STREAMPRO: TIMING-AWARE EVALUATION AND EFFECTIVE TRAINING FOR PROACTIVE STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Proactive streaming video understanding requires models to continuously monitor video streams and autonomously determine *whether*, *when*, and *what* to respond based on evolving user needs and visual context. Despite recent progress, both the evaluation and training of proactive models remain challenging. Existing benchmarks primarily focus on event-triggered responses, leaving anticipatory assistance underexplored, while their evaluation protocols often fail to jointly account for task-specific timing requirements, semantic correctness, and response frequency. Training also remains challenging: SFT suffers from severe imbalance between *silence* and *response* signals, whereas RL faces sparse response-level rewards and the challenge of optimizing interdependent response sequences. To address these challenges, we introduce **StreamPro**, a unified benchmark-and-training framework for proactive streaming video understanding. We first develop **StreamPro-Bench**, which evaluates three complementary dimensions of proactive capability: *Perceptual Understanding*, *Temporal Reasoning*, and *Anticipatory Assistance*. Its **StreamPro-F1** metric jointly evaluates semantic correctness and response timing, while accounting for both excessive and missed responses. We further propose a two-stage training framework that combines class-balanced supervised fine-tuning with reinforcement learning. Specifically, *CB-Stream Loss* alleviates supervision imbalance during SFT, while GRPO with a refined *Response-Matching (RM)* reward and a rubric-based *Sequence-Quality (SQ)* reward provides denser feedback and encourages coherent response sequences. We construct **StreamPro-SFT-63K** and **StreamPro-RL-3K** to support the two training stages. Experiments show that StreamPro substantially improves proactive streaming video understanding: it achieves an average StreamPro-F1 of 32.8 on StreamPro-Bench, compared with 9.2 for the strongest existing baseline, while maintaining strong performance on real-time streaming video understanding, reaching 79.8 on StreamingBench-RTVU.

1 INTRODUCTION

Streaming video understanding (Chen et al., 2024) requires models to continuously process visual inputs and make decisions as a video unfolds. In conventional streaming settings, models primarily focus on understanding the current context or responding to user queries. In proactive streaming video understanding setting (Wang et al., 2024b; 2025c; Xia et al., 2025), where the events of interest occur after the user poses a query, models must determine not only *what* to say, but also *whether* and *when* to respond. Such capabilities are essential for real-world assistants to provide timely and context-aware assistance.

Despite recent progress, developing effective proactive streaming video understanding models remains challenging. In particular, existing approaches lack comprehensive evaluation protocols and face distinctive difficulties in learning when and how to respond. ***(1) Limited coverage of proactive behaviors.*** Existing benchmarks commonly evaluate proactive behavior through event-triggered response settings, in which models respond after a relevant event has occurred or answer a proactive question based on an observed activity (Niu et al., 2025; Lin et al., 2026). Although these tasks are important, they do not fully capture the range of behaviors required by proactive assistants. ***In practical interactions, a model may also need to reason about the temporal evolution of an event,***

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.

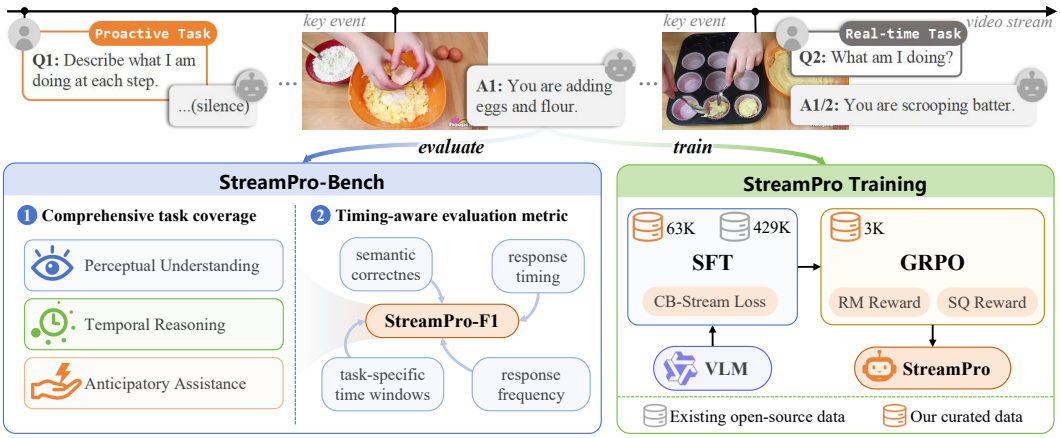

Figure 1: StreamPro Overview. **Top:** Difference between proactive and real-time tasks. **Bottom left:** StreamPro-Bench is a comprehensive benchmark covering three dimensions of proactive capability. Its StreamPro-F1 metric jointly assesses response timing and semantic correctness, uses task-specific time windows, and penalizes excessive and missed responses. **Bottom right:** StreamPro's two-stage training consists of SFT with CB-Stream Loss and GRPO with RM and SQ rewards.

or anticipate upcoming assistance needs from the evolving visual context and respond before the relevant situation fully unfolds. A systematic evaluation should therefore cover not only the recognition of relevant events, but also temporal reasoning and anticipatory assistance. ***(2) Incomplete evaluation of response behavior.*** Proactive understanding is inherently a sequential decision problem. A model must produce semantically appropriate responses at suitable times while remaining silent when no response is warranted. Existing metrics do not always capture these requirements jointly. For example, the PAUC metric used in ProactiveVideoQA (Wang et al., 2025d) does not explicitly account for repeated responses, potentially favoring models that produce redundant outputs. Moreover, different proactive tasks may tolerate different response delays. A modest delay may be acceptable when reporting an ordinary event, whereas timely intervention may be essential for anomaly detection or time-sensitive assistance. These observations motivate an evaluation protocol that jointly considers response correctness, timing, and response frequency under task-specific temporal requirements.

Besides, training proactive models presents a set of challenges. During supervised fine-tuning (SFT), streaming data typically contains many more time steps when the model should remain silent than time steps when it should respond. This imbalance can cause the training signal to be dominated by silence and make it difficult for models to learn when to intervene (Xia et al., 2025; Lu et al., 2026). During reinforcement learning (RL), directly optimizing a strict reward that requires both correct response content and precise response timing can result in sparse feedback. Such a reward provides limited guidance for responses that are partially correct or temporally near-miss. Furthermore, proactive assistance often consists of multiple interdependent responses. A sequence of individually plausible responses may still be redundant, incomplete, poorly timed, or inconsistent as a whole. Consequently, effective training requires both response-level feedback and sequence-level supervision.

To address these challenges, we introduce **StreamPro**, a unified benchmark-and-training framework for proactive streaming video understanding. As illustrated in Fig. 1, our framework consists of a new benchmark, a timing-aware evaluation metric, and a two-stage training paradigm. First, we introduce **StreamPro-Bench**, which organizes proactive capabilities into three complementary dimensions: *Perceptual Understanding*, *Temporal Reasoning*, and *Anticipatory Assistance*. The benchmark covers seven representative tasks across 577 videos and 1,285 question-answer pairs. To evaluate proactive response behavior, we propose *StreamPro-F1*, which jointly considers semantic correctness, response timing and response frequency, and uses task-specific temporal windows to reflect different timing requirements across tasks. Second, we propose a two-stage training framework for learning proactive response behaviors. In the SFT stage, we introduce *CB-Stream Loss*, which applies class-balanced reweighting (Cui et al., 2019) to alleviate the severe imbalance between silence and response supervision. In the RL stage, we adopt GRPO (Guo et al., 2025) with two complementary rewards. The *Response-Matching (RM)* reward is refined to provide denser feedback on response timing and semantic correctness. The rubric-based *Sequence-Quality (SQ)* reward evaluates the quality of the

Table 1: Comparison of proactive streaming benchmarks. Len. denotes average video length, and Proact. Ratio denotes the percentage of proactive questions. PU: Perceptual Understanding, TR: Temporal Reasoning, and AA: Anticipatory Assistance. △ indicates partial coverage.

Benchmark	#Videos	#Samples	Len.(s)	Proact. Ratio(%)	PU	TR	AA
StreamingBench (Lin et al., 2026)	900	4,500	243.1	5.5	✓	✗	✗
OVO-Bench (Niu et al., 2025)	644	2,814	428.9	10.5	✓	△	✗
ProactiveVideoQA (Wang et al., 2025d)	1,377	1,427	124.8	100.0	✓	✗	✗
Omni-MMI (Wang et al., 2025e)	1,121	2,290	324.3	22.0	✓	△	✗
StreamPro-Bench	577	1,285	130.0	100.0	✓	✓	✓

complete response sequence, including its granularity, sequencing, and coverage. Together, these rewards encourage models to produce responses that are not only individually appropriate but also coherent at the sequence level.

To support the two training stages, we construct **StreamPro-SFT-63K** and **StreamPro-RL-3K**. Extensive experiments demonstrate that StreamPro substantially improves proactive streaming video understanding while preserving performance on related tasks. On StreamPro-Bench, StreamPro achieves an average StreamPro-F1 of 32.8, compared with 9.2 for the strongest existing baseline. It also maintains strong performance on real-time streaming benchmarks, achieving 79.8 on StreamingBench-RTVU, as well as competitive performance on offline video understanding tasks.

Our contributions are summarized as follows: (1) We introduce **StreamPro-Bench**, a multi-dimensional benchmark for proactive streaming video understanding, covering *Perceptual Understanding*, *Temporal Reasoning*, and *Anticipatory Assistance* across seven representative tasks, together with **StreamPro-F1**, a timing-aware metric that accounts for semantic correctness, response timing, and response frequency. (2) We propose **StreamPro**, a two-stage training framework that addresses key challenges in proactive learning. It combines *CB-Stream Loss* for mitigating supervision imbalance during SFT with GRPO-based training using a refined *RM reward* and a rubric-based *SQ reward* for response-level and sequence-level optimization. We also construct the *StreamPro-SFT-63K* and *StreamPro-RL-3K* datasets. (3) Experiments demonstrate the effectiveness of StreamPro. It achieves an average StreamPro-F1 of 32.8 on StreamPro-Bench, substantially outperforming existing baselines while maintaining strong real-time streaming video understanding performance.

2 Related Work

Proactive Streaming Video Understanding. Existing approaches to proactive streaming video understanding can be broadly categorized into two groups. (1) ***Module-based Proactive Models*** introduce additional modules to explicitly control response timing and activation, often relying on heuristics or task-specific components (Zhang et al., 2026a; Ding et al., 2025; Zhang et al., 2026b; Fu et al., 2025b). For example, Dispider (Qian et al., 2025b) decomposes streaming interaction into perception, decision, and reaction modules. StreamBridge (Wang et al., 2025a) augments offline VideoLLMs with a lightweight external gating model. StreamAgent (Yang et al., 2025) constructs a complete agent framework for temporal decision-making. (2) ***Token-based End-to-End Proactive Model*** use spatial tokens to integrate proactive capability directly into a single model. A key challenge in this paradigm is the imbalance between *silence* and *response* supervision signals, which biases the model toward remaining silent excessively. VideoLLM-Online (Chen et al., 2024) mitigates the imbalance by controlling the activation threshold of the EOS token during inference. Streamo (Xia et al., 2025) further alleviates this issue. To better address this issue, we propose CB-Stream Loss, a simple yet effective loss that re-weights streaming control tokens based on their effective frequency.

Recently, several works have explored RL to optimize proactive behavior in streaming settings. Existing rewards emphasize response timing and correctness (Liu et al., 2026) and repetition control (Wang et al., 2025c), but provide limited supervision for overall sequence organization. Strict response matching in TOM (Qian et al., 2025a) also gives no credit for partial correctness. We combine a refined RM reward for denser feedback with a rubric-based SQ reward for overall sequence quality.

