37	107.42
PIF data w/ CoT	✓	✓	✗	**19.42**	64.94	57.99	**56.42**

We compare four training settings to isolate the effects of PIF supervision, focal loss, and CoT augmentation. The *Baseline* is the original Qwen3-Omni-30B-A3B-Instruct without proactive-interaction adaptation. *PIF data w/o CoT* fine-tunes the model on the full Proactive-Instruct-8K dataset using only the direct silence or response targets. *PIF data w/o CoT + focal loss* additionally applies focal loss to mitigate the imbalance between silence and reaction targets. *PIF data w/ CoT* uses our CoT-augmented supervision on the full synthetic training set. All trained variants otherwise use the same base model and training configuration. To reduce computational and GPU memory costs, we perform online evaluation at 4-second intervals using a 17-step sliding context window. We primarily use Low-Rank Adaptation (LoRA) (Hu et al., 2021) for model training. The comparison is summarized in Table 5. Overall, SFT on Proactive-Instruct-8K improves TotalAcc over the baseline, confirming the effectiveness of task-specific supervision. Training without CoT achieves the highest RespAcc, whereas CoT augmentation yields the best TotalAcc and the lowest WT-MAE, suggesting that explicit reasoning supervision improves the model's temporal alignment and overall proactive interaction capability. Although focal loss partially recovers ReactAcc, it reduces TotalAcc and substantially increases WT-MAE, indicating that addressing the silence–reaction imbalance alone does not lead to better proactive instruction following. In practice, we recommend using the CoT-augmented variant because it encourages explicit reasoning over multimodal evidence and provides richer supervision than direct silence-or-response targets.

6 Conclusion

In this work, we study proactive interaction in streaming videos from evaluation, data construction, and model adaptation perspectives. We formalize Proactive Instruction Following and introduce PIFBench, a comprehensive, multimodal, and rule-verifiable benchmark covering diverse temporal dependencies and both visual and audio-visual information. Our evaluation shows that current MLLMs remain limited in deciding whether, when, and how to respond, and that converting conventional time-sensitive questions into the PIF format introduces substantial additional difficulty. To improve this capability, we construct Proactive-Instruct-8K and adapt Qwen3-Omni-30B-A3B-Instruct into Qwen3-Omni-Proactive, demonstrating the effectiveness of task-specific data and model adaptation. Overall, our results establish proactive instruction following as a distinct challenge that requires persistent instruction tracking, multimodal trigger detection, response control, and temporal grounding. Further improvements in streaming memory, temporal reasoning, silence–response calibration, and scalable data construction may enable more reliable proactive multimodal systems in real-world applications.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical

report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 961–970, 2015.

Meng Cao, Pengfei Hu, Yingyao Wang, Jihao Gu, Haoran Tang, Haoze Zhao, Chen Wang, Jiahua Dong, Wangbo Yu, Ge Zhang, et al. Video simpleqa: Towards factuality evaluation in large video language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2616–2624, 2026.

Dibyadip Chatterjee, Zhanzhong Pang, Fadime Sener, Yale Song, and Angela Yao. Don't pause! every prediction matters in a streaming video. *arXiv preprint arXiv:2604.24317*, 2026.

Jiawei Chen, Zhe Chen, Chaoqun Du, Maokui He, Wei He, Hengtao Li, Qizhen Li, Zide Liu, Hao Ma, Xuhao Pan, et al. Streamingclaw technical report. *arXiv preprint arXiv:2603.22120*, 2026a.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18407–18418. IEEE, 2024.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Live: Learning video llm with streaming speech transcription at scale. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29083–29095. IEEE, 2025.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24675–24685, 2026b.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 42115–42127, 2025.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *Advances in Neural Information Processing Systems*, 37:89098–89124, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025a.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21778–21788. IEEE, 2025b.

Yuan Gao, Mattia Piccinini, Yuchen Zhang, Dingrui Wang, Korbinian Moller, Roberto Brusnicki, Baha Zarrouki, Alessio Gambi, Jan Frederik Totz, Kai Storms, et al. Foundation models in autonomous driving: A survey on scenario generation and scenario analysis. *IEEE Open Journal of Intelligent Transportation Systems*, 2026.

