QWEN
BUSINESS UNIT

SyncPro: Synchronous Reasoning and Responding for Proactive Streaming Video Understanding

Tian Xueyun[1,2,3], **Bingbing Xu**[1], **Jiaming Liu**[†3], **Jinpeng Yu**[3], **Yang Shi**[3], **Yu Zhao**[1], **Ruihua Huang**[3], **Yuanzhuo Wang**[1] **and Huawei Shen**[1,2]

[1]Institute of Computing Technology, CAS, [2]University of Chinese Academy of Sciences, [3]Qwen Business Unit of Alibaba

[†]Project Leader.
Email: tianxueyun23z@ict.ac.cn.
Project Page: https://eureka-maggie.github.io/SyncPro/.

Proactive streaming video understanding requires assistants to track unfolding events and respond when a user's requested condition is met. To make these decisions, assistants must accumulate relevant evidence and reassess response conditions as new observations arrive. Existing work offers complementary capabilities for this process: proactive video assistants learn when to respond, while streaming video reasoning methods generate scheduled intermediate text to organize observations and support subsequent answers. Together, these capabilities provide a foundation for proactive reasoning, but how to coordinate ongoing reasoning with timely responses remains underexplored. In a direct adaptation, fixed reasoning intervals can trigger updates when nothing relevant has changed or postpone useful updates until the next scheduled step. Sharing one generation path also couples two distinct requirements: preserving evidence for future decisions and producing only the response requested by the user. Moreover, responses generated after reasoning must wait for it to finish, even when sufficient evidence is already available. To address these challenges, we introduce SyncPro, a synchronous dual-stream framework with specialized generation paths for evidence tracking and user-facing responses. Its *Inner* and *Response* branches share causal history and activate independently, allowing relevant evidence changes to trigger internal updates and responses to proceed concurrently with ongoing reasoning. We construct 189K dual-stream examples and train SyncPro through supervised fine-tuning followed by trajectory-level reinforcement learning. Experiments across three proactive benchmarks show improved response timing and more accurate answers to user requests. Code will be released at this repository.

1. Introduction

Proactive video assistants respond as events unfold (Qian et al., 2025; Wang et al., 2025b; Yan et al., 2026), rather than after receiving complete videos (Maaz et al., 2024; Lin et al., 2024). This requires tracking query-relevant evidence and assessing response conditions. Explicit reasoning can support this monitoring through internal updates that are not shown to the user, while responses deliver the requested information or assistance. The challenge is to coordinate these activities while accommodating their distinct roles and timing.

Two lines of work provide complementary foundations: proactive assistants learn response timing over video streams (Chen et al., 2024; Qian et al., 2025), while streaming reasoning methods generate intermediate text as observations arrive (Guan et al., 2026; Zhang et al., 2026). Recent work combines these capabilities: ThinkStream uses incremental reasoning to decide whether to continue observing or respond (Liu et al., 2026b). However, it follows a fixed schedule of reasoning at every input step, with reasoning and response decisions arranged serially in one generation sequence.

The central challenge is to coordinate ongoing reasoning with proactive responses as evidence evolves. This coordination involves three aspects. First, reasoning should adapt its timing and duration to relevant evidence (Figure 1(a)). Fixed schedules can cause redundant or delayed updates. Second, reasoning should maintain query-relevant state, while responses should fulfill the user's request (Figure 1(b)). Finally, the roles need independent activation (Figure 1(c)). Responses supported by available evidence should proceed alongside

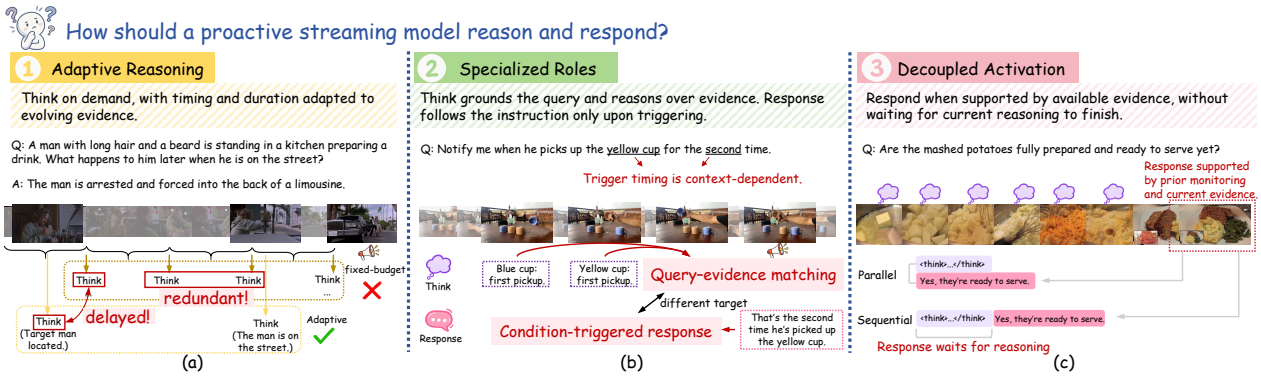


Figure 1 | **Requirements for proactive streaming reasoning. (a) Adaptive reasoning:** internal updates follow query-relevant evidence rather than a fixed schedule. **(b) Specialized roles:** reasoning supports event monitoring, while responses deliver the requested output. **(c) Decoupled activation:** responses supported by prior monitoring and current evidence need not await further reasoning.

ongoing reasoning.

We therefore introduce SyncPro, a synchronous dual-stream framework on a shared multimodal backbone. To support adaptive reasoning, each branch decides whether to generate or remain silent based on the shared history and current observation. Separate low-rank adapters specialize *Inner* for evidence tracking and *Response* for user-facing responses. Independent activation lets both branches decode concurrently without accessing each other's current outputs. Their completed outputs enter the shared history for later decisions. This design coordinates ongoing reasoning with proactive responses while preserving their distinct roles and timing.