Proactive Streaming Video Benchmark. With the rapid progress of streaming video understanding, many benchmarks have been proposed to evaluate proactive capabilities. StreamingBench evaluates proactive models through alert-based questions, where models are required to respond upon observing

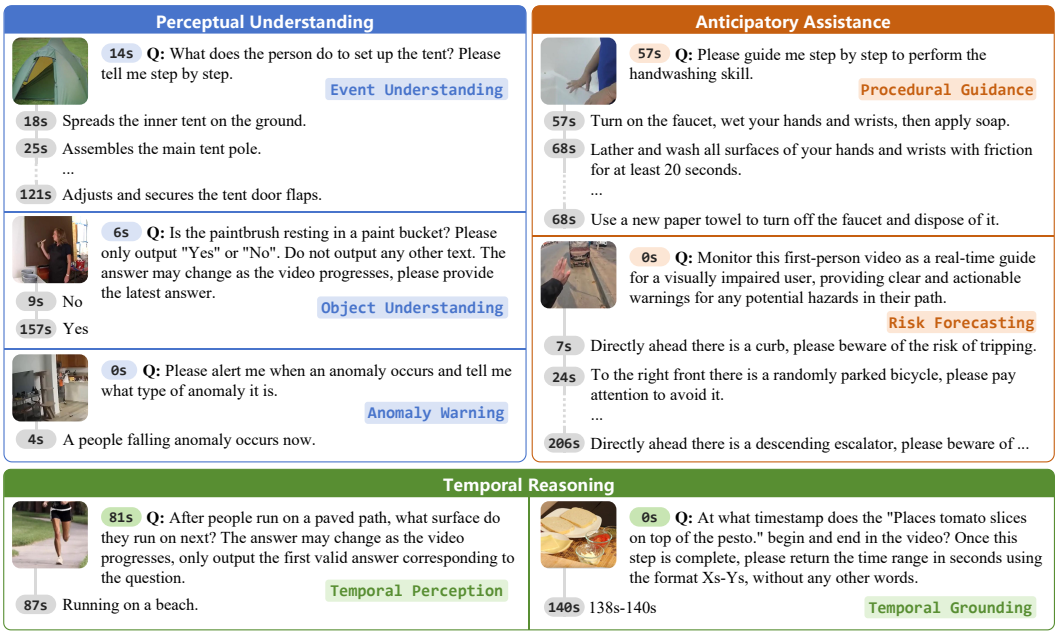

Figure 2: Task case illustration of StreamPro-Bench. It contains 7 tasks categorized into three major dimensions: Perceptual Understanding, Temporal Reasoning, and Anticipatory Assistance. The timestamps of the user query and the ground truth are explicitly annotated along the timelines.

visual evidence in the video (Lin et al., 2026). OVO-Bench (Niu et al., 2025) evaluates proactive models through three tasks: repetition event counting, sequential step recognition, and clue-revealing response. ProactiveVideoQA (Wang et al., 2025d) is the first benchmark specifically designed for proactive tasks, and it introduces PAUC evaluation metric. Omni-MMI (Wang et al., 2025e) aims for comprehensive multimodal interaction evaluation by introducing audio inputs and multi-turn dialogue tasks. Compared with existing benchmarks, StreamPro-Bench covers a broader range of tasks that require models to reason about temporal relationships, identify appropriate assistance opportunities from evolving visual context, and guide users through subsequent actions. It uses StreamPro-F1 to jointly assess response timing and semantic correctness while penalizing excessive and missed responses.

3 StreamPro-Bench

In this section, we introduce **StreamPro-Bench**. We first define the task taxonomy across three dimensions (Section 3.1). We then detail the data construction pipeline and benchmark statistics (Section 3.2). Finally, we present the **StreamPro-F1 Score**, a tailored evaluation metric that jointly assesses temporal alignment and semantic correctness (Section 3.3).

3.1 Task Taxonomy

We argue that a strong proactive model should exhibit capabilities across three dimensions: Perceptual Understanding, Temporal Reasoning, and Anticipatory Assistance. Across these three dimensions, we define 7 specific tasks, as illustrated in Figure 2. We provide more cases in Appendix D.

Perceptual Understanding. This dimension evaluates the foundational ability to continuously perceive the states, dynamics, and changes of entities within streaming video inputs. We assess this capability through three tasks: (1) *Event Understanding* (EU): describe the continuous steps of an evolving event as it unfolds (e.g., the sequential actions of a person setting up a tent). (2) *Object Understanding* (OU): track the status of specific objects, requiring the model to instantly confirm when a state transition occurs. (3) *Anomaly Warning* (AW): trigger immediate warnings upon detecting sudden anomalous events, such as a person falling or a fire breaking out.

Temporal Reasoning. Building on foundational perception, this dimension assesses whether models can track when events happen and how they depend on each other in time. It includes two tasks: (1) *Temporal Perception* (TP): given a target event, identify which events occur after it, testing fine-grained chronological understanding. (2) *Temporal Grounding* (TG): output the start and end timestamps of the given event in a streaming setting as soon as the event concludes.

Anticipatory Assistance. This dimension evaluates whether models can anticipate upcoming task-relevant events from the evolving visual context and provide timely guidance before the next action or potential risk unfolds. It comprises two tasks: (1) *Procedural Guidance* (PG): given a goal, identify which procedural step the user is currently performing, track its progress, infer the next step, and promptly provide appropriate guidance once the step is completed. (2) *Risk Forecasting* (RF): in first-person navigation scenarios from the perspective of a visually impaired person, anticipate potential accidents roughly 3 seconds before they occur, warn the user, and provide actionable navigation advice.

3.2 Benchmark Construction

We construct data for all 7 tasks using a pipeline that includes a two-agent verification loop and human refinement. Given the complexity of Risk Forecasting task, every sample is manually reviewed, and incorrect annotations are manually redone. Further details are provided in Appendix A.1. In total, StreamPro-Bench comprises 577 videos and 1,285 QA pairs. Figure 3 shows the number of instances for each task and the total number of corresponding responses, as a single question may be associated with multiple answers. Although Anticipatory Assistance contains only 108 queries, each query corresponds to an average of 7.6 ground-truth response instances, resulting in 819 response instances in total. The comparison with current mainstream benchmarks is shown in Table 1.

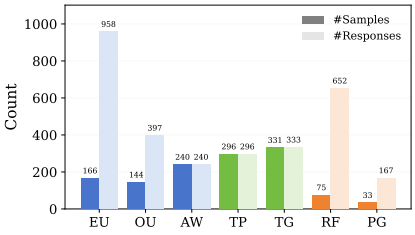

Figure 3: Number of samples and responses of each StreamPro-Bench task.

3.3 Evaluation Protocol

Evaluation Metric. We introduce **StreamPro-F1**, which jointly evaluates response timing and semantic correctness while penalizing excessive and missed responses. For a predicted response and a ground-truth response, the **semantic score** S_{sem} is evaluated by an LLM judge for most tasks. For Temporal Grounding, we use Intersection over Union (IoU). The **timing score** S_{time} is evaluated using the temporal distance between the predicted response time t_{pred} and the optimal response timing t_{gt}. It decreases linearly to zero as this distance reaches the task-specific tolerance τ. The joint score S is the product of the timing and semantic scores, so a response receives a high score only when it is both timely and semantically accurate:

$$S_{\text{time}} = \max\left(0, 1 - \frac{\text{dist}(t_{\text{pred}}, t_{\text{gt}})}{\tau}\right), \qquad S = S_{\text{time}} \cdot S_{\text{sem}}. \tag{1}$$

At the trajectory level, each matched pair contributes its joint score S_i. We compute score-weighted precision P and recall R using the total numbers of predicted and ground-truth responses, denoted by N_{pred} and N_{gt}. Excessive responses lower precision, while missed ground-truth responses lower recall. The final StreamPro-F1 score is the harmonic mean of P and R:

$$P = \frac{\sum_i S_i}{N_{\text{pred}}}, \qquad R = \frac{\sum_i S_i}{N_{\text{gt}}}, \qquad F_1 = \frac{2PR}{P+R}. \tag{2}$$

StreamPro-F1 aligns well with human preferences. See detailed information in Appendix A.3.

Evaluation Method. For proactive models, we evaluate only those with officially released proactive inference scripts to ensure fair and reproducible evaluation (Chen et al., 2024; Wang et al., 2024b; 2025c; OpenBMB, 2026; Xia et al., 2025). For offline models (Bai et al., 2025b;a), we design a separate decision-tree-based frame-by-frame evaluation protocol to assess their proactive capabilities. Further details regarding the evaluation protocol are provided in Appendix A.2.

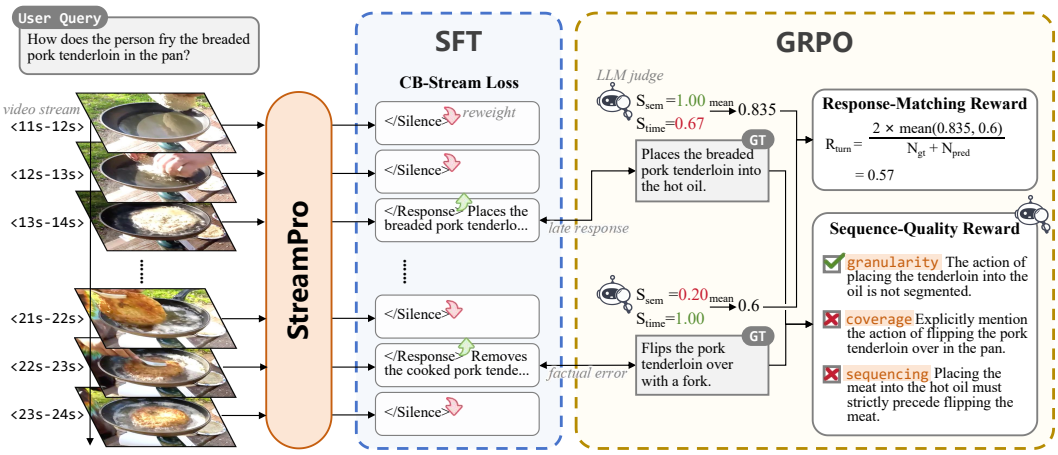

Figure 4: Overview of the StreamPro training framework. **In the SFT stage**, the CB-Stream Loss adjusts the weights of the `</Silence>` and `</Response>` decision tokens according to their effective sample sizes in each batch. **In the GRPO stage**, the RM reward combines the semantic correctness assessed by an LLM judge with the timeliness decayed based on temporal distance. Here, N_{gt} and N_{pred} denote the respective total numbers of ground truth and predicted events, which are both 2 in this illustrated example. The rubric-based SQ reward assesses granularity, sequencing, and coverage across the complete response sequence.

4 StreamPro Training Framework

This section introduces our two-stage StreamPro training framework (Figure 4): Section 4.1 details the SFT stage, Section 4.2 the RL stage, and Section 4.3 the training data for both stages.

4.1 Supervised Fine-Tuning

Proactive Streaming Decision Format. At each time step, the model either remains silent or generates a response conditioned on the current video context. Specifically, the model outputs `</Silence>` when no response is required, and `</Response>` followed by the corresponding textual answer when a response is triggered. We denote the decision token set as $\mathcal{S} = \{\texttt{</Silence>}, \texttt{</Response>}\}$, and the standard language token set as $\mathcal{T}$.

CB-Stream Loss. Proactive streaming video data exhibits a severe imbalance between response and silence signals, which causes standard cross-entropy training to favor conservative policies that over-predict silence. To mitigate this issue, we adopt a simple yet effective class-balanced reweighting strategy (Cui et al., 2019) based on the effective number of samples. For each decision token class $k \in \mathcal{S}$, we define the effective sample size E_k and corresponding class-balanced weight $\hat{w}_k^{\text{CB}}$ as:

$$E_k = \frac{1-\beta^{n_k}}{1-\beta}, \quad \hat{w}_k^{\text{CB}} = \frac{1/E_k}{\sum_{j\in\mathcal{S}} 1/E_j} \cdot |\mathcal{S}|, \tag{3}$$

where n_k is the frequency of class k computed over the current batch, and $\beta \in [0, 1)$ controls the degree of reweighting. We further introduce a constant scaling factor λ_{text} to balance optimization between decision tokens and language tokens. The final training objective is defined as:

$$\mathcal{L}_{\text{CB}} = \frac{1}{N}\sum_{i=1}^{N} w_i^{\text{CB}} \cdot \big[-\log p_i\big], \quad w_i^{\text{CB}} = \begin{cases} \hat{w}_{y_i}^{\text{CB}}, & y_i \in \mathcal{S}, \\ \lambda_{\text{text}}, & y_i \in \mathcal{T}. \end{cases} \tag{4}$$

4.2 Reinforcement Learning

Although SFT equips the model with basic streaming and proactive response capabilities, it is limited by exposure bias: the model is trained on ground-truth decisions but must rely on its own decisions at inference, so errors accumulate over the video stream. To mitigate this, we employ GRPO (Guo

et al., 2025) with three rewards: a format reward, a refined *Response-Matching (RM)* reward, and a rubric-based *Sequence-Quality (SQ)* reward. The total reward is:

$$R = w_{\text{fmt}} R_{\text{fmt}} + w_{\text{RM}} R_{\text{RM}} + w_{\text{SQ}} R_{\text{SQ}}, \tag{5}$$

where the w terms are the corresponding reward weights.