Google DeepMind. Gemini 3 pro model card, 2025a. URL https://deepmind.google/models/model-cards/gemini-3-pro/. Model released November 2025; model card updated May 2026.

Google DeepMind. Gemini 3 Flash Model Card, December 2025b. URL https://deepmind.google/models/model-cards/gemini-3-flash/.

Simon Graf, Tobias Herbig, Markus Buck, and Gerhard Schmidt. Features for voice activity detection: a comparative analysis. *EURASIP Journal on Advances in Signal Processing*, 2015(1):91, 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: Ovbench and videochat-online. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338. IEEE, 2025.

Caorui Li, Yu Chen, Yiyan Ji, Jin Xu, Zhenyu Cui, Shihao Li, Yuanxing Zhang, Zhenghao Song, Dingling Zhang, Ying He, et al. Omnivideobench: Towards audio-visual understanding evaluation for omni mllms. In *International Conference on Learning Representations*, volume 2026, pp. 138214–138236, 2026a.

Yizhi Li, Ge Zhang, Yinghao Ma, Ruibin Yuan, Hangyu Guo, Yiming Liang, Jiaheng Liu, Noah Wang, Jian Yang, Siwei Wu, et al. Omnibench: Towards the future of universal omni-language models. *Advances in Neural Information Processing Systems*, 38, 2026b.

Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151. IEEE, 2026.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Yuanxin Liu, Kun Ouyang, Haoning Wu, Yi Liu, Lin Sui, Xinhao Li, Yan Zhong, Xinyu Zhou, and Xu Sun. Videoreasonbench: Can mllms perform vision-centric complex video reasoning? In *International Conference on Learning Representations*, volume 2026, pp. 3475–3502, 2026.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, volume 2026, pp. 146197–146246, 2026.

Xudong Lu, Huankang Guan, Yang Bo, Jinpeng Chen, Xintong Guo, Shuhan Li, Fang Liu, Peiwen Sun, Xueying Li, Wei Zhang, et al. Phostream: Benchmarking real-world streaming for omnimodal assistants in mobile scenarios. *arXiv preprint arXiv:2601.22575*, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913. IEEE, 2025.

Alejandro Pardo, Fabian Caba Heilbron, Juan León Alcázar, Ali Thabet, and Bernard Ghanem. Moviecuts: A new dataset and benchmark for cut type recognition. In *European Conference on Computer Vision*, pp. 668–685. Springer, 2022.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24045–24055. IEEE, 2025.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *arXiv preprint arXiv:2604.02317*, 2026.

Yansong Shi, Qingsong Zhao, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. River: A real-time interaction benchmark for video llms. *arXiv preprint arXiv:2603.03985*, 2026.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6479–6488. IEEE, 2018.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1207–1216. IEEE, 2019.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 2025a.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18523–18533, 2026a.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models. *arXiv preprint arXiv:2507.09313*, 2025b.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. In *EMNLP (Findings)*, pp. 6338–6359, 2025c.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Guanglu Wan, Huishuai Zhang, and Dongyan Zhao. Mmduet2: Enhancing proactive interaction of video mllms with multi-turn reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 23257–23270, 2026b.

Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. Omnimmi: A comprehensive multi-modal interaction benchmark in streaming video contexts. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18925–18935. IEEE, 2025d.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Xuecheng Wu, Dingkang Yang, Danlei Huang, Xinyi Yin, Yifan Wang, Jia Zhang, Jiayu Nie, Liangyu Fu, Yang Liu, Junxiao Xue, et al. emotions: A large-scale dataset and audio-visual fusion network for emotion analysis in short-form videos. *arXiv preprint arXiv:2508.06902*, 2025.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

Shuhang Xun, Sicheng Tao, Jungang Li, Yibo Shi, Zhixin Lin, Zhanhui Zhu, Yibo Yan, Hanqian Li, Linghao Zhang, Shikang Wang, et al. Rtv-bench: Benchmarking mllm continuous perception, understanding and reasoning through real-time video. *Advances in Neural Information Processing Systems*, 38, 2026.