We construct 189K dual-stream examples and train SyncPro through supervised fine-tuning and reinforcement learning. On the Proactive Output task of StreamingBench (Lin et al., 2026a), it achieves 41.2% accuracy within a two-second response window, exceeding the strongest compared baseline by 3.2 percentage points. It also outperforms prior methods on all three Forward Active Responding tasks of OVO-Bench (Niu et al., 2025), with an average F1 score of 34.19%, and obtains the highest PAUC score of 32.43% among the compared methods on the VAD subset of ProactiveVideoQA (Wang et al., 2025a). Ablations and analyses validate explicit reasoning and role separation, with specialized branches and largely preserved offline video understanding.

Our main contributions are:

- **Synchronous reasoning and responding.** We introduce a dual-stream framework for proactive streaming video understanding. Its specialized streams activate independently and jointly update a shared history for subsequent decisions.
- **Dual-stream data and training.** We provide 189K trajectories with per-step supervision for internal updates and user-facing responses. Supervised fine-tuning learns both behaviors, and trajectory-level reinforcement learning refines response timing.
- **Empirical evaluation and analysis.** Experiments on three proactive benchmarks demonstrate gains in response timing and content. Controlled comparisons support dual-stream modeling over response-only and interleaved generation.

2. Related Work

Proactive Video Understanding. Offline video models respond after observing complete videos (Maaz et al., 2024; Lin et al., 2024), whereas streaming models ground outputs in the observed video prefix (Chen et al., 2024; Huang et al., 2025). Proactive methods further regulate response timing through explicit control mechanisms or learned policies (Wang et al., 2025b; 2026b; Xueyun et al., 2026; Yan et al., 2026; Azad et al., 2026; Ma et al., 2026; Yang et al., 2026). Policy optimization incorporates class reweighting and rewards over turns and trajectories (Li et al., 2026a), while sequential control coordinates memory compression with response readiness and adaptive reasoning (Liu et al., 2026a). Both formulations retain a single user-facing output stream.

Streaming Video Reasoning. Streaming reasoning methods incorporate intermediate reasoning and responses into a single autoregressive sequence as observations arrive (Zhao et al., 2026; Guan et al., 2026; Zhang et al., 2026; Wang et al., 2026a). Accumulated reasoning traces also guide response decisions throughout the stream (Liu et al., 2026b). Such formulations often schedule reasoning at predefined processing intervals and generate responses after the associated reasoning segment. This couples response generation to the timing and completion of reasoning.

Concurrent Multimodal Interaction. Existing work introduces concurrency at different levels. At the output level, parallel generation of semantic and acoustic tokens produces a single spoken response (Xu et al., 2025a;b). Input–output duplex models extend this overlap to perception during response generation while retaining a single semantic output stream (Lin et al., 2026b; Cui et al., 2026). Beyond perception and generation, concurrency also supports internal coordination. Reasoning and interaction modules communicate asynchronously (Huang et al., 2026), while text agents use parallel role-specific streams to solve a shared task (Su et al., 2026). SyncPro instead separates ongoing event monitoring from condition-triggered task execution. The two streams share causal history but activate independently without accessing each other's outputs at the current step.

3. Method

3.1. Task Formulation

Let q be a user instruction specifying a monitoring objective and the required response. A video arrives as a causal sequence $\mathcal{V} = (v_1, \ldots, v_T)$. Each observation v_t defines micro-turn t. Under our 1-fps experimental protocol, each v_t is one sampled video frame. The model receives $c_t = (q, \mathcal{H}_{t-1}, v_t)$, where $\mathcal{H}_{t-1}$ contains information retained from previous micro-turns. Future observations are unavailable during generation.

We model internal reasoning and user-facing responses as two generation branches. *Inner* (I) tracks query-relevant evidence, while *Response* (R) produces the requested output. At each micro-turn, each branch chooses whether to remain silent or generate content:

$$a_t^I \in \mathcal{A}_I = \{\langle\text{silent}\rangle, \langle\text{inner}\rangle\}, \qquad a_t^R \in \mathcal{A}_R = \{\langle\text{silent}\rangle, \langle\text{speak}\rangle\}. \tag{1}$$

Let o_t^I and o_t^R denote the complete output sequences of the two branches. Each branch attends only to c_t and its own preceding tokens. The outputs follow branch-specific conditional distributions:

$$o_t^I \sim p_I(\cdot \mid c_t), \qquad o_t^R \sim p_R(\cdot \mid c_t), \tag{2}$$

Each output begins with an action token. A ⟨silent⟩ action terminates the sequence immediately. A ⟨inner⟩ or ⟨speak⟩ action is followed by branch-specific autoregressive content. Both branches may activate together when the current observation requires state tracking and a user-facing response. Only active *Response* content is returned to the user during interaction.

The two streams synchronize at micro-turn boundaries. Once both finish, their complete outputs jointly update the shared context for the next observation:

$$\mathcal{H}_t = U(\mathcal{H}_{t-1}, o_t^I, o_t^R). \tag{3}$$

Here U incorporates both output sequences and their layer-specific attention states. The visual tokens of v_t are then discarded. Both outputs remain in the history and inform later decisions.

3.2. SyncPro Model Architecture

We propose SyncPro to coordinate ongoing reasoning with timely responses. As shown in Figure 2, it combines a shared multimodal backbone with two specialized branches. *Inner* maintains evidence for later decisions, while *Response* delivers the requested output. Independent branch activation allows supported responses to proceed alongside ongoing reasoning.