Format Reward. The format reward checks the output at each time step: a standalone `</Silence>` token receives a score of 1, as does a `</Response>` token followed by non-empty text; all other outputs receive 0. We average the step scores to obtain R_{fmt}.

RM Reward. Although StreamPro-F1 aligns well with human preferences (Appendix A.3), it is too strict to serve directly as an RL reward: its multiplicative score and narrow temporal tolerance yield sparse rewards, giving the model little guidance for improving partially correct responses. We therefore make two changes. (1) *Denser scoring*: we use the mean of the timing and semantic scores instead of their product, so a weak component no longer erases the other. (2) *Wider tolerance*: we increase the temporal tolerance τ to widen the matching window and slow the decay of the timing score, allowing small temporal deviations to receive partial credit. We use a single τ across all tasks to keep the reward design simple and task-agnostic. The reward retains the F1 penalty for extra and missed responses:

$$R_{\text{RM}} = \frac{2 \sum_i S'_i}{N_{\text{pred}} + N_{\text{gt}}}, \qquad S'_i = \frac{S_{\text{time},i} + S_{\text{sem},i}}{2}. \tag{6}$$

SQ Reward. The RM reward scores individual responses but cannot assess whether they form a coherent sequence. In Event Understanding and Procedural Guidance, the model must choose an appropriate level of detail, cover the key points, and preserve their order. We use the SQ reward to assess these properties over the complete sequence. The rubric covers three properties: (1) **Granularity**, responses are neither too fragmented nor too coarse relative to event duration; (2) **Sequencing**, chronological consistency; (3) **Coverage**, essential points are included while hallucinations are penalized. Concretely, before training, an LLM designer generates a rubric of N_c binary checkpoints from each sample's question and ground truth; during GRPO, an LLM evaluator checks the generated sequence against each checkpoint and assigns $c_i \in \{0, 1\}$. The final reward averages the checkpoint scores:

$$R_{\text{SQ}} = \frac{1}{N_c} \sum_{i=1}^{N_c} c_i. \tag{7}$$

4.3 Training Data

We construct two datasets using the StreamPro-Bench data pipeline: **StreamPro-SFT-63K** and **StreamPro-RL-3K**. We enforce strict data isolation throughout dataset construction to ensure that all training data are fully disjoint from the benchmark evaluation set, preventing any potential data leakage. We then use these datasets in the StreamPro training framework. In the SFT stage, we jointly train on real-time streaming and proactive data, including TimeChat-Online-139K Yao et al. (2025), VideoChat-Flash-3K Li et al. (2024b), StreamPro-SFT-63K, and 287K filtered samples from Streamo-Instruct-465K Xia et al. (2025). In the RL stage, we focus on proactive tasks and train the model using StreamPro-RL-3K. The dataset details and statistics are provided in Appendix B.2.

5 Experiments

5.1 Experiment Settings

Benchmarks. We evaluate our proposed model across three distinct categories of tasks: **(1) *Proactive Tasks.*** We evaluate on StreamPro-Bench (SPB) and Forward Active Responding (FAR) tasks from OVO-Bench Niu et al. (2025). Performance is measured using StreamPro-F1. We compare our approach with baseline models that provide open-source scripts for proactive inference Chen et al. (2024); Wang et al. (2024b; 2025c); OpenBMB (2026); Xia et al. (2025). **(2) *Real-time Streaming Tasks.*** We evaluate on Real-Time Visual Perception (RTVP) from OVO-Bench, alongside Real-Time Visual Understanding (RTVU) and Contextual Understanding (CU) from StreamingBench Lin et al. (2026). We exclude the PO task from RTVU as it evaluates proactive behavior, and compute the

Table 2: Performance on proactive tasks across StreamPro-Bench and OVO-Bench. PU: Perceptual Understanding, TR: Temporal Reasoning, AA: Anticipatory Assistance.

Methods	Params	StreamPro-Bench										OVO-Bench		
		PU			TR			AA			Avg. F1	FAR		
		Time	Sem.	F1	Time	Sem.	F1	Time	Sem.	F1		Time	Sem.	F1
Offline Models + Proactive Prompt														
Qwen2.5-VL-7B (Bai et al., 2025b)	7B	8.7	16.3	4.2	5.0	0.9	0.6	7.5	6.1	4.4	3.1	-	-	-
Qwen3-VL-8B (Bai et al., 2025a)	8B	59.5	50.1	5.7	28.8	17.6	0.9	33.0	17.4	3.1	3.2	-	-	-
Open-Source Proactive Models														
VideoLLM-Online (Chen et al., 2024)	8B	14.6	1.3	0.3	0.6	0.1	0.0	21.5	8.0	**8.8**	3.0	4.4	0.4	0.1
MMDuet (Wang et al., 2024b)	7B	50.5	34.7	9.1	72.9	21.4	1.6	33.6	15.8	3.8	4.8	42.5	19.2	5.7
MMDuet2 (Wang et al., 2025c)	3B	41.0	29.2	9.7	49.3	31.8	3.4	27.3	9.7	3.8	5.6	33.5	18.1	6.5
MiniCPM-o-4.5 (OpenBMB, 2026)	9B	18.8	16.9	13.2	2.4	0.4	0.1	7.5	6.1	4.4	5.9	15.6	10.6	8.4
Streamo (Xia et al., 2025)	3B	24.6	20.5	9.0	27.9	19.5	14.3	9.2	4.0	4.4	9.2	11.5	7.7	5.4
StreamPro-SFT	3B	51.8	44.1	13.7	45.8	26.8	22.0	42.0	20.3	4.3	13.3	41.6	25.4	9.5
StreamPro-GRPO	3B	55.6	51.3	35.0	58.7	47.4	34.7	14.2	8.9	5.6	25.1	37.5	20.8	14.4
StreamPro-SFT	4B	67.2	61.8	23.8	46.3	33.5	29.1	43.6	21.9	6.5	19.8	41.6	32.2	17.3
StreamPro-GRPO	4B	65.7	60.6	**44.8**	68.3	61.0	**45.1**	20.8	15.6	8.5	**32.8**	44.2	33.9	**20.6**

Table 3: Performance on real-time streaming tasks and offline video benchmarks. † denotes results from our evaluation of the official Streamo-3B checkpoint.

Methods	Params	OVO-Bench	StreamingBench		VideoMME	LongVideoBench
		RTVP	RTVU	CU		
Non-Proactive Streaming Models						
Flash-VStream (Zhang et al., 2024a)	7B	28.4	25.9	25.6	-	-
TimeChat-Online (Yao et al., 2025)	7B	61.9	75.3	37.5	62.4	55.4
StreamForest (Zeng et al., 2025)	7B	61.2	77.3	-	61.4	-
StreamingVLM (Xu et al., 2025)	7B	62.0	-	-	65.1	59.0
Module-based Proactive Models						
Dispider (Qian et al., 2025b)	7B	54.5	67.5	34.0	57.2	-
ViSpeak (Fu et al., 2025b)	7B	66.3	74.5	39.9	58.0	-
StreamAgent (Yang et al., 2025)	7B	61.3	74.6	36.5	62.9	57.9
QueryStream (Zhang et al., 2026a)	7B	63.1	75.3	-	63.8	57.3
Thinking-QwenVL (Zhang et al., 2026b)	7B	64.7	71.6	-	67.7	**62.0**
Token-based End-to-End Proactive Models						
VideoLLM-Online (Chen et al., 2024)	8B	20.8	36.0	28.1	-	-
Streamo-3B† (Xia et al., 2025)	3B	61.2	75.8	41.1	58.0	51.8
Streamo-7B (Xia et al., 2025)	7B	66.0	-	-	**67.9**	59.2
StreamPro-SFT	3B	62.1	77.1	40.0	60.7	54.6
StreamPro-GRPO	3B	64.2	75.9	39.6	60.9	54.6
StreamPro-SFT	4B	68.2	79.3	**46.9**	66.5	61.9
StreamPro-GRPO	4B	**69.7**	**79.8**	46.7	67.3	60.4

RTVU average over the remaining tasks. For these tasks, we compare against 11 representative streaming models Zhang et al. (2024a); Yao et al. (2025); Zeng et al. (2025); Xu et al. (2025); Qian et al. (2025b); Fu et al. (2025b); Yang et al. (2025); Zhang et al. (2026a;b); Chen et al. (2024); Xia et al. (2025). **(3) *Offline Tasks.*** We evaluate on VideoMME Fu et al. (2025a) and LongVideoBench Wu et al. (2024).

Implementation Details. We employ Qwen2.5-VL-3B Bai et al. (2025b) and Qwen3-VL-4B Bai et al. (2025a) as our backbone models, and ablate on the 3B model. During SFT, we train both the projector and the LLM for 1 epoch using 64 H100 GPUs for 24 hours, with a learning rate of 1×10^{-5} and a batch size of 512. The reweighting hyperparameter β is set to 0.9999, and the text scaling factor λ_{text} is set to 2. During RL, we implement the GRPO pipeline using veRL Sheng et al. (2024) and vLLM Kwon et al. (2023), training for 1 epoch using 8 H100 GPUs for 28 hours with a learning rate of 10^{-6} and a global batch size of 16. We sample $G = 8$ generations per video context with a temperature of 1.0. For the GRPO rewards, we set the temporal tolerance $\tau = 8$, along with the reward weights $w_{\text{fmt}} = 0.1$, $w_{\text{RM}} = 0.45$, and $w_{\text{SQ}} = 0.45$. All videos are sampled at 1 FPS. During inference, we employ a sliding window of 200 dialogue turns to improve inference efficiency.

Table 4: Ablation of the loss function and text token weight λ_{text} during SFT.

Loss	SPB	OVO-RTVP	VideoMME
CE	4.8	**62.3**	60.4
CB-Stream ($\lambda = 1$)	**13.7**	61.8	60.4
CB-Stream ($\lambda = 2$)	13.3	62.1	60.7
CB-Stream ($\lambda = 3$)	12.7	60.8	**60.9**

Table 5: Ablation of RM score combination, temporal tolerance τ, and the SQ reward during RL.

No.	Comb.	τ	SQ Reward	SPB	OVO-RTVP	VideoMME
1	Mean	3	✓	18.7	61.6	60.9
2	Mean	5	✓	22.2	61.4	60.8
3	Mean	8	✓	**25.1**	**64.2**	60.9
4	Mean	8	✗	20.9	60.9	60.6
5	Prod.	8	✓	21.8	60.6	**61.0**

5.2 Main Results

Proactive Tasks. As shown in Table 2, both training stages improve proactive performance. On StreamPro-Bench, the 4B SFT and GRPO models achieve F1 scores of 19.8 and 32.8, respectively, substantially exceeding the best previous proactive model's score of 9.2. On OVO-Bench FAR, evaluated with the StreamPro-F1 metric, the 4B model improves from 17.3 after SFT to 20.6 after GRPO, further demonstrating that our performance gains generalize beyond our StreamPro-Bench.

Real-Time Streaming and Offline Tasks. As shown in Table 3, StreamPro-GRPO-4B outperforms the published 7B baselines on real-time streaming tasks (OVO-Bench RTVP and StreamingBench RTVU). Furthermore, it achieves performance comparable to many 7B models on offline benchmarks (VideoMME and LongVideoBench), despite the absence of offline data throughout both training stages. When compared against StreamPro-SFT-4B, GRPO yields slightly reduced scores on CU as well as LongVideoBench. These modest performance drops are reasonable, considering that the RL stage is trained solely using proactive streaming data. Taken together, these results suggest that StreamPro-GRPO-4B maintains favorable generalization capabilities beyond proactive tasks. More experimental results are provided in Appendix C.

5.3 Ablation Study

Effect of CB-Stream Loss. As shown in Table 4, replacing the standard CE loss with CB-Stream substantially improves proactive performance, increasing the SPB score from 4.8 to 13.7, while largely preserving performance on OVO-RTVP and VideoMME. For the text-token weight λ_{text}, the results with $\lambda_{\text{text}} = 1$ and 2 are comparable overall. Considering the balance between proactive performance and general video understanding, we set $\lambda_{\text{text}} = 2$ in our final configuration.

RM Reward Ablation. Compared to StreamPro-F1, the RM reward uses a larger shared tolerance τ and takes the mean instead of the product of timing and semantic scores to provide denser feedback. As shown in rows 1–3 of Table 5, increasing τ progressively improves StreamPro-Bench F1. A larger tolerance widens the matching window and allows responses close to the reference time to receive partial credit. Comparing rows 3 and 5, replacing the product with the mean improves StreamPro-Bench F1 by 3.3 points and OVO-RTVP by 3.6 points, suggesting that averaging provides more effective feedback when either timing or semantic quality is weak.