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. Svbench: A benchmark with temporal multi-turn dialogues for streaming video understanding. In *International Conference on Learning Representations*, volume 2025, pp. 42522–42556, 2025.

Shanle Yao, Armin Danesh Pazho, Narges Rashvand, and Hamed Tabkhi. Are multimodal llms ready for surveillance? a reality check on zero-shot anomaly detection in the wild. *arXiv preprint arXiv:2603.04727*, 2026.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. *Advances in Neural Information Processing Systems*, 38:75804–75835, 2026.

Gengyuan Zhang, Tanveer Hannan, Hermine Kleiner, Beste Aydemir, Xinyu Xie, Jian Lan, Thomas Seidl, Volker Tresp, and Jindong Gu. Avila: Asynchronous vision-language agent for streaming multimodal data interaction. *arXiv preprint arXiv:2506.18472*, 2025a.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21059–21069. IEEE, 2025b.

Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430, 2026.

Ziyi Zhang, Zhen Sun, Zongmin Zhang, Zifan Peng, Yuemeng Zhao, Zichun Wang, Zeren Luo, Ruiting Zuo, and Xinlei He. " i can see forever!": Evaluating real-time videollms for assisting individuals with visual impairments. *arXiv preprint arXiv:2505.04488*, 2025c.

Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8475–8489. IEEE, 2025a.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29733–29735, 2025b.

Yikai Zheng, Xin Ding, Yifan Yang, Shiqi Jiang, Hao Wu, Qianxi Zhang, Weijun Wang, Ting Cao, and Yunxin Liu. Em-garde: A propose-match framework for proactive streaming video understanding. *arXiv preprint arXiv:2603.19054*, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18243–18252. IEEE, 2024.

A PIFBENCH TASK DEFINITIONS

Table 6: Definitions of the 13 tasks in PIFBench.

Category	Task	Definition
V&S	Object Recognition (**OR**)	Identify an object that appears in the video.
V&S	Action Recognition (**AR**)	Identify a specific action that has a short duration.
V&S	Event Recognition (**ER**)	Understand and locate the event mentioned in the instruction.
V&S	Anomaly Detection (**AD**)	React when an anomalous event occurs.
V&L	Temporal Occurrence Detection (**TOD**)	A specific object, action, or person appears multiple times in the video; react when it appears for the n-th time.
V&L	Character Recognition (**CR**)	The video has several characters that show up one after another, and the model should react whenever a new character appears.
V&L	Contextual Inference (**CI**)	Reason based on the context and react when the instruction is met.
A&S	Visual-Assisted Audio Detection (**VAAD**)	React when a specific audio event is detected using both visual information and audio signals.
A&S	Multimodal Emotion Recognition (**MER**)	React to a specific emotion when detected, using the tone of speech and visual information for determination.
A&S	Scene Transition Detection (**STD**)	Detect transitions between scenes with the help of audio-visual signals.
A&L	Audio-Flagged Visual Detection (**AFVD**)	A specific audio event in the video serves as a flag, and the model reacts to the visual instance that appears after this flag.
A&L	Visual-Flagged Audio Detection (**VFAD**)	Specific visual information in the video serves as a flag, and the model reacts to the audio event that appears after this flag.
A&L	Audio-Visual Synchronization Detection (**AVSD**)	React when there is a mismatch between the audio and visual elements in the video.

B HUMAN VERIFICATION FOR PIFBENCH

After constructing PIFBench with the annotation pipeline described in Section 3.3, we conduct human verification to ensure annotation accuracy. Human annotators review the generated instructions, response contents, and response timestamps, correcting annotation errors and removing low-quality samples. This verification step ensures that the final benchmark follows the task definitions and provides reliable ground truth for evaluation. To maintain consistent annotation quality, we use three verification criteria:

1. The instruction must follow the definition of the corresponding task.
2. The ground-truth response content must match the instruction requirements, including fine-grained details such as case sensitivity and punctuation.
3. The response timestamp must be within a 3-second error margin.