At each micro-turn, the backbone encodes c_t once for both branches. Separate low-rank adaptation (LoRA) modules in each decoder layer's MLP projections parameterize their action and content distributions (Hu et al.,

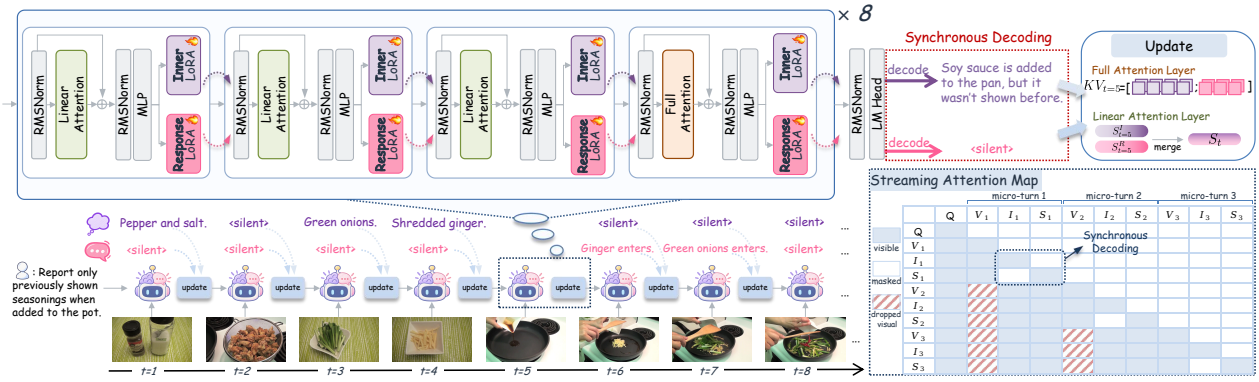


Figure 2 | **Dual-stream modeling in SyncPro.** At each micro-turn, *Inner* and *Response* select actions independently and decode from the same multimodal prefix through separate LoRA routes. Only *Response* is user-visible. Both outputs update the context for the next micro-turn before the visual tokens are discarded to limit persistent cache growth.

2022), implementing Equation 2 with independent decoding states. After generation, SyncPro appends o_t^I then o_t^R to the shared history and merges their layer states.

For a full-attention layer, let (K_t^0, V_t^0) denote the shared-prefix KV cache and (K_t^b, V_t^b) the cache increment for branch b's independently decoded action and content tokens. Concatenating these caches retains both outputs for the next micro-turn:

$$K_t = [K_t^0; K_t^I; K_t^R], \qquad V_t = [V_t^0; V_t^I; V_t^R], \tag{4}$$

where semicolons denote sequence concatenation. The block-causal mask enforces same-prefix decoding without same-turn cross-branch attention. Following prior position-sharing designs (Chen et al., 2025; Xueyun et al., 2026), both increments use the first MRoPE position after the prefix to mark a common temporal origin. Appendix A details the mask and position rules.

Qwen3.5 inherits Qwen3-Next's hybrid decoder, whose linear-attention layers use Gated DeltaNet (Qwen Team, 2026; Yang et al., 2025). In each such layer, the shared prefix yields recurrent state S_t^0, and independent decoding yields S_t^b for branch b. Appendix A gives the inherited recurrence. We define $\Delta S_t^b = S_t^b - S_t^0$ and merge both branch updates:

$$S_t = S_t^0 + \gamma \left(\Delta S_t^I + \Delta S_t^R\right), \tag{5}$$

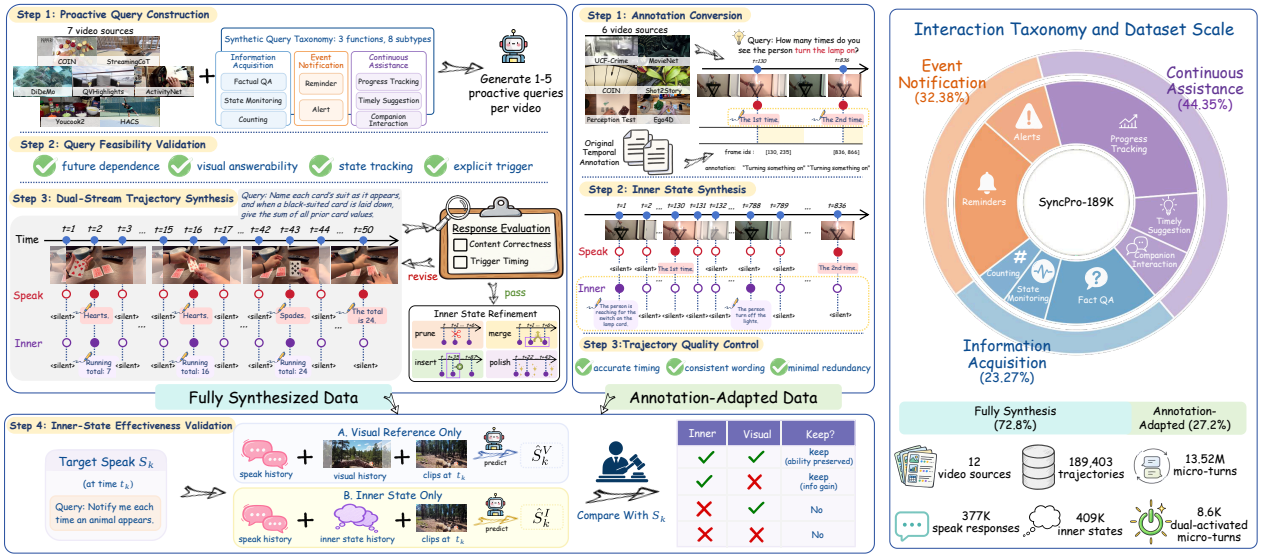
With $\gamma = 0.5$, Equation 5 reduces to $S_t = (S_t^I + S_t^R)/2$. This symmetric heuristic assigns neither semantic role prior weight and is invariant to branch order. The transition generally differs from serial composition. Using the same rule during training and inference lets the LoRA routes learn subsequent token distributions under the merged recurrent states. The concatenated text and merged layer states together instantiate the history-update operator U in Equation 3.

Streaming inference. We encode each frame once and fork the shared state for parallel branch decoding. Only active *Response* content is streamed to the user. We then merge both branch states and discard frame tokens before the next frame. Appendix A gives implementation details.

3.3. Dual-Stream Data Construction

Each training example pairs a query and video with a dual-stream trajectory. We label each branch's action at every micro-turn and provide text targets for active branches. Existing datasets lack these paired targets (Hu et al., 2025; Guan et al., 2026; Liu et al., 2026b). We construct these trajectories through proactive synthesis and temporal-label adaptation (Figure 3a).

The first route builds proactive queries from seven video sources (Krishna et al., 2017; Lei et al., 2021; Tang et al., 2019; Zhou et al., 2018; Zhao et al., 2019; Hendricks et al., 2017; Hu et al., 2025). We retain queries requiring future evidence, state tracking, and an observable trigger. Each target at t is generated from the



(a) Construction pipeline. (b) Corpus composition.