SQ Reward Ablation. The SQ reward evaluates granularity, sequencing, and coverage across the complete response sequence. Comparing rows 3 and 4 of Table 5, adding SQ improves StreamPro-Bench F1 by 4.2 points and OVO-RTVP by 3.3 points. These gains indicate that proactive training benefits from considering how responses relate to one another, beyond merely scoring each matched response independently.

6 Conclusions

In this paper, we propose StreamPro-Bench, a comprehensive benchmark to evaluate proactive video understanding along three dimensions: Perceptual Understanding, Temporal Reasoning, and Anticipatory Assistance. To address the silence–response imbalance and reward sparsity, we introduce the StreamPro framework: a CB-Stream loss that balances the two signals during SFT, and a RL stage combining a refined RM reward with a rubric-based SQ reward that assesses sequence-level quality. StreamPro substantially improves proactive capability while preserving real-time streaming performance and remaining competitive on offline benchmarks. StreamPro-Bench and StreamPro together provide a systematic foundation for proactive streaming video understanding, and we hope they facilitate future work toward real-world proactive assistant systems.

AI USE STATEMENT

Generative AI tools were used during benchmark construction, as detailed in the paper. They were additionally used to assist with language editing and LaTeX formatting.

ETHICS STATEMENT

This work uses publicly available video data under their original licenses and does not involve new human-subject data collection or the release of personally identifiable information. Proactive models may produce premature, missed, or incorrect responses; StreamPro is intended for research and should be used with human oversight in safety-critical or privacy-sensitive settings. We encourage users to audit the benchmark and models for bias and misuse.

REPRODUCIBILITY STATEMENT

We provide comprehensive details for reproducing our results. The benchmark construction, evaluation protocol, StreamPro-F1 metric, training objectives, reward designs, and all hyperparameters are described in the main paper and appendix. We will release StreamPro-Bench, StreamPro-SFT-63K, StreamPro-RL-3K, training/evaluation code, prompts, and configuration files.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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, 2025b. URL https://arxiv.org/abs/2502.13923.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18407–18418, 2024.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

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.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Donglin Bai, Zhibo Chen, and Ting Cao. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13448–13459, 2025.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21778–21788, 2025b.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024b.

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.

Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang W Chen. Et bench: Towards open-ended event-level video-language understanding. *Advances in Neural Information Processing Systems*, 37:32076–32110, 2024.

Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video. *arXiv preprint arXiv:2603.12938*, 2026.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, et al. Aura: Always-on understanding and real-time assistance via video streams. *arXiv preprint arXiv:2604.04184*, 2026.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18902–18913, 2025.

OpenBMB. Minicpm-o 4.5 technical report. `https://github.com/OpenBMB/MiniCPM-o/blob/main/docs/MiniCPM_o_45_technical_report.pdf`, 2026. GitHub technical report.

Jiawen Qian, Hang Du, Guoshun Nan, Shan Huang, Jiaqi Yu, Haoqing Wang, Jingfeng Chen, Mengxue Cai, Mengdi Yang, Jiarong Li, Zhilong Li, Hua Wang, Jun Liu, Xudong Jiang, and Sicong Leng. Learning to respond: A large-scale benchmark and progressive learning framework for trigger-centric online video understanding. `https://openreview.net/pdf?id=gmpnSSiJt7`, 2025a.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24045–24055, 2025b.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. *arXiv preprint arXiv:2505.05467*, 2025a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

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*, 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. *arXiv preprint arXiv:2411.17991*, 1(3):5, 2024b.

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. *arXiv preprint arXiv:2512.06810*, 2025c.

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*, 2025d.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18925–18935, 2025e.

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.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. *arXiv preprint arXiv:2512.21334*, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025.

Haolin Yang, Feilong Tang, Lingxiao Zhao, Xiang An, Ming Hu, Huifa Li, Xinlin Zhuang, Yifan Lu, Xiaofeng Zhang, Abdalla Swikir, et al. Streamagent: Towards anticipatory agents for streaming video understanding. *arXiv preprint arXiv:2508.01875*, 2025.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10807–10816, 2025.

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. *arXiv preprint arXiv:2509.24871*, 2025.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024a.

Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. Querystream: Advancing streaming video understanding with query-aware pruning and proactive response. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Kecheng Zhang, Zongxin Yang, Mingfei Han, Haihong Hao, Yunzhi Zhuge, Changlin Li, Junhan Zhao, Zhihui Li, and Xiaojun Chang. Progressive online video understanding with evidence-aligned timing and transparent decisions. *arXiv preprint arXiv:2604.18459*, 2026b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024b.

Liyun Zhu, Lei Wang, Arjun Raj, Tom Gedeon, and Chen Chen. Advancing video anomaly detection: A concise review and a new dataset. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

APPENDICES

A DETAILS OF STREAMPRO-BENCH

In this section, we first detail the data pipeline in Section A.1, which includes video collection and filtering, multi-granularity QA generation, and a multi-stage verification process to ensure high-quality annotations. We then present the evaluation protocol in Section A.2, covering the evaluation methods for both proactive and offline models, as well as the details of the StreamPro-F1 metric. All experimental results and further analysis on StreamPro-Bench are provided in Section C.2.

A.1 DATA PIPELINE

We construct StreamPro-Bench through a three-stage pipeline: video collection and filtering, data generation, and multi-stage verification.

A.1.1 VIDEO COLLECTION AND FILTERING

We collect raw videos from four sources: ET-Instruct-164k (Liu et al., 2024), LLaVA-Video-178k (Zhang et al., 2024b), MSAD (Zhu et al., 2024), and manually crawled videos from diverse online platforms. To ensure sufficient temporal dynamics for proactive QA, we employ Qwen3-VL-8B (Bai et al., 2025a) to retain videos that satisfy two criteria: (1) the number of scene transitions exceeds a predefined threshold, and (2) each scene exhibits a sufficiently long duration. This pre-filtering stage removes static or overly monotonous videos that are unsuitable for proactive QA tasks. For task-specific quality control, we further apply targeted filtering strategies: for Procedural Guidance, we enforce *step uniqueness* to ensure that procedural steps are clearly distinguishable and non-redundant; for Risk Forecasting, we apply an *egocentric* + *outdoor* filter to retain first-person outdoor videos aligned with real-world assistive navigation scenarios.

A.1.2 DATA GENERATION

For each filtered video, we construct three complementary types of captions: **(1) event-level captions** that describe high-level activities and temporal events; **(2) action-level captions** that focus on human actions and behaviors; and **(3) object-level captions** that enumerate salient objects and their attributes. Based on these multi-granularity captions, we synthesize QA pairs.

A.1.3 DATA VERIFICATION

Given that proactive streaming video understanding requires each question to be associated with multiple temporally grounded answers and precise temporal localization—an inherently strict requirement—we design a three-stage verification pipeline to ensure data quality. **(1) Basic Verification.**

Each synthesized QA pair is first subjected to a basic verification and refinement process, where we correct semantic inconsistencies and align answers with their corresponding temporal segments. **(2) Two-Agent Iterative Verification.** A Discrimination Agent evaluates each QA pair against predefined quality criteria. If a sample is deemed unsatisfactory, it returns structured rejection feedback. A Generation Agent then revises the sample accordingly. This discriminate–generate loop is executed for up to three iterations to progressively improve data quality. **(3) Human Review.** QA pairs that pass the two-agent verification stage undergo final human review to ensure correctness and consistency. All steps involving LLMs are implemented using Gemini 2.5 Pro (Comanici et al., 2025).

A.1.4 Annotation Guideline for Risk Forecasting

For Risk Forecasting task, due to its prohibitive difficulty for reliable automatic annotation, we adopt a human annotation pipeline followed by human review to ensure high-quality and consistent labels. We design the annotation protocol around warnings issued 3 seconds before an estimated accident time, aiming to simulate proactive assistance for visually impaired users from an egocentric perspective. The 3-second horizon is chosen based on consultations with visually impaired individuals, corresponding to a practical and perceptually meaningful reaction window for real-world navigation. Annotators are instructed to estimate when a potential accident would occur under the assumption of constant pedestrian velocity and trajectory. **The warning moment is defined as 3 seconds before the estimated accident time**, and annotators are required to mark this moment accordingly. After initial annotation, all labels are further reviewed and refined by two independent human to ensure annotation quality.

Risk Taxonomy. We define four categories of risks to systematically characterize navigation hazards in streaming egocentric video.

(1) **Structural Terrain**, referring to ground-level geometric irregularities that may compromise pedestrian stability (e.g., stairs, curbs, slopes).
(2) **Path Blockage**, denoting physical obstacles that directly obstruct the walking trajectory (e.g., parked bicycles, barriers).
(3) **Overhanging Objects**, representing elevated hazards that are difficult to detect using a white cane due to their height (e.g., low branches, signboards).
(4) **Functional Elements**, capturing semantically critical environmental cues necessary for navigation decisions (e.g., traffic lights, crosswalks).

A.2 Evaluation Protocol

A.2.1 Evaluation Method

For proactive models, we evaluate only those with officially released proactive inference scripts to ensure fair and reproducible evaluation (Chen et al., 2024; Wang et al., 2024b; 2025c; OpenBMB, 2026; Xia et al., 2025). Following prior evaluation protocols, videos are sampled at 1 FPS and fed into the model in a streaming manner. At each time step, the model is required to produce a response based on the currently observed content.

For offline models, we evaluate Qwen2.5-VL-7B (Bai et al., 2025b), and Qwen3-VL-8B (Bai et al., 2025a) using a decision-tree-style evaluation protocol to simulate proactive response behavior in a streaming setting. We display the evaluation prompt in Table E. Specifically, the model processes the video in a frame-by-frame manner. Before the query is issued, it passively observes the visual stream without producing any output. After the query is issued, the model follows a structured decision process at each subsequent time step. First, it assesses whether the accumulated visual context (from the beginning of the video up to the current frame) is sufficient to answer the query without speculation. If the information is insufficient, the model outputs "Wait" and continues observing. If the information is sufficient, the model then compares the currently inferred answer with its previously generated output. If the answer remains unchanged, it still outputs "Wait" to avoid redundant responses. Otherwise, it generates the updated answer. This protocol enables a systematic evaluation of whether offline models can support proactive decision-making under a streaming setting.

Table 6: Temporal definitions for optimal response times and evaluation windows across tasks.

Task	Optimal Response Time	Time Tolerance Window
Event Understanding	Within event duration	A [start time, end time + 3s] window
Object Understanding	Object state transition moment	A $[-3\text{s}, +3\text{s}]$ window centered at the transition moment
Anomaly Warning	Anomaly onset time	A $[0, +5\text{s}]$ post-onset window following anomaly onset
Temporal Perception	Within event duration	A [start time, end time + 3s] window
Temporal Grounding	Event ending time	A $[-3\text{s}, +3\text{s}]$ window centered at the event end time
Procedural Guidance	Previous step completion time	A $[-3\text{s}, 0]$ window around step completion time
Risk Forecasting	3s before the estimated accident time (warning time)	A $[-1\text{s}, +3\text{s}]$ window centered at the warning time

A.2.2 Evaluation Metric

Temporal Evaluation Metrics. To accommodate the heterogeneous temporal characteristics of different tasks, we define task-specific optimal response timings and corresponding temporal tolerance windows, as summarized in Table 6. Based on the form of the optimal response timing, we categorize tasks into two groups. For **interval-based tasks**, including *Event Understanding* and *Temporal Perception*, the optimal response is defined as an interval spanning the event duration. A response achieves full score if it falls within this interval. For **timestamp-based tasks**, i.e., all tasks except the interval-based ones, the optimal response corresponds to a specific moment (e.g., state transition, anomaly onset, or step completion). A response achieves full score if it occurs exactly at the designated timestamp.

Beyond the optimal response timing, we introduce task-dependent temporal tolerance windows to allow deviations while penalizing temporally misaligned responses. Specifically, the penalty follows a linear decay as the response time moves away from the optimal point or interval, controlled by a tolerance parameter τ. Concretely, for *Event Understanding* and *Temporal Perception*, we allow a tolerance window extending 3 seconds after the event ends, with $\tau = 4$. For *Object Understanding*, a symmetric tolerance of ± 3 seconds around the state transition moment is adopted, with $\tau = 4$. For *Anomaly Warning*, we define a post-onset tolerance window of 5 seconds, with $\tau = 6$, reflecting delayed but acceptable responses. For *Temporal Grounding*, a symmetric ± 3 second window around the event end time is used, with $\tau = 4$. For *Procedural Guidance*, the optimal response is defined as the moment when the previous step is completed. We allow responses within a $[-3, 0]$ second window relative to step completion, with $\tau = 4$. Finally, for *Risk Forecasting*, the optimal response timestamp is defined as 3 seconds before the estimated accident time, emphasizing early warning capability. We adopt an asymmetric tolerance window of $[-1, +3]$ seconds around this point. To further reflect the asymmetry between premature and delayed responses, we use different decay factors: $\tau = 2$ for early responses (before the optimal timestamp) to more strongly penalize overly early predictions, and $\tau = 4$ for delayed responses, allowing slightly greater tolerance after the optimal warning time.