For each erroneous sample, annotators either fix it or discard it. If the error is minor, such as a slightly inaccurate response or timestamp, annotators manually correct the sample. If an error causes the sample to violate the definition of its corresponding task or cannot be corrected reliably, the sample is removed. During verification, annotators also follow task-specific rules: for audio-related tasks, the audio cue in the instruction must be necessary for deciding when to react, rather than allowing the trigger time to be determined from visual information alone; purely visual tasks such as OR, AR, and ER should be checked without using audio; and audio annotations for silent videos should be removed.

To help annotators follow a consistent standard, we provide representative examples before annotation. These examples show annotators what a correct sample looks like and what common errors

look like, such as missing occurrence counts, incorrect timestamps, incorrect response contents, and severe hallucinations that require discarding the sample. We hire seven annotators for this verification process. Each annotator reviews about 200 examples, spends approximately 6 hours on average, and receives 50 USD as compensation.

After reviewing the annotations, we summarize the frequent error types into three groups:

1. **General errors**: abnormal responses caused by model API failures, hallucinated annotations, character-level mismatches between the ground truth and instruction requirements, and timestamp errors exceeding the 3-second threshold.
2. **OR/AR/ER errors**: missing occurrence counts for repeated targets, confusion between actions and events, and confusion between objects and events.
3. **TOD and audio-related errors**: incorrect event counts, incorrect judgments of background sounds, and spurious audio annotations for silent videos.

We also randomly check verified samples from all tasks to monitor annotation quality. If a sample is ambiguous or controversial, annotators first give their own judgment and then discuss it with the supervisor when needed. If the group still cannot reach an agreement, we discard the sample. This process provides an additional quality-control layer for the final benchmark.

C Additional Data Construction Details

C.1 Response Format Variant Definitions

In addition to deciding when to respond, PIF evaluates whether a model can follow user-specified response-format requirements, an important aspect of instruction following (Zhou et al., 2023). We create response-format variants by modifying only the required output form while keeping the trigger condition, response content, and response timestamp unchanged. This design jointly tests whether a model can identify the right response moment and format its response according to the instruction. It also increases instruction diversity and enables fine-grained instruction-following evaluation without additional video annotation. We define seven response-format variants:

- **Original**: Output the response content without modification.
- **Reversed lowercase**: Reverse the response sequence and convert all letters to lowercase.
- **Uppercase**: Convert the response content to uppercase.
- **Lowercase**: Convert the response content to lowercase.
- **Repeat n times**: Repeat the response content n times, separated by spaces, where n is randomly sampled from 2 to 5.
- **First n letters lowercase**: Output only the first n letters in lowercase, where n is randomly sampled from 2 to 5. Spaces are preserved but do not count toward the letter limit.
- **Letters to numbers**: Convert alphabetic characters to their corresponding numeric indices, with $a = 0$, $b = 1$, and so on, while preserving spaces and non-letter symbols.

C.2 Negative Sample Construction

The positive samples described above contain at least one time segment that requires the model to react. However, in real streaming scenarios, an entire video may contain no moment that satisfies the instruction, and the correct behavior is to remain silent throughout the video. We therefore construct negative samples with no response timestamp. For each task, negative samples are designed to remain related to the original positive videos or trigger conditions without fully satisfying the instruction. They account for approximately 10% of the benchmark and are deliberately misleading rather than trivially irrelevant, testing whether models can suppress unnecessary responses in realistic no-reaction cases. For general tasks, we mainly modify the prompts used in the data-construction pipeline so that the annotation model generates similar no-reaction samples, followed by human review. For specific tasks, we primarily rely on manual selection and labeling, using videos collected from YouTube. For Character Recognition (**CR**), we select videos in which no person appears.

For Audio-Visual Synchronization Detection (**AVSD**), we use normal videos whose audio and visual streams are synchronized. For Anomaly Detection (**AD**), we select videos without anomalous events. For Scene Transition Detection (**STD**), we select one-shot videos without scene transitions. For Multimodal Emotion Recognition (**MER**), we modify the target emotion in the instruction so that it differs from the emotion appearing in the video, such as changing the target from sad to happy.