Figure 3 | **Construction of SyncPro-189K.** Two routes produce causal dual-stream trajectories. A synthesized trajectory is retained only when its accumulated *Inner* entries suffice to reconstruct every later response from restricted visual context. The resulting 189,403 trajectories span three interaction functions, eight subtypes, and 12 video sources.

physically truncated prefix $\mathcal{V}_{\leq t}$, while other micro-turns receive silent actions. An offline evaluator audits the full video for content, timing, and event coverage. We allow up to three revisions under the corresponding prefix, then remove redundant entries, insert missing event-boundary states, and normalize language. Synthesized *Inner* and *Response* entries occupy distinct micro-turns, with the final *Inner* update preceding its *Response*. This avoids repeating trigger evidence already preserved by *Response* in shared history.

The second route maps temporal annotations from six datasets to this format (Tang et al., 2019; Han et al., 2025; Grauman et al., 2022; Huang et al., 2020; Pătrăucean et al., 2023; Sultani et al., 2018). Event and answer intervals determine response timing and content. Query-conditioned generation inserts *Inner* at relevant state changes and preserves coincident boundaries when a state change matches response onset. We use only official training splits, with video IDs disjoint from the held-out ProactiveVideoQA test videos (Wang et al., 2025a).

We evaluate synthesized *Inner* updates by reconstructing responses under two contexts. The visual-reference context contains prior responses and all frames since the preceding response. The inner-state context contains prior *Inner* and *Response* entries but only the current visual window. An independent semantic judge evaluates both reconstructions. We retain trajectories only when the inner-state context reconstructs every response after earlier visual evidence is removed.

The corpus contains 189,403 trajectories from 12 sources across three functions and eight subtypes (Figure 3b). Its 137,827 synthesized and 51,576 adapted examples span 13.52M micro-turns, with 409K *Inner* entries, 377K *Response* outputs, and 8.6K joint activations preserved from coincident boundaries in the adaptation route. These cases supervise simultaneous branch activation under the same model interface used throughout training and inference.

3.4. Two-Stage Training

We first use supervised fine-tuning (SFT) on SyncPro-189K to teach the model when and how to reason and respond. We then use trajectory-level reinforcement learning (RL) to improve response timing across streaming trajectories.

3.4.1. Supervised Fine-Tuning

At each micro-turn, cross-entropy over the branch-specific spaces $\mathcal{A}_I$ and $\mathcal{A}_R$ defines $\mathcal{L}_{\text{action}}$ over the decisions in Equation 1 (Chen et al., 2024). Subsequent tokens from active branches define $\mathcal{L}_{\text{content}}$. To address label imbalance, we upweight ⟨inner⟩ and ⟨speak⟩ separately and downweight all but the first ⟨silent⟩ action in each silent span (Yao et al., 2026).

The action loss scores each micro-turn independently, although correct timing requires the response position to dominate nearby silent positions. Action Temporal Ranking (ATR) therefore imposes a local margin at each response boundary (Burges et al., 2005). Let $z^R_{t,a}$ be the *Response* logit for action a at time t, and let $r_t = z^R_{t,\langle\text{speak}\rangle} - z^R_{t,\langle\text{silent}\rangle}$ be its response preference. For a ground-truth response time t, let $\mathcal{N}_K(t)$ contain up to K adjacent micro-turns labeled ⟨silent⟩. We select the hardest local negative $u_t = \arg\max_{u\in\mathcal{N}_K(t)} r_u$ and optimize

$$\mathcal{L}_{\text{ATR}} = \frac{1}{|\mathcal{P}|}\sum_{t\in\mathcal{P}} \text{softplus}\big(\delta + r_{u_t} - r_t\big), \tag{6}$$

where $\mathcal{P}$ contains response times with silent neighbors and δ is the margin. ATR encourages the annotated response preference to exceed that of nearby silent micro-turns. The full objective is

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{action}} + \mathcal{L}_{\text{content}} + \lambda_{\text{ATR}}\mathcal{L}_{\text{ATR}}. \tag{7}$$

Only the two LoRA adapters and action-token parameters are updated. Appendix A specifies the adapter configuration, action weights, and ATR hyperparameters.

3.4.2. Trajectory-Level Reinforcement Learning

Adaptive response timing requires decisions under the model's evolving history. SFT and ATR supervise actions under annotated histories, while inference conditions on the model's own outputs. Early timing errors can therefore affect later decisions. Trajectory-level RL trains on complete causal rollouts and optimizes event coverage while penalizing unmatched responses. SFT provides content supervision, and we evaluate post-RL content quality separately in Table 3.

For trajectory τ, we match predicted response times one-to-one with ordered ground-truth windows. Matched predictions are true positives. Unmatched predictions and uncovered windows are false positives and false negatives, respectively. We use their hard-window F1 as the trajectory reward:

$$R(\tau) = F_1^{\text{hard}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{8}$$

We uniformly sample four groups: single-response trajectories with short or long post-response silence, and multi-response trajectories with sparse or dense events. This balances response frequency and silence duration. For each query, we sample G trajectories $\{\tau_i\}_{i=1}^G$ and standardize their rewards within the group to obtain relative advantages for policy optimization:

$$\widehat{A}_i = \frac{R(\tau_i) - \overline{R}}{\text{std}(R(\tau_1), \ldots, R(\tau_G)) + \epsilon_A}. \tag{9}$$

Here $\overline{R}$ is the group mean and $\epsilon_A > 0$ stabilizes normalization. For token $y_{i,j}$ generated from causal branch history $h_{i,j}$, define the policy ratio $\rho_{i,j}(\theta) = \pi_\theta(y_{i,j} \mid h_{i,j})/\pi_{\text{old}}(y_{i,j} \mid h_{i,j})$. We optimize the Group Relative Policy Optimization (GRPO) objective (Shao et al., 2024)

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{L_i}\sum_{j=1}^{L_i}\min\Big(\rho_{i,j}(\theta)\widehat{A}_i, \text{clip}\big(\rho_{i,j}(\theta), 1-\epsilon, 1+\epsilon\big)\,\widehat{A}_i\Big), \tag{10}$$

where L_i counts the tokens generated by both branches and ϵ is the clipping radius. The terminal timing advantage weights every generated token in the paired rollout.