In the timing score defined in Section 3.3, $\mathrm{dist}(t_{\mathrm{pred}}, t_{\mathrm{gt}})$ is the absolute difference between the predicted response time and the optimal timestamp for timestamp-based tasks. For interval-based tasks, this distance is zero within the target interval and otherwise equals the minimum temporal distance to its boundary.

Answer–Prediction Matching Design. Since both the ground-truth answer and the model prediction are represented as trajectories, establishing a one-to-one correspondence between them is a critical problem. We adopt a *prediction-first* matching strategy, where each prediction is sequentially checked against all answer time windows. If a prediction falls within the temporal window of an answer, it is considered a successful match and is not allowed to match any other answer. Under this matching scheme, a single answer may be associated with multiple candidate predictions. We then select the prediction with the highest overall score as the final score for that answer. This matching strategy is simple and efficient.

A.3 Human Preference Alignment Validation

To validate the proposed StreamPro-F1 metric and demonstrate its alignment with human judgment, we conduct a pairwise human evaluation.

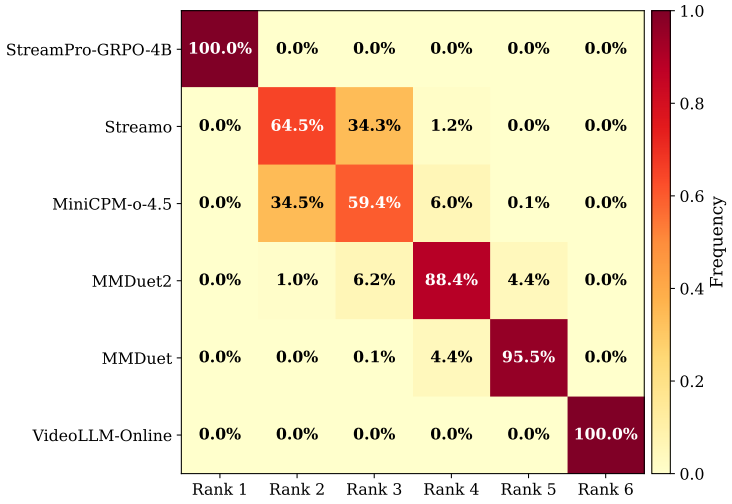

Figure 5: Bootstrap rank stability heatmap.

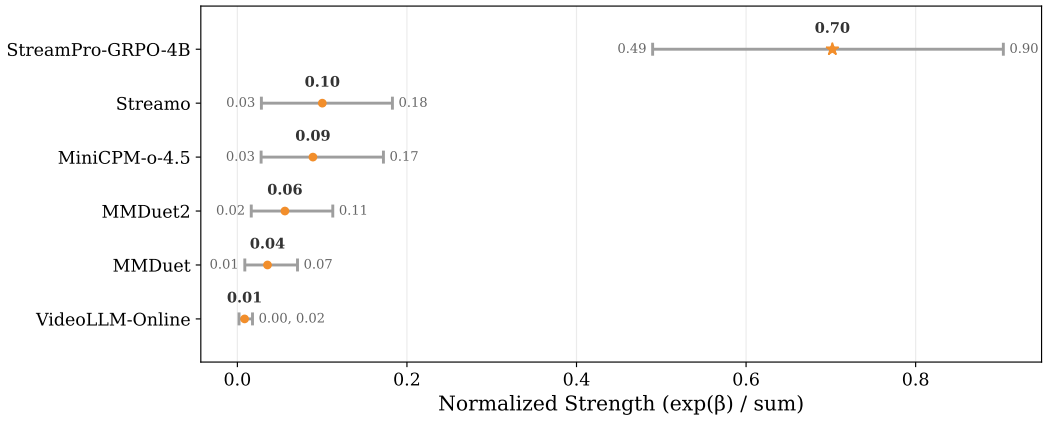

Figure 6: Estimated Bradley-Terry scores and 95% confidence intervals via bootstrap resampling.

Evaluation Methodology. Following the Chatbot Arena paradigm (Chiang et al., 2024), human annotators evaluate anonymized response trajectories from two sampled models and declare a winner or a tie. To infer global rankings from these relative preferences, we employ the Bradley-Terry (BT) model (Bradley & Terry, 1952). Given the inherent difficulty of streaming tasks, ties are frequent. We handle them via the Weighted BT approach (Chiang et al., 2024), encoding each tie as two symmetric 0.5-weighted pseudo-observations, allowing seamless likelihood optimization without discarding data.

Ranking Reliability Assessment. To quantify the statistical reliability of the derived rankings, we employ Bootstrap resampling (Efron & Tibshirani, 1994). Specifically, we generate 1,000 bootstrap samples with replacement from the collected pairwise comparisons and independently fit the Weighted BT model on each. This yields a Rank Stability Matrix (frequency of a model occupying each rank) and Score Confidence Intervals (median BT parameter with 95% CI).

Results and Analysis. As shown in the rank stability heatmap (Figure 5) and score confidence intervals (Figure 6), the evaluated models exhibit highly confident stratification. Our proposed StreamPro-GRPO-4B unambiguously secures the top rank across all bootstrap replicates. Its confidence bounds are completely detached from the rest of the cohort, demonstrating absolute stability and significant superiority in proactive streaming capabilities. While minor variance exists among closely matched middle-tier models (e.g., Streamo and MiniCPM-o-4.5), the hierarchical boundaries remain strictly defined, with VideoLLM-Online consistently anchoring the final rank.

Alignment with Benchmark Metric. Crucially, the final human preference ranking perfectly matches the ranking produced by our proposed StreamPro-F1 metric (yielding a Spearman's rank correlation $\rho = 1.0$). This absolute consensus demonstrates that our automated evaluation protocol faithfully captures human perception of proactive streaming quality, validating StreamPro-Bench as a robust and reliable evaluation standard.

B Details of StreamPro Training Framework

B.1 Details of Reinforcement Learning

Matching Algorithm for the RM Reward. Here, we detail the matching procedure for the RM reward $R_{\mathrm{RM}}(\mathcal{Y})$. We process ground truth responses g_j sequentially, considering all unmatched predictions p_i within the current temporal window and selecting the one with the highest pair score S'_{ij}. Predictions are not preassigned to ground truth windows as in benchmark evaluation. Once selected, a prediction is removed from the candidate pool for subsequent ground truth responses. Each selected pair contributes one score S'_i to Equation 6. Algorithm 1 gives the matching procedure.

It is worth noting that for tasks under the Anticipatory Assistance dimension, we modify the matching window to strictly encourage predictive behavior. Since the model is expected to make predictions before an event fully unfolds rather than answering afterwards, the temporal window is restricted to $[t_{\mathrm{gt}} - \tau, t_{\mathrm{gt}}]$. Any predictions generated after t_{gt} are not considered for matching in these tasks.

Algorithm 1 Matching for the RM Reward

Require: Ground truth responses $\mathcal{G} = \{g_j\}$, predicted responses $\mathcal{P} = \{p_i\}$, temporal tolerance τ
Ensure: RM reward $R_{\mathrm{RM}}(\mathcal{Y})$

$\mathcal{U} \leftarrow \emptyset, T \leftarrow 0$
for each ground truth $g_j \in \mathcal{G}$ **do**
  $W_j \leftarrow [g_j.t - \tau,\ g_j.t + \tau]$
  if the task is Anticipatory Assistance **then**
    $W_j \leftarrow [g_j.t - \tau,\ g_j.t]$
  end if
  $s_{\mathrm{best}} \leftarrow 0, p_{\mathrm{best}} \leftarrow \mathrm{NULL}$
  for each prediction $p_i \in \mathcal{P} \setminus \mathcal{U}$ **do**
    if $p_i.t \in W_j$ **then**
      $S_{\mathrm{time},ij} \leftarrow 1 - |p_i.t - g_j.t|/\tau$
      Evaluate $S_{\mathrm{sem},ij}$ for (p_i, g_j)
      $S'_{ij} \leftarrow (S_{\mathrm{time},ij} + S_{\mathrm{sem},ij})/2$
      if $S'_{ij} > s_{\mathrm{best}}$ **then**
        $s_{\mathrm{best}} \leftarrow S'_{ij}$
        $p_{\mathrm{best}} \leftarrow p_i$
      end if
    end if
  end for
  if $p_{\mathrm{best}} \neq \mathrm{NULL}$ **then**
    $\mathcal{U} \leftarrow \mathcal{U} \cup \{p_{\mathrm{best}}\}$
    $T \leftarrow T + s_{\mathrm{best}}$
  end if
end for
$R_{\mathrm{RM}}(\mathcal{Y}) \leftarrow \frac{2T}{|\mathcal{P}|+|\mathcal{G}|}$
return $R_{\mathrm{RM}}(\mathcal{Y})$

B.2 StreamPro-SFT-63K and StreamPro-RL-3K

We use the StreamPro-Bench data pipeline (excluding the human review step in the verification stage) to construct a 66K-scale dataset. We then further filter the data to obtain StreamPro-SFT-63K and StreamPro-RL-3K. Due to the difficulty of constructing the Risk Forecasting task, this dataset does not include Risk Forecasting samples.

In the SFT stage, our primary objective is to endow the model with fundamental proactive capabilities. The training tasks encompass Event Captioning (EC), Action Captioning (AC), Event Understanding (EU), Object Understanding (OU), Temporal Perception (TP), and Temporal Grounding (TG). In the RL stage, we further introduce more challenging proactive tasks. The training tasks encompass Event Understanding (EU), Object Understanding (OU), Anomaly Warning (AW), Temporal Perception (TP), Temporal Grounding (TG) and Procedural Guidance (PG). The task distributions of the two datasets are shown in Figure 7.

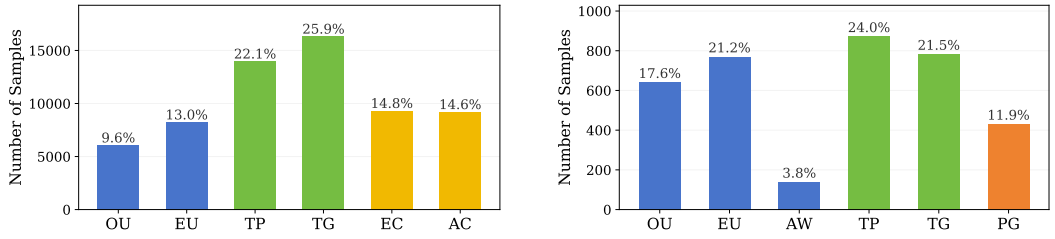

Figure 7: Dataset Statistics. **Left**: StreamPro-SFT-63K; **Right**: StreamPro-RL-3K.

C Additional Experiments

C.1 Additional Ablation Studies

C.1.1 Effectiveness of StreamPro-SFT-63K in SFT

To verify the quality of the data generated by our pipeline, we conduct an ablation study as shown in Table 7. When training only with the 429K open-source data, the model achieves 8.6 on SPB and 60.7 on OVO-RTVP. After incorporating the additional StreamPro-SFT-63K data, performance improves substantially, reaching 13.3 on SPB and 62.1 on OVO-RTVP, while also improving VideoMME from 58.5 to 60.7. These results demonstrate the effectiveness and high quality of our constructed SFT data in enhancing both proactive and real time streaming tasks.

Table 7: Ablation study on the effectiveness of StreamPro-SFT-63K data during SFT training.

Data	**SPB**	**OVO-RTVP**	**VideoMME**
Open source 429k	8.6	60.7	58.5
429k+StreamPro-SFT-63k	**13.3**	**62.1**	**60.7**

C.1.2 Impact of Matching Strategy in RL

We also ablate the matching strategy used to assign predictions to ground truth during RL. We compare our proposed *GT-first* (Optimal) matching with the *Prediction-first* (Greedy) matching directly adopted from the benchmark evaluation.

Table 8: Ablation of matching strategies for RM reward matching during GRPO.