C.3 Time-Sensitive Benchmark Conversion

We convert time-sensitive video benchmarks into PIF format because many timestamp-related tasks can naturally be reformulated as proactive instruction following. In the original format, a model is usually asked to answer a question at a given timestamp or for a given temporal segment. In the PIF format, the model receives the instruction at the beginning of the video and must decide when to respond during streaming. This conversion shows that PIF provides a general task format for timestamp-related video understanding tasks, and it also supports the ablation study in Section 5.3.2, where we compare the same question under the original format and the PIF format. As shown in our experiments, the same underlying task can lead to very different model behavior under these two formats, further revealing the weakness of current models in proactive interaction.

In general, our conversion follows three steps. First, we select representative time-sensitive benchmarks, including StreamingBench (Lin et al., 2026), OVO-Bench (Niu et al., 2025), and ViSpeak (Fu et al., 2025b), because they provide classic evaluation settings for timestamp-related video understanding. Second, we examine the subtask definitions in each benchmark and map the corresponding samples to the task types in PIFBench. For StreamingBench, we convert the Real-Time Visual Understanding and Omni-Source Understanding tasks; for its Contextual Understanding category, we convert the Anomaly Context Understanding and Proactive Output tasks. We also convert timestamp-related subtasks from OVO-Bench and ViSpeak according to their original task definitions. Third, we use annotation models to perform rule-based transformations that rewrite the original annotations into PIF-format instructions, response contents, and response timestamps. Most converted samples can be transformed reliably because their original annotations already contain timestamp-related information.

D Dataset Statistics

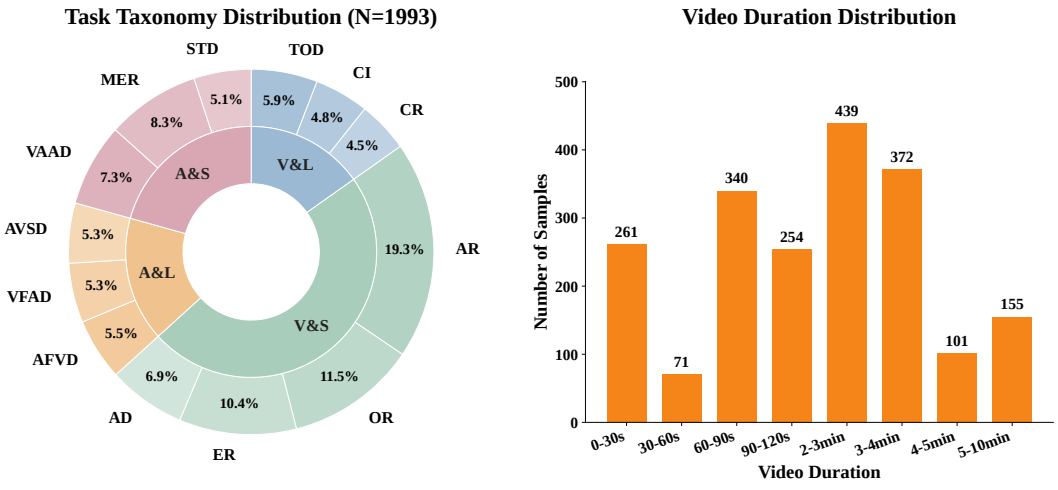

Figure 4: Statistics of PIFBench. Left: the task distribution across the 13 PIFBench task types. Right: the video-duration distribution, showing that PIFBench covers both short clips and longer streaming videos.

Figure 4 summarizes the statistics of PIFBench from two perspectives. First, the task distribution shows that PIFBench covers all 13 task types defined in Appendix A, with Action Recognition (**AR**), Object Recognition (**OR**), and Event Recognition (**ER**) contributing the largest portions. According to the task categories, PIFBench contains 732 audio-related samples and 1,261 non-audio

samples, covering both audio-visual interaction and visual-only proactive instruction following. In terms of data source, 1,391 samples (69.8%) are newly constructed from raw videos, while 602 samples (30.2%) are converted from existing time-sensitive benchmarks. Second, the duration distribution shows that PIFBench includes both short clips and longer streaming videos, with samples distributed from under 30 seconds to 5–10 minutes. This distribution supports evaluation across different streaming lengths, where models must decide when to respond during the video.