Table 1 | **Results on StreamingBench and ProactiveVideoQA.** StreamingBench's Proactive Output (PO) uses Acc@2s with a two-second tolerance. ProactiveVideoQA subsets use PAUC ($\omega = 0.5$) to evaluate timing and content. Values are percentages.

Method	StreamingBench	ProactiveVideoQA		
	PO	WEB	EGO	VAD
Streaming models				
VideoLLM-online (Chen et al., 2024)	4.00	25.90	25.00	25.00
MMDuet (Wang et al., 2025b)	29.44	38.90	<u>46.00</u>	27.40
MMDuet2 (Wang et al., 2026b)	34.60	**53.30**	33.60	28.90
Em-Garde (Zheng et al., 2026)	38.00	44.30	**52.30**	27.40
VideoChat3 (Li et al., 2026b)	–	28.40	28.10	25.10
Reasoning-enabled models				
VST (Guan et al., 2026)	19.60	21.11	24.97	25.08
ThinkStream (Liu et al., 2026b)	27.20	30.19	24.47	18.17
Ours				
SyncPro-RL	**41.20**	<u>48.28</u>	33.74	**32.43**
SyncPro-SFT	<u>38.80</u>	47.83	33.89	<u>32.14</u>
Interleaved	34.00	29.89	25.00	25.91
Response-only	22.80	27.98	27.98	31.17

Best and second-best results are **bold** and <u>underlined</u>. Streaming-model scores follow Em-Garde, except MMDuet's PO and VideoChat3's PVQA results from their papers. We evaluate VST and ThinkStream at 2 fps using released code and original protocols.

4. Experiments

4.1. Experimental Setup

Training and inference. We initialize SyncPro from Qwen3.5-4B and train SyncPro-SFT on SyncPro-189K (Qwen Team, 2026). RL makes one pass over 800 unique examples balanced across the four behavior groups. For the proactive evaluations, SyncPro-RL, SyncPro-SFT, *Interleaved*, and *Response-only* run at 1 fps. Appendix A provides the training, decoding, and baseline configurations used in these controlled comparisons.

Comparison methods. We compare with VideoLLM-online, Flash-VStream, Dispider, StreamForest, MMDuet, MMDuet2, Em-Garde, and VideoChat3 (Chen et al., 2024; Zhang et al., 2025; Qian et al., 2025; Zeng et al., 2025; Wang et al., 2025b; 2026b; Zheng et al., 2026; Li et al., 2026b). Scores follow Em-Garde, except MMDuet's PO and VideoChat3's PVQA results from their papers. We evaluate VST and ThinkStream at 2 fps with their released implementations and protocols.

Benchmarks and metrics. We follow the official frame-indexed protocols (Zheng et al., 2026). StreamingBench-PO Acc@2s evaluates timing, while Joint@Ks also requires containment-matched content under the same predictions and timing rule (Lin et al., 2026a). OVO-FAR covers repetition counting (REC), sequential steps (SSR), and clue sufficiency (CRR). It reports precision, recall, and F1 within two seconds (Niu et al., 2025). ProactiveVideoQA PAUC ($\omega = 0.5$) integrates GPT-4.1 content scores over valid WEB, EGO, and VAD response windows (Wang et al., 2025a; OpenAI, 2025). Appendix C provides full results.

4.2. Main Results

StreamingBench Proactive Output. SyncPro-RL reaches 41.2% Acc@2s, exceeding the highest reported baseline by 3.2 points (Table 1). Trajectory-level RL improves on SyncPro-SFT's 38.8% by 2.4 points. Table 3 additionally evaluates joint timing and content accuracy.

OVO-Bench Forward Active Responding. SyncPro-RL achieves 34.19 mean F1, exceeding the highest reported baseline by 3.20 points, with SyncPro-SFT second at 32.37 (Table 2). It leads external baselines on all three tasks, including 50.00 F1 on REC, 6.84 points above Em-Garde. RL improves SSR and REC but reduces CRR F1, yielding a higher overall mean.

Table 2 | **OVO-Bench Forward Active Responding.** Precision, recall, and F1 use a 2 s tolerance. Avg. F1 is the task mean. Values are percentages. Best/second-best values are **bold**/underlined.

Method	CRR (P/R/F1)	SSR (P/R/F1)	REC (P/R/F1)	Avg. F1
Streaming models				
VideoLLM-online (Chen et al., 2024)	2.08/2.08/2.08	6.88/11.07/8.49	16.46/7.43/10.24	6.93
Flash-VStream (Zhang et al., 2025)	5.21/6.25/5.68	9.52/2.46/3.91	7.49/3.46/4.73	4.77
StreamForest (Zeng et al., 2025)	2.08/2.08/2.08	13.69/8.02/10.11	28.90/30.49/29.67	13.95
MMDuet2 (Wang et al., 2026b)	8.47/47.92/14.40	–	**42.72**/19.33/26.62	20.51
Em-Garde (Zheng et al., 2026)	18.22/47.92/26.40	14.87/54.88/23.40	39.66/47.33/43.16	30.99
Reasoning-enabled models				
VST (Guan et al., 2026)	4.65/18.75/7.46	3.80/2.86/3.26	9.63/**95.00**/17.48	9.40
ThinkStream (Liu et al., 2026b)	3.09/18.75/5.30	3.65/49.60/6.80	31.48/23.01/26.58	12.89
Ours				
SyncPro-RL	16.99/**64.58**/26.91	**17.50**/48.17/**25.67**	35.43/84.95/**50.00**	**34.19**
SyncPro-SFT	18.09/62.50/28.06	13.14/**57.06**/21.36	33.12/85.14/47.69	32.37
Interleaved	**23.18**/41.97/29.86	2.38/1.59/1.90	0.52/2.08/0.83	10.87
Response-only	20.93/61.75/**31.26**	4.74/47.22/8.61	2.31/18.75/4.11	14.66

ProactiveVideoQA. SyncPro-RL leads anomaly-triggered VAD at 32.43 PAUC, with SyncPro-SFT second at 32.14 (Table 1). It also ranks second on WEB at 48.28 and obtains 33.74 on EGO. EGO's long egocentric procedures require sustained tracking of fine-grained events and repeated updates (Wang et al., 2025a), which our timing-focused RL objective addresses indirectly.