Matching Strategy	**SPB**	**OVO-RTVP**	**VideoMME**
Prediction-first	20.70	63.77	60.26
GT-first	**25.14**	**64.18**	**60.85**

C.1.3 Sensitivity to Different LLM Judges

To assess the robustness of StreamPro-Bench to the choice of LLM judge, we evaluate StreamPro-GRPO-4B using three different judges: Gemini 2.5 Pro, GPT-5.5, and Gemini 3.1 Pro. As shown in

Table 9: Sensitivity of StreamPro-Bench to different LLM judges.

LLM Judge	Average Score
Gemini 2.5 Pro	32.8
GPT-5.5	33.4
Gemini 3.1 Pro	33.2

Table 10: Detailed Results on Perceptual Understanding.

Methods	Params	Event Understanding					Object Understanding					Anomaly Warning					Avg. F1
		Time	Sem.	P	R	F1	Time	Sem.	P	R	F1	Time	Sem.	P	R	F1	
Offline Models + Proactive Prompt																	
Qwen2.5-VL-7B (Bai et al., 2025b)	7B	8.9	8.0	0.3	4.6	0.5	15.6	19.2	19.8	11.7	11.3	1.7	21.7	0.7	1.4	0.8	4.2
Qwen3-VL-8B (Bai et al., 2025a)	8B	43.1	25.1	1.7	24.2	2.5	68.9	61.8	1.7	57.8	3.0	66.6	63.3	7.2	48.7	11.7	5.7
Open-Source Proactive Models																	
VideoLLM-Online (Chen et al., 2024)	8B	0.6	0.1	0.3	0.1	0.1	11.4	0.0	0.0	0.0	0.0	31.7	3.7	0.7	2.6	0.8	0.3
MMDuet (Wang et al., 2024b)	7B	82.1	39.4	8.4	36.0	13.0	44.0	40.7	2.7	23.2	4.8	25.5	24.1	9.1	10.4	9.4	9.1
MMDuet2 (Wang et al., 2025c)	3B	81.6	59.3	11.8	56.5	17.7	31.5	9.5	3.1	4.9	3.3	9.9	18.7	7.9	8.7	8.1	9.7
MiniCPM-o-4.5 (OpenBMB, 2026)	9B	10.9	8.0	15.5	7.7	9.8	14.4	20.9	25.9	9.6	13.9	31.0	21.7	15.3	16.9	15.8	13.2
Streamo (Xia et al., 2025)	3B	13.4	8.2	20.5	7.4	8.9	57.9	49.5	12.5	40.8	15.8	2.6	3.9	2.3	2.3	2.3	9.0
StreamPro-SFT	**3B**	86.1	58.3	11.9	56.1	18.4	48.6	54.2	8.1	41.6	12.2	20.7	19.8	9.9	12.4	10.5	13.7
StreamPro-GRPO	**3B**	75.3	57.0	42.6	53.6	<u>46.1</u>	59.1	66.7	36.5	50.9	**41.0**	32.5	30.2	17.7	18.5	18.0	<u>35.0</u>
StreamPro-SFT	**4B**	90.2	67.3	18.0	65.2	26.3	53.2	58.4	11.6	45.9	16.7	58.3	59.8	24.8	39.6	<u>28.5</u>	23.8
StreamPro-GRPO	**4B**	82.0	67.4	49.2	64.6	**54.6**	51.9	49.5	35.1	40.1	<u>36.4</u>	63.3	64.9	41.5	48.8	**43.5**	**44.8**

Table 9, the resulting average scores are 32.8, 33.4, and 33.2, respectively, with a maximum difference of only 0.6 points. The small variation across judges indicates that the evaluation results are largely insensitive to the choice of judging model, supporting the robustness of StreamPro-Bench under different LLM evaluators.

C.2 All Results on StreamPro-Bench

As shown in Table 10, Table 11 and Table 12, we present comprehensive results on StreamPro-Bench across all three evaluation dimensions and seven tasks, reporting Time Score, Semantic Score, Precision, Recall, and StreamPro-F1, where higher values indicate better performance.

Existing proactive models perform poorly on StreamPro-Bench, indicating that their capabilities are still far from those required for an ideal real-world proactive system. Among the three evaluated dimensions, they achieve relatively acceptable performance in perceptual understanding, while showing significant deficiencies in temporal reasoning and anticipatory assistance.

Specifically, compared with other proactive models, MMDuet (Wang et al., 2024b) and MMDuet2 (Wang et al., 2025c) achieve relatively high Time Score, Semantic Score, and Recall, since they tend to generate responses more frequently, which increases the probability of hitting the correct temporal window. However, due to this tendency toward over-generation, Precision is penalized and thus drops to a more moderate level. Besides, Streamo (Xia et al., 2025) demonstrates strong temporal reasoning capabilities, due to the presence of video temporal grounding tasks in its training data. Note that VideoLLM-Online (Chen et al., 2024) exhibits an inflated performance on Procedural Guidance, which is mainly caused by its tendency to produce overly generic and highly abstract responses. Such responses are often assigned relatively stable and moderate scores by the LLM-based evaluator across different cases, leading to an apparently high average score.

C.3 All Results on StreamingBench RTVU and CU

Table 13 presents the detailed results for each task in StreamingBench.

C.4 All Results on OVO-Bench RTVP

Table 14 presents the detailed results for RTVP in OVO-Bench.

Table 11: Detailed Results on Temporal Reasoning.

Methods	Params	Temporal Perception					Temporal Grounding					Avg. F1
		Time	Sem.	P	R	F1	Time	Sem.	P	R	F1	
Offline Models + Proactive Prompt												
Qwen2.5-VL-7B (Bai et al., 2025b)	7B	1.1	1.2	0.7	1.0	0.8	8.8	0.5	0.4	0.4	0.4	0.6
Qwen3-VL-8B (Bai et al., 2025a)	8B	50.8	34.8	0.9	34.3	1.6	6.8	0.3	0.1	0.2	0.1	0.9
Open-Source Proactive Models												
VideoLLM-Online (Chen et al., 2024)	8B	0.7	0.1	0.0	0.1	0.0	0.4	0.0	0.0	0.0	0.0	0.0
MMDuet (Wang et al., 2024b)	7B	91.8	40.6	1.6	38.6	3.0	53.9	2.1	0.1	1.1	0.1	1.6
MMDuet2 (Wang et al., 2025c)	3B	85.8	63.6	4.0	61.3	6.7	12.7	0.0	0.0	0.0	0.0	3.4
MiniCPM-o-4.5 (OpenBMB, 2026)	9B	1.3	0.8	0.1	0.6	0.1	3.5	0.0	0.0	0.0	0.0	0.1
Streamo (Xia et al., 2025)	3B	39.2	26.6	18.4	25.4	20.4	16.5	12.4	8.1	8.6	8.2	14.3
StreamPro-SFT	**3B**	65.8	38.7	37.2	38.6	37.6	25.8	14.9	5.4	11.3	6.4	22.0
StreamPro-GRPO	**3B**	79.7	62.8	45.7	60.3	50.3	37.6	32.0	17.4	23.6	19.2	34.7
StreamPro-SFT	**4B**	69.4	51.4	50.6	51.3	50.7	23.3	15.6	6.3	11.8	7.4	29.1
StreamPro-GRPO	**4B**	83.3	71.6	54.8	69.0	**58.8**	53.4	50.5	29.0	38.1	**31.4**	**45.1**

Table 12: Detailed Results on Anticipatory Assistance.

Methods	Params	Procedural Guidance					Risk Forecasting					Avg. F1
		Time	Sem.	P	R	F1	Time	Sem.	P	R	F1	
Offline Models + Proactive Prompt												
Qwen2.5-VL-7B (Bai et al., 2025b)	7B	9.8	9.3	7.5	5.5	5.8	5.1	2.9	9.0	2.0	3.0	4.4
Qwen3-VL-8B (Bai et al., 2025a)	8B	58.7	30.7	4.8	29.1	4.3	7.3	4.0	2.9	3.3	1.8	3.1
Open-Source Proactive Models												
VideoLLM-Online (Chen et al., 2024)	8B	20.6	10.5	47.4	10.4	**16.9**	22.4	5.5	0.9	4.8	0.7	**8.8**
MMDuet (Wang et al., 2024b)	7B	30.2	16.1	2.1	9.3	3.4	36.9	15.5	3.1	9.3	4.2	3.8
MMDuet2 (Wang et al., 2025c)	3B	40.2	15.4	6.4	12.4	6.5	14.4	3.9	0.6	2.9	1.0	3.8
MiniCPM-o-4.5 (OpenBMB, 2026)	9B	9.8	9.3	7.5	5.5	5.8	5.1	2.9	9.0	2.0	3.0	4.4
Streamo (Xia et al., 2025)	3B	17.2	7.7	15.5	7.6	8.8	1.1	0.3	0.0	0.3	0.0	4.4
StreamPro-SFT	**3B**	44.6	26.8	4.6	20.7	7.1	39.4	13.9	0.9	9.7	1.5	4.3
StreamPro-GRPO	**3B**	17.9	13.3	6.9	8.9	7.6	10.6	4.5	6.3	2.9	3.7	5.6
StreamPro-SFT	**4B**	37.2	25.6	6.8	20.9	8.8	50.1	18.2	3.2	13.0	4.3	6.5
StreamPro-GRPO	**4B**	28.0	23.0	8.6	15.1	10.6	13.6	8.2	8.9	5.6	**6.3**	8.5

Table 13: Evaluation results on StreamingBench.

Method	Param	Real-Time Visual Understanding											Contextual Understanding				Overall
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Avg.	ACU	MCU	SQA	Avg.	Avg.
Human	-	89.5	92.0	93.6	91.5	95.7	92.5	88.0	88.8	89.7	91.3	91.2	88.8	90.4	95.0	91.4	91.3
Offline Models																	
Gemini 1.5 pro (Team et al., 2024)	-	79.0	80.5	83.5	79.7	80.0	84.7	77.8	64.2	72.0	48.7	75.8	51.4	40.7	**54.8**	**49.0**	69.6
GPT-4o (Achiam et al., 2023)	-	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3	41.2	38.4	32.8	37.5	65.1
LLaVA-OneVision (Li et al., 2024a)	7B	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.4	65.7	45.1	71.2	35.6	36.0	27.3	33.0	62.4
Qwen2-VL-7B (Wang et al., 2024a)	7B	75.2	82.8	73.2	77.5	68.3	71.0	72.2	61.2	61.5	46.1	69.0	31.2	26.0	39.6	32.3	60.5
InternVL2-8B (Wang et al., 2025b)	8B	68.1	60.9	69.4	77.1	67.7	62.9	59.3	53.3	55.0	56.5	63.8	32.0	31.2	32.3	31.8	56.4
Streaming Models																	
VideoLLM-Online (Chen et al., 2024)	8B	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0	24.2	29.2	30.8	28.1	34.1
Flash-VStream (Zhang et al., 2024a)	7B	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	25.9	24.8	25.2	26.8	25.6	25.8
Dispider (Qian et al., 2025b)	7B	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.5	39.6	27.7	34.8	34.0	59.8
TimeChat-Online (Yao et al., 2025)	7B	80.2	82.0	79.5	83.3	76.1	78.5	78.7	64.6	69.6	**58.0**	75.3	39.2	31.6	41.6	37.5	66.6
ViSpeak (Fu et al., 2025b)	7B	79.8	**88.3**	83.3	81.1	76.4	75.1	70.4	65.9	77.3	34.2	74.5	38.8	36.8	44.0	39.9	66.5
StreamForest (Zeng et al., 2025)	7B	83.1	82.8	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3	-	-	-	-	-
StreamAgent (Yang et al., 2025)	7B	79.6	78.3	79.3	75.9	74.7	76.9	**82.9**	66.3	73.7	55.4	74.6	39.7	30.3	39.6	36.5	65.8
QueryStream (Zhang et al., 2026a)	7B	82.4	84.4	79.2	82.4	78.0	81.3	78.7	65.0	69.3	47.3	75.3	-	-	-	-	-
Streamo-3B† (Xia et al., 2025)	3B	83.7	80.5	83.9	84.6	80.5	81.0	71.3	67.5	71.6	37.2	75.8	43.6	38.4	41.2	41.1	67.8
StreamPro-SFT	**3B**	82.7	79.7	80.1	84.6	81.1	82.2	76.9	69.9	76.7	44.7	77.1	42.4	34.8	42.8	40.0	68.5
StreamPro-GRPO	**3B**	81.6	79.7	81.1	81.7	80.5	82.6	75.9	69.9	77.6	33.5	75.9	42.0	34.8	42.0	39.6	67.5
StreamPro-SFT	**4B**	**86.7**	78.9	**88.0**	**84.9**	83.0	**86.0**	82.4	**74.0**	**78.7**	33.0	79.3	48.8	**43.6**	48.4	46.9	71.8
StreamPro-GRPO	**4B**	85.4	77.3	87.7	84.0	**84.3**	84.4	77.8	73.2	**78.7**	50.0	**79.8**	**53.6**	43.2	43.2	46.7	**72.1**

Table 14: Evaluation results on OVO-Bench.