E Fine-Grained Analysis

E.1 Failure Analysis

As shown in Table 3, several models achieve notably low scores on PIFBench. We therefore inspect their predictions to identify the primary failure modes behind this performance. Phi-4-Multimodal struggles with the PIF format, and its low ReactAcc indicates difficulty in determining when to respond. Following the official recommendation for video input, we represent each video using a fixed number of sampled frames, which may limit temporal grounding compared with continuous streaming input. Appending a textual timestamp after each image does not resolve this issue, suggesting that explicit timestamps cannot compensate for weak temporal perception. The model also frequently violates the required JSON schema or produces repetitive malformed strings from which no valid reaction can be parsed. VideoLLM-Online produces few effective reactions and remains silent on most samples. Its frame-level response policy is poorly aligned with PIF's persistent trigger instructions, causing it either to miss the trigger or describe the current frame instead of producing the required response. MMDuet2 performs slightly better but exhibits a similar instruction-following failure mode. Because it is trained primarily on open-ended streaming dialogues, its responses are biased toward free-form descriptions of the current scene rather than strict instruction following. Consequently, it frequently produces visually relevant descriptions instead of the specified response, leading to low response and overall accuracy. MiniCPM-o 4.5 also performs poorly on PIFBench. MiniCPM-o 4.5 employs Omni-Flow to support proactive interaction through real-time reminders and comments grounded in ongoing multimodal context (Cui et al., 2026). In contrast, PIF requires persistent instruction tracking, detection of specified audio-visual triggers, and exact response generation, creating a mismatch between the model's proactive interaction setting and PIF. For most existing methods, proactive instruction following lies substantially outside their training distribution, presenting a distinct challenge beyond the proactive interaction behaviors targeted during training.

E.2 Subtask Analysis

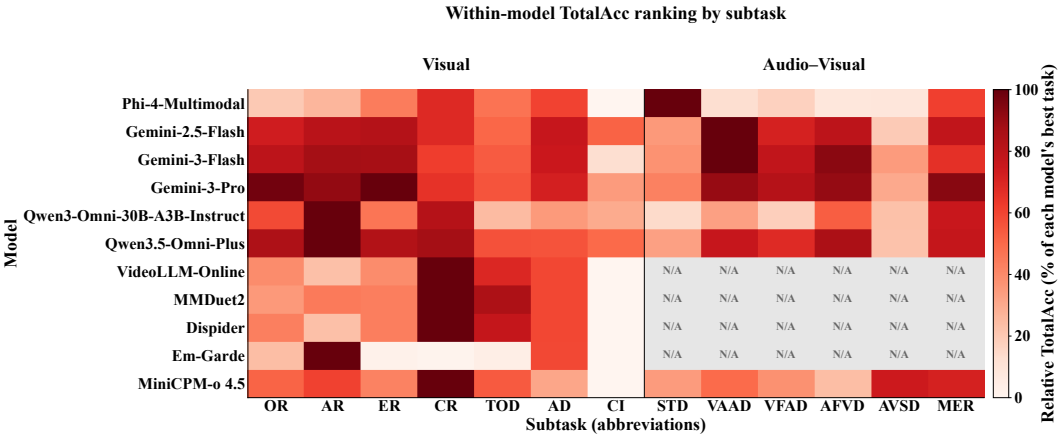

Figure 5: Within-model TotalAcc comparison across PIFBench subtasks. Each cell reports a model's TotalAcc on a subtask as a percentage of its best-performing subtask, with darker colors indicating stronger relative performance. Gray cells denote tasks that are not evaluated because the corresponding visual-only models do not support audio inputs.