4.3. Formulation Comparisons

We compare three supervised formulations on StreamingBench-PO with the same backbone, source trajectories, and decoding budget. *Response-only* retains only *Response* targets. *Interleaved* uses one LoRA route to select ⟨silent⟩, ⟨inner⟩, or ⟨speak⟩ at each micro-turn. It retains earlier *Inner* updates but only the *Response* target at joint activations. SyncPro-SFT uses independent branch actions and specialized routes. Table 3 compares these formulations. Section 4.4 examines update timing, role specialization, and synchronous decoding.

Table 3 | **Formulation comparisons on StreamingBench-PO.** Content metrics use containment or exact matching. @Response evaluates samples with a response. Values are percentages.

Method	Content Acc.	Exact Content Acc.	Content Acc. @Response	Exact Content Acc. @Response	Joint @1s	Joint @2s
SyncPro-RL	**44.0**	**25.6**	79.14	**46.04**	**23.2**	**33.6**
SyncPro-SFT	40.0	22.8	**79.37**	45.24	20.8	30.8
Interleaved	39.6	6.0	59.28	8.98	10.0	17.2
Response-only	24.0	10.0	50.85	21.19	6.4	10.4

Comparison with response-only generation. Removing internal-update targets lowers both joint accuracy and conditional content accuracy. Relative to *Response-only*, SyncPro-SFT raises Joint@2s from 10.4% to 30.8% and Content Acc.@Response from 50.85% to 79.37%. These gains support internal updates for retaining task-relevant information that later responses can use.

Interleaved streaming reasoning. *Interleaved* preserves internal updates but couples the two roles through one activation sequence. It reaches 59.28% Content Acc.@Response but only 8.98% Exact Content Acc.@Response. SyncPro-SFT raises exact accuracy to 45.24%, supporting independent branch decisions and specialized routes for generating complete responses.

Effect of trajectory-level reinforcement learning. RL refines the SFT policy using timing rewards. Joint@2s rises from 30.8% to 33.6%. Content Acc.@Response remains stable at 79.14%, indicating preserved content quality under timing-focused optimization.

Table 4 | **General video understanding.**

Method	MVBench	EgoSchema	Video-MME
Qwen3.5-4B	**65.42**	**61.60**	**67.41**
SyncPro-RL	63.48	59.60	66.30
SyncPro-SFT	<u>63.55</u>	59.60	66.41
Interleaved	60.82	<u>60.40</u>	64.78
Response-only	61.40	59.60	<u>66.74</u>

Table 5 | **Route specificity.**

Branch	Assigned	Swapped	ΔNLL↑
Inner	1.121	1.827	**0.706**
Response	0.672	2.250	**1.578**

Table 6 | **Decoding latency.**

Metric	Time saved	Speedup
Both branches finish	114.74 ms	1.3706×
First response token	199.52 ms	2.2152×

Scheduled reasoning. Scheduled uses fixed *Inner* triggers while *Response* decides autonomously. Figure 4 shows repeated text patterns and redundant *Inner* updates after the requested response. Adaptive activation can skip redundant updates and focus reasoning on task-relevant changes.



Figure 4 | **A streaming rollout with scheduled *Inner* activation.**

4.4. Dual-Stream Mechanisms and Capability Retention

Temporal alignment of *Inner* updates. From step 500 to 1000, *Inner* activations on 50 rollouts fall from 113 to 93 while exact matches to annotated updates rise from 53 to 56 (Figure 5). At step 1000, random update times with the same activation count yield 14.3 exact matches on average, versus 56 for *Inner*. This supports timing selectivity beyond activation frequency (Appendix D).

Role specialization. We swap adapters on 1,870 held-out SFT examples. Reference-token negative log-likelihood rises by 0.706 for *Inner* and 1.578 for *Response* (Table 5). Assigned adapters better model their own targets, supporting distinct roles for internal updates and responses.

Validation of synchronous decoding. Concurrent and *Inner*-first decoding use the same model, branch conditionals, and attention masks. Outputs agree on all 1,870 held-out examples, preserving generation behavior on these examples. Table 6 additionally reports latency for the two execution orders. Appendix F illustrates selective updates and separate or joint branch activation.

General video understanding. Across three offline benchmarks (Li et al., 2024; Mangalam et al., 2023; Fu et al., 2025), both SyncPro variants remain within 2.0 points of Qwen3.5-4B and retain more MVBench accuracy than *Interleaved* and *Response-only* after proactive adaptation (Table 4). The learned streaming policies thus largely preserve general video understanding.

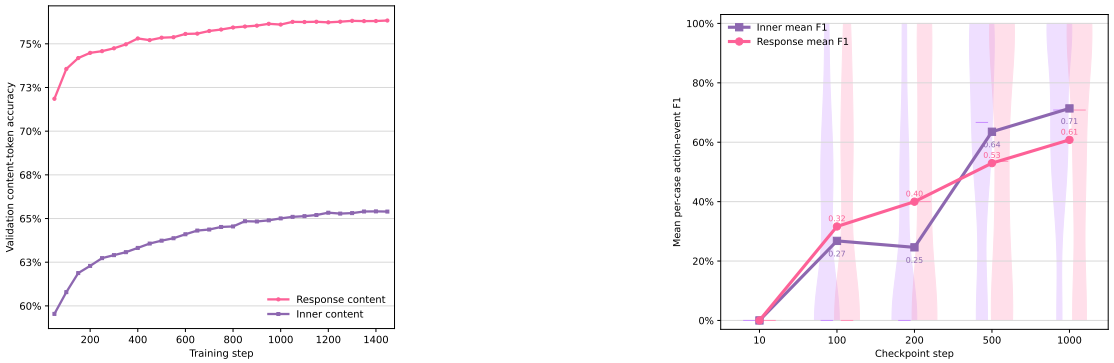


Figure 5 | **SFT learning dynamics.** Left: validation content-token accuracy over training. Right: mean action-event F1 on 50 evaluated rollouts at five checkpoints.