Method	Param	Real-Time Visual Perception						
		OCR	ACR	ATR	STU	FPD	OJR	Avg.
Human	-	94.0	92.6	94.8	92.7	91.1	94.0	93.2
Offline Models								
Gemini 1.5 Pro (Team et al., 2024)	-	**85.9**	67.0	**79.3**	**58.4**	63.4	62.0	69.3
GPT-4o (Achiam et al., 2023)	-	69.8	64.2	71.6	51.1	70.3	59.8	64.5
LLaVA-Video (Zhang et al., 2024b)	7B	69.1	58.7	68.8	49.4	74.3	59.8	63.4
LLaVA-OneVision (Li et al., 2024a)	7B	66.4	57.8	73.3	53.4	71.3	62.0	64.0
Qwen2-VL-7B (Wang et al., 2024a)	7B	60.4	50.5	56.0	47.2	66.3	55.4	56.0
InternVL2-8B (Wang et al., 2025b)	8B	67.1	60.6	63.8	46.1	68.3	56.5	60.4
Streaming Models								
VideoLLM-online (Chen et al., 2024)	8B	8.1	23.9	12.1	14.0	45.5	21.2	20.8
Flash-VStream (Zhang et al., 2024a)	7B	24.2	29.4	28.5	33.7	25.7	28.8	28.4
Dispider (Qian et al., 2025b)	7B	57.7	49.5	62.1	44.9	61.4	51.6	54.5
TimeChat-Online (Yao et al., 2025)	7B	75.2	46.8	70.7	47.8	69.3	61.4	61.9
ViSpeak (Fu et al., 2025b)	7B	75.2	58.7	71.6	51.1	74.3	66.9	66.3
StreamForest (Zeng et al., 2025)	7B	68.5	53.2	71.6	47.8	65.4	60.9	61.2
StreamAgent (Yang et al., 2025)	7B	71.2	53.2	63.6	53.9	67.3	58.7	61.3
QueryStream (Zhang et al., 2026a)	7B	75.2	49.5	69.8	50.0	71.3	62.5	63.1
Thinking-QwenVL (Zhang et al., 2026b)	7B	74.1	57.2	68.1	55.3	**75.0**	58.3	64.7
Streamo-3B† (Xia et al., 2025)	3B	73.8	51.4	74.1	47.2	57.4	63.0	61.2
Streamo-7B (Xia et al., 2025)	7B	79.2	57.8	75.0	49.4	64.4	70.1	66.0
StreamPro-SFT	**3B**	73.2	51.4	75.0	50.6	58.4	64.1	62.1
StreamPro-GRPO	**3B**	74.5	59.6	74.1	50.6	59.4	66.8	64.2
StreamPro-SFT	**4B**	79.2	**68.8**	69.0	53.9	66.3	**71.7**	68.2
StreamPro-GRPO	**4B**	82.6	67.9	73.3	55.1	68.3	71.2	**69.7**

D CASE STUDY

D.1 BENCHMARK EXAMPLES

We present additional examples of the benchmark tasks, as shown in Figure 8, Figure 9, and Figure 10.

D.2 MODEL COMPARISON

We compare the outputs of different models to comprehensively evaluate the proactive capabilities of StreamPro, as shown in Figure 11, 12, 13, and 14. StreamPro demonstrates strong performance on both object understanding and temporal grounding tasks, consistently producing accurate and well-timed responses. In contrast, for the risk forecasting examples, all existing models still struggle to achieve satisfactory results, reflecting the limited anticipatory assistance capability in this more challenging setting.

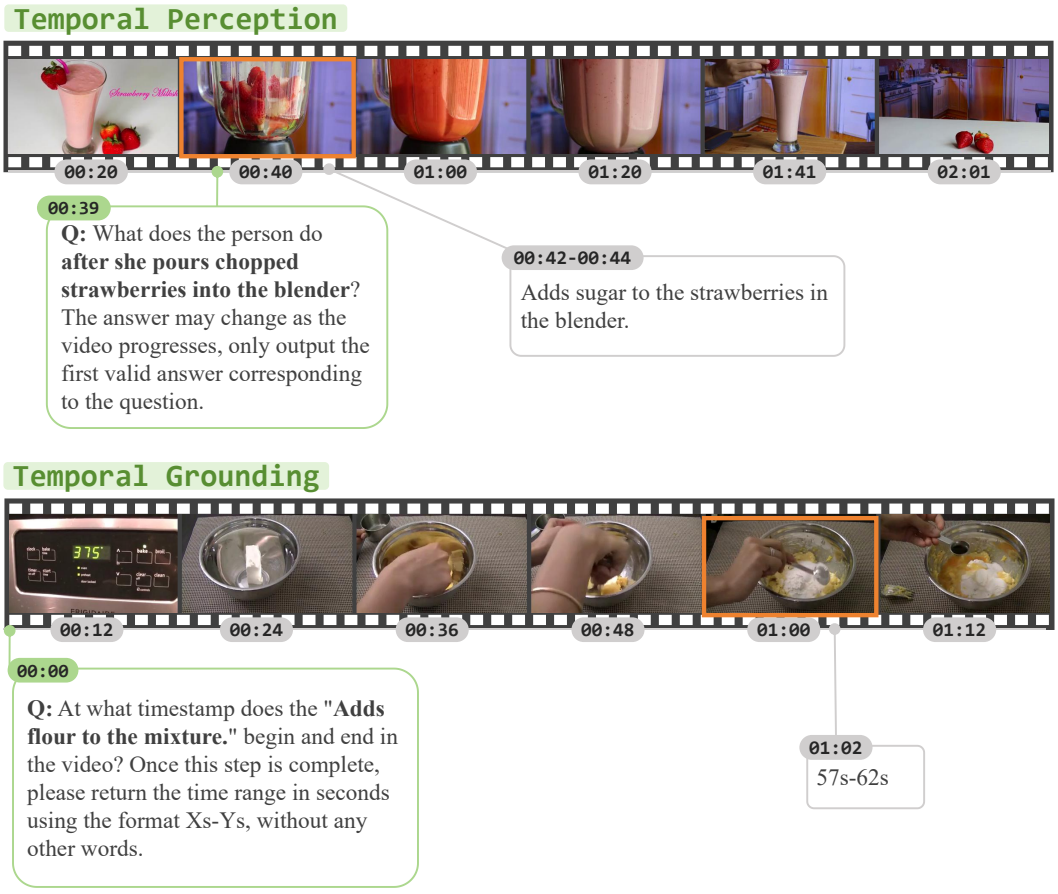

Figure 8: Benchmark samples of Temporal Reasoning tasks

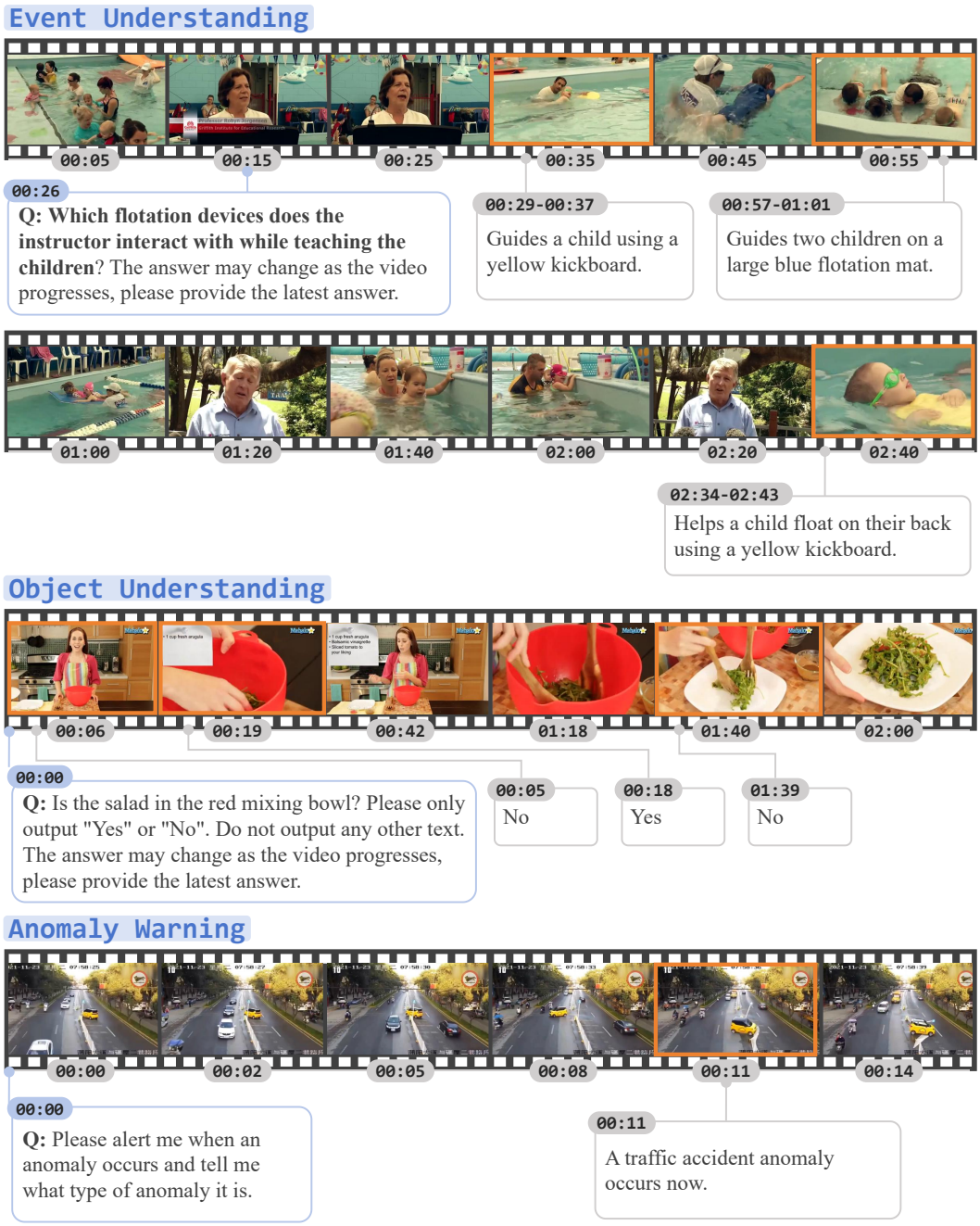

Figure 9: Benchmark samples of Perceptual Understanding tasks



Figure 10: Benchmark samples of Anticipatory Assistance tasks.

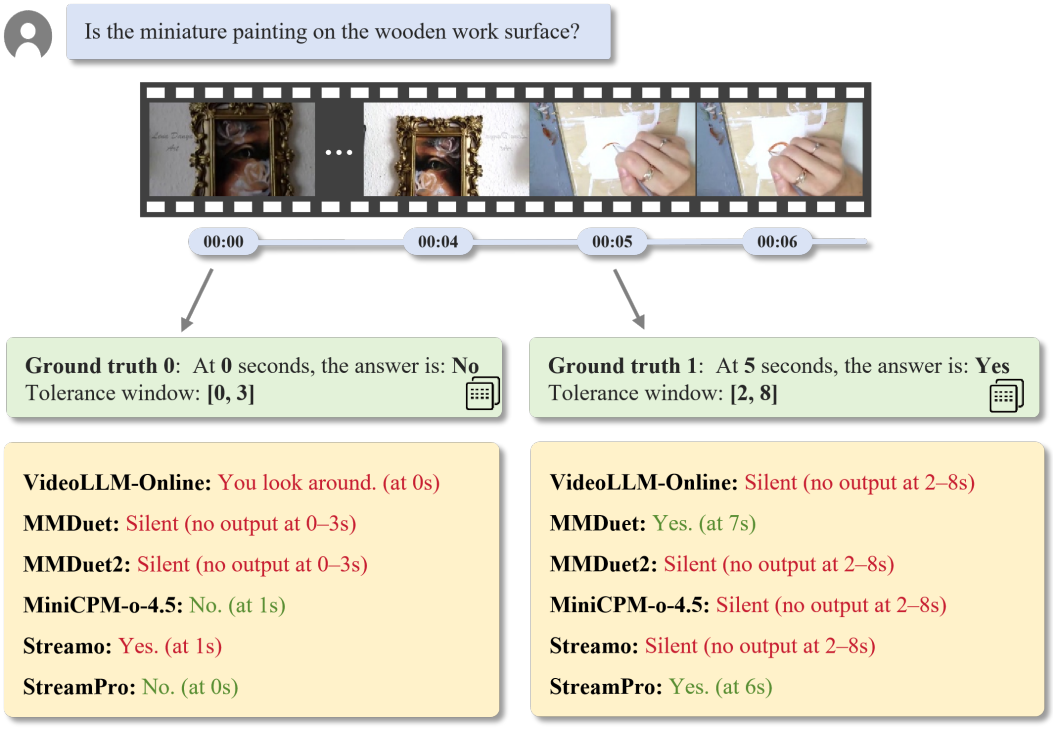

Figure 11: Comparison of different models on object understanding tasks.