Figure 5 reveals distinct task-level strengths and weaknesses across models. Although no subtask is uniformly easy for every model, Character Recognition (CR) and Anomaly Detection (AD) show relatively strong performance across most model families, while Action Recognition (AR) is also a common strength. Among the audio-visual subtasks, Multimodal Emotion Recognition (MER) exhibits the most consistently strong relative performance across the evaluated omni-modal models. Gemini-3-Flash maintains high relative performance on most visual and audio-visual subtasks, while Gemini-2.5-Flash follows a similar but less consistent pattern. Contextual Inference (CI), Scene Transition Detection (STD), and Audio-Visual Synchronization Detection (AVSD) are challenging for most evaluated models, particularly the non-Gemini omni-modal models. The visual-only proactive models perform relatively well on Character Recognition (CR), although their aggregate TotalAcc remains low in absolute terms. Em-Garde exhibits large variation across tasks, performing relatively well on Action Recognition (AR) but poorly on Event Recognition (ER), Temporal Occurrence Detection (TOD), and CI. MiniCPM-o 4.5 performs relatively well on CR and several audio-visual subtasks, although its absolute TotalAcc remains limited. Because each row is normalized by the corresponding model's best-performing subtask, the heatmap reflects task difficulty and model preferences rather than absolute performance differences across models.

E.3 Case Analysis

We qualitatively compare Gemini-3-Flash, Em-Garde, and MiniCPM-o 4.5 on four examples from PIFBench. These cases illustrate differences in response content, reaction timing, silence decisions, and modality coverage.

Table 7: Qualitative comparison of an offline omni-modal model and two proactive models. A prediction is correct only when both its content and response time match the ground truth.

Case	Ground Truth	Gemini-3-Flash	Em-Garde	MiniCPM-o 4.5
Action Recognition `action_recognition_14`	"Good sportsmanship" 1:38–1:41	"Good sportsmanship" at 1:39 Correct	"Good sportsmanship" at 0:02 Incorrect time	"Good sportsmanship." at 1:01 Content and time mismatch
Contextual Inference `contextual_inference_52`	"Leaving" 2:51–2:57	"Leaving" at 2:53 Correct	No output Missed	No output Missed
Character Recognition `character_recognition_1`	"2" 0:00–0:01	No output Missed	"2" at 0:18 Incorrect time	"2" at 0:00 Correct
Audio-Flagged Visual Detection `audio_flagged_visual_detection_5`	"Installation complete" 1:01–1:06	"Installation complete" at 1:02 Correct	Not evaluated Visual-only model	"Installation complete." at 0:47 Content and time mismatch

These cases reveal distinct strengths and failure modes. Gemini-3-Flash succeeds in three cases, including delayed contextual inference and cross-modal detection, but misses the immediately observable counting event. Em-Garde often identifies relevant content but struggles with timing and delayed triggers; it is not evaluated on the audio-visual case because it supports only visual input. MiniCPM-o 4.5 handles the early counting event correctly but remains sensitive to exact formatting and temporal alignment.

F Limitations

Our study has several limitations. First, although PIFBench covers diverse tasks, modalities, and video durations, it cannot represent the full range of real-world proactive interactions, including open-ended, multi-turn, dynamically changing, and personalized scenarios. Second, due to computational constraints, we use LoRA rather than full-parameter supervised fine-tuning; whether full fine-tuning can yield further improvements remains unexplored. Third, Proactive-Instruct-8K relies on automatic annotation and filtering, which may introduce noisy or imperfect supervision despite the quality-control procedure. Further improvements in training-data quality and chain-of-thought supervision may lead to more effective model adaptation. Finally, our model adaptation experiments focus on a single base model, and further experiments are needed to determine how well the proposed approach generalizes across architectures and deployment settings.

G Metric Definitions and Computation

PIFBench contains both positive and negative samples. A sample is positive if its ground truth contains at least one required reaction; otherwise, it is negative and the expected behavior is to

remain silent. Positive samples are further divided into *single-label* and *multi-label* samples. A single-label sample contains exactly one expected reaction, whereas a multi-label sample contains multiple expected reactions at different times in the same video. Multi-label cases commonly arise in tasks such as Anomaly Detection, where several anomalous events may occur, and Character Recognition, where multiple target characters may appear.