5. Conclusion

We introduced SyncPro to coordinate ongoing reasoning with timely responses in streaming video. Its specialized *Inner* and *Response* streams share causal history on one backbone, activate independently, and decode concurrently. We constructed 189K dual-stream examples and combined supervised fine-tuning with reinforcement learning. Across three benchmarks, SyncPro improves response timing and content while largely preserving offline video understanding. These results support explicit reasoning and role separation for proactive video assistants.

AI use statement

Gemini 3.1 Pro Preview (Google, 2026), Qwen-VL-Max-Latest (Alibaba Cloud, 2024), and GPT-5.4 (OpenAI, 2026) were used for the synthetic-data pipeline described in Section 3.3 and Appendix E. Their roles included proactive-query generation, dual-stream trajectory synthesis and revision, response reconstruction, and semantic judging. AI tools were also used for language editing. The authors designed the pipeline and conducted the experiments. They verified the reported claims, results, and citations, and retain full responsibility for all scientific decisions and the final manuscript.

Ethics statement

This work collects no new human-subject data. All experiments use established public research datasets under their released terms. We identify no additional ethical concerns beyond those associated with the source datasets and proactive video systems. Deployment in privacy-sensitive settings requires appropriate consent, access control, and bias assessment.

Reproducibility statement

Section 3 specifies the model, objectives, and data pipeline. Section 4 defines the evaluation protocol, and Appendix A provides implementation details. Abridged construction prompts appear in Appendix E. We will release the complete prompts, data, training and evaluation code, configurations, and model checkpoints required to reproduce the reported experiments.

References

Alibaba Cloud. Qwen-VL series model price reduction notice, December 2024. URL https://help.aliyun.com/en/model-studio/qwen-vl-model-billing-notice.

Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. StreamReady: Learning what to answer and when in long streaming videos, 2026. URL https://arxiv.org/abs/2603.08620.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005. doi: 10.1145/1102351.1102363. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2005/08/icml_ranking.pdf.

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. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_VideoLLM-online_Online_Video_Large_Language_Model_for_Streaming_Video_CVPR_2024_paper.html.

Keyu Chen, Zhifeng Shen, Daohai Yu, Haoqian Wu, Wei Wen, Jianfeng He, Ruizhi Qiao, and Xing Sun. ASPD: Unlocking adaptive serial-parallel decoding by exploring intrinsic parallelism in LLMs, 2025. URL https://arxiv.org/abs/2508.08895.

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.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. FlexAttention: A programming model for generating fused attention variants. In *Proceedings of the Eighth Conference on Machine Learning and Systems*, 2025. URL https://openreview.net/forum?id=2QMYV4bA0R.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. 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, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Google. Gemini 3.1 Pro Preview, February 2026. URL https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Grauman_Ego4D_Around_the_World_in_3000_Hours_of_Egocentric_Video_CVPR_2022_paper.html.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously. *arXiv preprint arXiv:2603.12262*, 2026.

Mingfei Han, Linjie Yang, Xiaojun Chang, Lina Yao, and Heng Wang. Shot2Story: A new benchmark for comprehensive understanding of multi-shot videos. In *International Conference on Learning Representations*, volume 2025, pp. 41665–41677, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/672d794a6052b6beab0e2e002204d974-Abstract-Conference.html.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5803–5812, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Hendricks_Localizing_Moments_in_ICCV_2017_paper.html.

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. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yuhang Hu, Zhenyu Yang, Shihan Wang, Shengsheng Qian, Bin Wen, Fan Yang, Tingting Gao, and Changsheng Xu. StreamingCoT: A dataset for temporal dynamics and multimodal chain-of-thought reasoning in streaming VideoQA. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 13464–13470, 2025. doi: 10.1145/3746027.3758311. URL https://arxiv.org/abs/2510.25332.

Muye Huang, Lingling Zhang, Xingyu Yu, Lei Shi, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, and Jun Liu. Duplexomni: Real-time listening, seeing, thinking, and speaking for full-duplex interaction. *arXiv preprint arXiv:2606.09186*, 2026.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. MovieNet: A holistic dataset for movie understanding. In *Proceedings of the European Conference on Computer Vision*, 2020. doi: 10.1007/978-3-030-58548-8_41. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/2710_ECCV_2020_paper.php.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3328–3338,

2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Huang_Online_Video_Understanding_OVBench_and_VideoChat-Online_CVPR_2025_paper.html.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 706–715, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Krishna_Dense-Captioning_Events_in_ICCV_2017_paper.html.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/62e0973455fd26eb03e91d5741a4a3bb-Abstract.html.

Ao Li, Zihan Xiao, Zihao Yue, Boshen Xu, Linli Yao, Jiaze Li, Pei Fu, Jianzhong Ju, Jian Luan, and Qin Jin. StreamPro: From reactive perception to proactive decision-making in streaming video, 2026a. URL https://arxiv.org/abs/2605.16381.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_MVBench_A_Comprehensive_Multi-modal_Video_Understanding_Benchmark_CVPR_2024_paper.html.

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, Yuhao Dong, Haoning Wu, Zhiqiu Zhang, Yuandong Yang, Changlian Ma, Qingyu Zhang, Yansong Shi, Xinyu Chen, Haoran Chen, Zizheng Huang, Jun Zhang, Kun Ouyang, Lin Sui, Ziang Yan, Yicheng Xu, Chenting Wang, Yinan He, Hongjie Zhang, Yi Wang, Yu Qiao, Yali Wang, Ziwei Liu, Kai Chen, and Limin Wang. VideoChat3: Fully open video MLLM for efficient and generalist video understanding. *arXiv preprint arXiv:2607.14935*, 2026b. URL https://arxiv.org/abs/2607.14935.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.342. URL https://aclanthology.org/2024.emnlp-main.342/.

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, 2026a.

Junyan Lin, Junlong Tong, Hao Wu, Jialiang Zhang, Jinming Liu, Xin Jin, and Xiaoyu Shen. Speak while watching: Unleashing true real-time video understanding capability of multimodal large language models. *arXiv preprint arXiv:2601.06843*, 2026b.