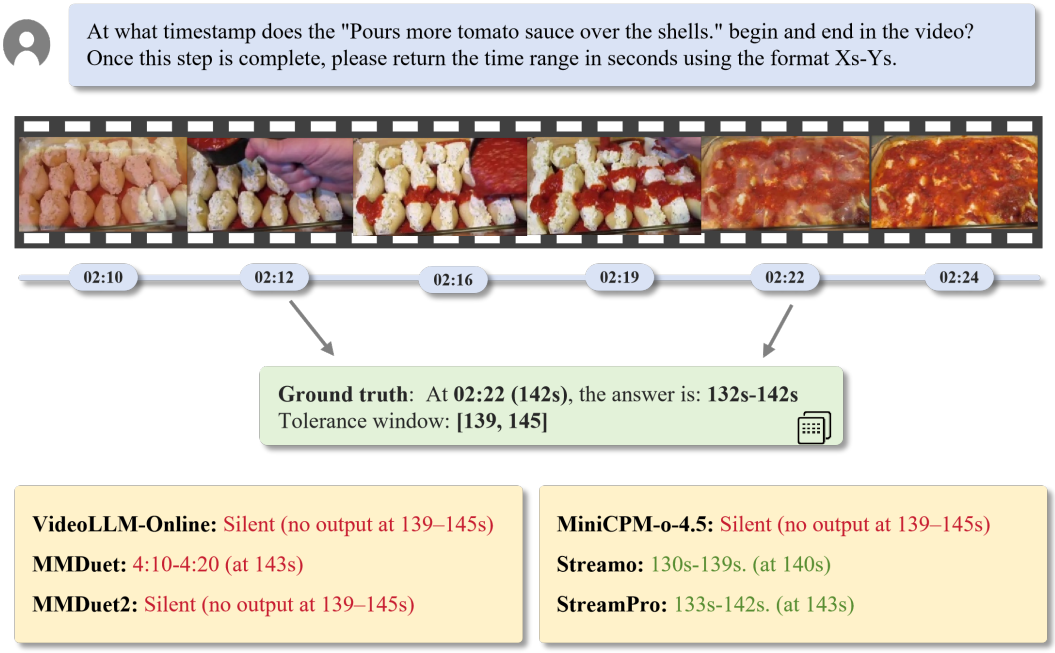

Figure 12: Comparison of different models on temporal grounding tasks.

Risk Forecasting

Monitor this first-person video as a real-time guide for a visually impaired user, providing clear and actionable warnings for any potential hazards in their path. Always give warnings 3 seconds in advance if possible.



Ground Truth:

00:09 Directly ahead a scooter is approaching on the sidewalk, please pay attention to avoid it.

01:05 Directly ahead there is a manhole cover, please beware of the uneven terrain.

01:07 Directly ahead there is a manhole cover, please beware of the uneven terrain.

01:26 Directly ahead there are bumps on the ground, please beware of the risk of falling.

01:42 Directly ahead there are low metal barriers and a traffic cone, please pay attention to avoid them.

02:31 Directly ahead there are bumps on the ground, please beware of the risk of falling.

VideoLLM-Online: You walk on the roal. (All the time)

MMDuet: Continuously generates captions.

MMDuet2: Continuously generates captions.

Streamo: Silent. (All the time)

Figure 13: Comparison of different models on risk forecasting tasks.

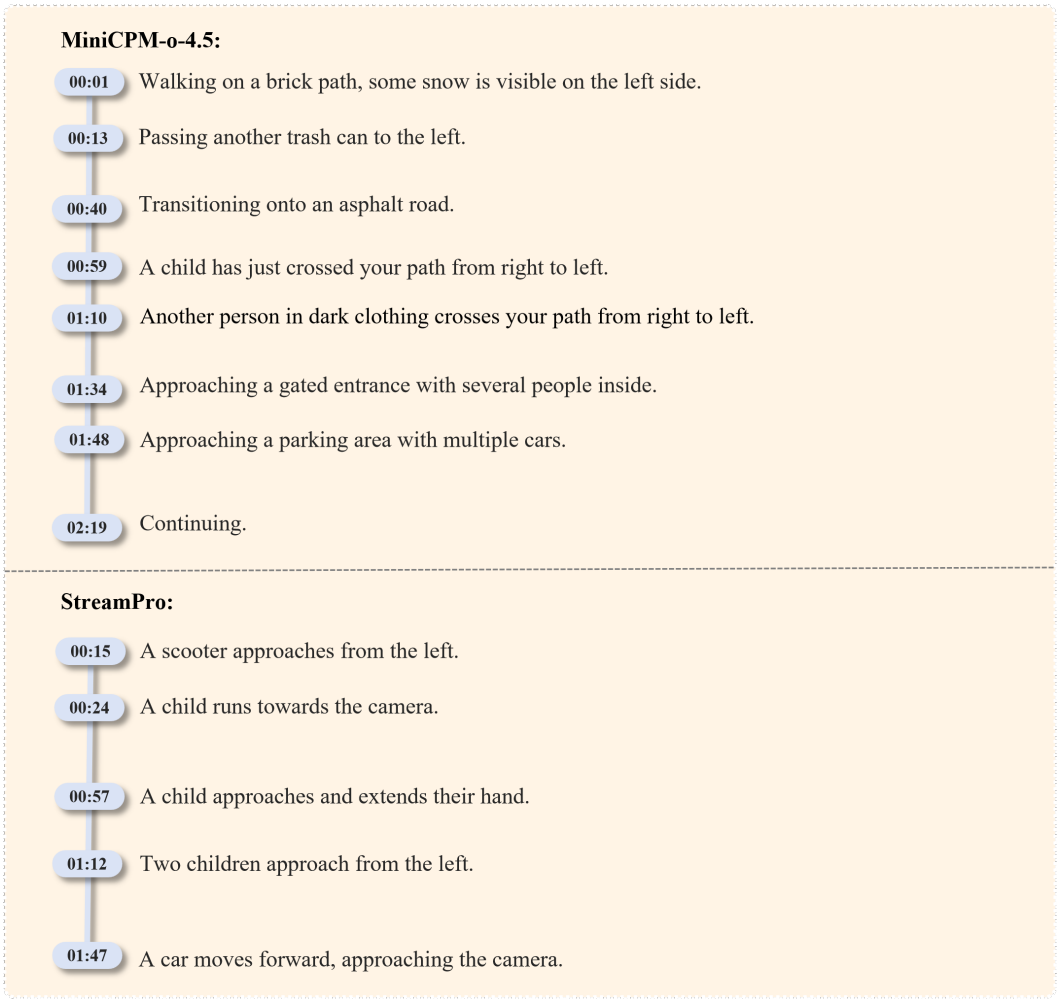

Figure 14: Comparison of different models on risk forecasting tasks.

E PROMPT DESIGN

We present the system prompt used in StreamPro framework, the prompt used for evaluating offline models, the prompt for rubric generation, and the prompt used for open-ended evaluation.

System Prompt

You are a helpful assistant specializing in streaming video analysis. You will receive input frame by frame, where each frame is labeled with absolute time intervals in the exact format `<Xs-Ys>` (e.g., `<0s-1s>`). Follow the rules below strictly: **Response Protocol:**

- Use `</Silence>` when:
  - A relevant event has not yet fully completed, **or**
  - The current input is irrelevant to the given question.
- Use `</Response>` only when:
  - An event has fully concluded, **or**
  - The available information is sufficient to fully answer the question.

  At this point, provide a complete and final answer.

Additional Constraints:
- Do not provide partial answers.
- Do not speculate beyond the given information.
- Every valid answer must begin with `</Response>`.

Offline Proactive Decision Prompt

You are a Streaming Video Analysis Expert. You will receive a video sequence frame-by-frame. A user Query will be issued once at an arbitrary time during the video stream. Before the Query is issued, observe the video carefully. After the Query is issued, follow the decision logic below.

Objective: Determine whether the visual information observed so far (from the beginning of the video up to the current moment) is sufficient to answer the user's Query.

Decision Logic: For each input frame sequence after the Query has been issued:
1. Is the visual information observed so far sufficient to answer the Query without guessing?
   - If NO → reply `Wait`.
   - If YES → proceed to Step 2.
2. Does the inferred answer differ from the last answer you provided?
   - If NO → reply `Wait`.
   - If YES (First Trigger) → reply with the actual answer.
   - If YES (Answer Update due to new evidence) → reply with the updated answer.

Output Constraint:
- Do not output anything other than `Wait` or the actual content of the answer.

Rubric Generation Prompt

You are designing evaluation rubrics for a streaming video QA system. The rubrics will be used to evaluate model trajectories AFTER generation — the model's trajectory is not available yet. You see the ground truth and the video.

Input:
- **Question:** {question}
- **Ground Truth Answer(s):** {ground_truth_text}

Task: Design 6–10 binary (pass/fail) evaluation rubrics that assess a model's streaming trajectory quality. Each rubric must be **SPECIFIC** to this particular video and question — not generic criteria.

Important Requirements:
- Each rubric must be an **ASSERTIVE statement**:
  - Use words like `must`, `shall not`, `is prohibited`.
  - Do NOT write vague descriptions (e.g., "the model should do X well").
  - Write concrete assertions (e.g., "The response must include Y — missing this is not allowed").

Rubric Dimensions (each rubric covers one):
- **Granularity:**
  - Assess whether step decomposition matches the natural structure of the video.
  - The model shall not over-split atomic actions.
  - The model shall not merge distinct actions into vague descriptions.
- **Coherence:**
  - Steps must follow correct temporal and physical logic.
  - The model shall not skip necessary intermediate steps.
- **Coverage:**
  - All essential actions or phases must be included.
  - Missing key stages is not allowed.

Post-check (self-review):
- Each rubric must tightly match the specific video and QA pair.
- No rubric should be generic or broadly applicable.
- All three dimensions must be covered.
- Total number must be between 6 and 10.
- Each rubric must be written as a concrete assertion.

Output Format: Return **ONLY** the final JSON array:

```
[
  {
    "rubric_id": 0,
    "dimension": "granularity|coherence|coverage",
    "rubric": "Assertive statement about what to check"
  }
]
```

Open-Ended Evaluation Prompt

You are an expert evaluator judging whether a model's answer provides a reasonable and factually correct response that directly addresses the question, based on the reference answer.

Evaluation Guideline:

- Focus on whether the model provides information that directly answers the question.
- The answer does not need to reproduce all details from the reference—it only needs to provide factually correct and relevant information.
- An answer that captures the essential information required by the question should be considered strong, even if it omits descriptive details.
- Accept simplified, rephrased, or high-level responses as long as they are consistent with the reference and do not contradict known facts.
- Do not deduct points for omitting secondary or illustrative details when the key information needed to answer the question is present, or for using concise or abstract phrasing.
- Only penalize if the answer is factually wrong, fails to provide relevant information, or is so vague that it does not actually answer the question.

Scoring (integer 0–5):

- 5: Fully accurate and complete answer.
- 4: Correct and sufficiently informative answer; may omit non-essential details but contains the key information.
- 3: Partially relevant but missing some important information.
- 2: Tangential or speculative without solid factual grounding.
- 1: Factually incorrect.
- 0: No attempt to answer or completely off-topic.

Output Format:

- Return a valid JSON object with exactly two keys:
  - `"explanation"`: one sentence focusing on whether the answer provides correct and relevant information for the question
  - `"score"`: an integer from 0 to 5
- Output only the JSON. No other text, markdown, or commentary.

Inputs:

- Question: {question}
- Predicted Answer: {model_output}
- Correct Answer: {reference_answer}