Let N be the number of evaluation samples and $\mathbb{I}[\cdot]$ be the indicator function. For sample i, let

$$\mathcal{G}_i = \{(y_{ij}, [a_{ij}, b_{ij}])\}_{j=1}^{m_i}$$

denote the temporally ordered ground-truth reactions, where y_{ij} is the required content and $[a_{ij}, b_{ij}]$ is its valid response window. Here, $m_i = 0$ for a negative sample, $m_i = 1$ for a positive single-label sample, and $m_i > 1$ for a positive multi-label sample. After removing outputs that explicitly indicate silence, let

$$\mathcal{P}_i = \{(\hat{y}_{ij}, \hat{t}_{ij})\}_{j=1}^{n_i}$$

denote the model's n_i effective reactions in temporal order, where $\hat{y}_{ij}$ is the predicted response content and $\hat{t}_{ij}$ is its predicted timestamp in seconds. For positive samples, predictions and ground-truth reactions are matched by their temporal order; a missing prediction is treated as an incorrect reaction. Following our evaluation implementation, only the first m_i predictions are matched when $n_i > m_i$.

For a positive sample, we first define its normalized reaction count as

$$E_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathbb{I}[j \le n_i].$$

We compute **TotalAcc** as

$$S_i = \begin{cases} \mathbb{I}[n_i = 0], & m_i = 0, \\ \dfrac{1}{m_i} \displaystyle\sum_{j=1}^{m_i} \mathbb{I}\big[j \le n_i \wedge \hat{y}_{ij} = y_{ij} \wedge \hat{t}_{ij} \in [a_{ij}, b_{ij}]\big], & m_i > 0, \end{cases}$$

$$\mathrm{TotalAcc} = \frac{1}{N} \sum_{i=1}^{N} S_i.$$

TotalAcc averages normalized correctness across samples: a reaction is correct only when the model produces the required content within the valid time window, while a negative sample is correct only when the model remains silent.

We compute **ReactAcc** using the sample-level decision score

$$R_i = \begin{cases} \mathbb{I}[n_i = 0], & m_i = 0, \\ E_i, & m_i > 0. \end{cases}$$

The overall reaction accuracy is

$$\mathrm{ReactAcc} = \frac{1}{N} \sum_{i=1}^{N} R_i.$$

ReactAcc focuses only on whether the model correctly decides whether a response is required, regardless of response content and timing. For multi-label samples, it gives partial credit according to the fraction of expected reactions produced.

We compute **RespAcc** as

$$C_i = \frac{1}{m_i} \sum_{j=1}^{m_i} \mathbb{I}[j \le n_i \wedge \hat{y}_{ij} = y_{ij}] \quad (m_i > 0),$$

$$\mathrm{RespAcc} = \frac{\sum_{i:m_i>0} C_i}{\sum_{i:m_i>0} E_i}.$$

RespAcc focuses on whether emitted reactions contain exactly the required content; therefore, only effective reactions from positive samples are included in its denominator.

Finally, we compute **WT-MAE** over reactions with correct content. For the j-th reaction in sample i, its weighted timing error is

$$e_{ij} = \begin{cases} \lambda_{\text{early}}(a_{ij} - \hat{t}_{ij}), & \hat{t}_{ij} < a_{ij}, \\ \lambda_{\text{late}}(\hat{t}_{ij} - b_{ij}), & \hat{t}_{ij} > b_{ij}, \\ 0, & \hat{t}_{ij} \in [a_{ij}, b_{ij}], \end{cases}$$

and the aggregate metric is

$$\text{WT-MAE} = \frac{\sum_{i:m_i>0} \frac{1}{m_i} \sum_{j=1}^{m_i} \mathbb{I}[j \le n_i \wedge \hat{y}_{ij} = y_{ij}] e_{ij}}{\sum_{i:m_i>0} C_i}.$$

Here, λ_{early} and λ_{late} are the penalty weights for early and late reactions, respectively. We set $\lambda_{\text{early}} = 2$ and $\lambda_{\text{late}} = 1$ because reacting before the required event occurs is generally more harmful than reacting late. WT-MAE is measured in seconds and is reported only when at least one response has correct content; lower values indicate more precise response timing. We report TotalAcc, ReactAcc, and RespAcc as percentages by multiplying the corresponding ratios by 100.

H USE OF LARGE LANGUAGE MODELS

We used large language models to assist with language polishing and improve the clarity and readability of this manuscript. The authors take full responsibility for the content of the paper.