Jinming Liu, Jianguo Huang, Zhaoyang Jia, Jiahao Li, Xiaoyi Zhang, Zongyu Guo, Bin Li, Wenjun Zeng, Yan Lu, and Xin Jin. An efficient streaming video understanding framework with agentic control, 2026a. URL https://arxiv.org/abs/2605.17921.

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*, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ke Ma, Jiaqi Tang, Bin Guo, Xueting Han, Ruonan Xu, Qingfeng He, Ziheng Wang, Xu Wang, Qifeng Chen, Zhiwen Yu, and Yunhao Liu. Response-G1: Explicit scene graph modeling for proactive streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. URL https://aclanthology.org/2026.acl-long.2042/.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602. Association for

Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.679. URL https://aclanthology.org/2024.acl-long.679/.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46212–46244, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/90ce332aff156b910b002ce4e6880dec-Abstract-Datasets_and_Benchmarks.html.

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.

OpenAI. Introducing GPT-4.1 in the API, April 2025. URL https://openai.com/index/gpt-4-1/.

OpenAI. Introducing GPT-5.4, March 2026. URL https://openai.com/index/introducing-gpt-5-4/.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and Joao Carreira. Perception test: A diagnostic benchmark for multimodal video models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42748–42761. Curran Associates, Inc., 2023. doi: 10.52202/075280-1852. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8540fba4abdc7f9f7a7b1cc6cd60e409-Abstract-Datasets_and_Benchmarks.html.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24045–24055, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Qian_Dispider_Enabling_Video_LLMs_with_Active_Real-Time_Interaction_via_Disentangled_CVPR_2025_paper.html.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020. doi: 10.1109/SC41405.2020.00024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Guinan Su, Yanwu Yang, Xueyan Li, and Jonas Geiping. Multi-Stream LLMs: Unblocking language models with parallel streams of thoughts, inputs and outputs, 2026. URL https://arxiv.org/abs/2605.12460.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00678. URL https://www.crcv.ucf.edu/research/real-world-anomaly-detection-in-surveillance-videos/.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1207–1216, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Tang_COIN_A_Large-Scale_Dataset_for_Comprehensive_Instructional_Video_Analysis_CVPR_2019_paper.html.

Lu Wang, Zhuoran Jin, Yupu Hao, Yubo Chen, Kang Liu, Yulong Ao, and Jun Zhao. Think while watching: Online streaming segment-level memory for multi-turn video reasoning in multimodal large language models. *arXiv preprint arXiv:2603.11896*, 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*, 2025a.

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 *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6338–6359. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.findings-emnlp.336. URL https://aclanthology.org/2025.findings-emnlp.336/.

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*, pp. 23257–23270, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/281970ef377580ab9ec3c7d3781110d4-Abstract-Conference.html.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025a. URL https://arxiv.org/abs/2503.20215.

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*, 2025b.

Tian Xueyun, Wei Li, Bingbing Xu, Heng Dong, Yuanzhuo Wang, and Huawei Shen. ROMA: Real-time omni-multimodal assistant with interactive streaming understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 23018–23039. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1153. URL https://aclanthology.org/2026.findings-acl.1153/.

Weicai Yan, Yuhong Dai, Qi Ran, Haodong Li, Wang Lin, Tao Jin, Xing Xie, Hao Liao, and Jianxun Lian. ProactVL: A proactive VideoLLM for real-time AI companions. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62067.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Zhenyu Yang, Kairui Zhang, Bing Wang, Shengsheng Qian, and Changsheng Xu. LiveStarPro: Proactive streaming video understanding with hierarchical memory for long-horizon streams, 2026. URL https://arxiv.org/abs/2606.17798.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Haowen Hou, Zheming Liang, Congcong Wang, Yuhang Cao, Shenglong Ye, Shuai Xie, Shuhuan Gu, Haoyang Huang, Qingyi Si, Nan Duan, and Jiaqi Wang. JoyAI-VL-Interaction: Real-time vision-language interaction intelligence. *arXiv preprint arXiv:2606.14777*, 2026. URL https://arxiv.org/abs/2606.14777.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. StreamForest: Efficient online video understanding with persistent event memory. In *Advances in Neural Information Processing Systems*, volume 38, pp. 75804–75835, 2025. doi: 10.52202/085713-2547. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6dd91fec726dbed8915a1fbadd91d1d2-Abstract-Conference.html.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-VStream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21059–21069, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_Flash-VStream_Efficient_Real-Time_Understanding_for_Long_Video_Streams_ICCV_2025_paper.html.

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

Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. HACS: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8668–8678, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Zhao_HACS_Human_Action_Clips_and_Segments_Dataset_for_Recognition_and_ICCV_2019_paper.html.

Zicheng Zhao, Kangyu Wang, Shijie Li, Rui Qian, Weiyao Lin, and Huabin Liu. CogStream: Context-guided streaming video question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16): 13332–13341, 2026. doi: 10.1609/aaai.v40i16.38336. URL https://ojs.aaai.org/index.php/AAAI/article/view/38336.

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.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):7590–7598, 2018. doi: 10.1609/aaai.v32i1.12342.

A. Implementation Details

Supervised fine-tuning. The vision encoder and shared Qwen3.5-4B backbone remain frozen. Each branch uses a LoRA adapter of rank 64, scaling factor 128, and dropout 0.05. We additionally optimize the vocabulary parameters for the new action tokens. Training uses AdamW with learning rate 10^{-4} and global batch size 128 (Loshchilov & Hutter, 2019). The action-token weights are $w_{\text{inner}} = 2.5$ and $w_{\text{speak}} = 3.0$. Within each consecutive silent span, the first silent action has weight 1.0 and later silent actions have weight 0.2. ATR uses $\lambda_{\text{ATR}} = 0.05$, margin $\delta = 0.5$, and neighborhood size $K = 4$. We train for one epoch using bfloat16 and DeepSpeed ZeRO-2 for distributed optimization (Rajbhandari et al., 2020).

SFT learning dynamics. SFT improves both content generation and action prediction (Figure 5). Validation content-token accuracy rises by 5.86 points for *Inner* and 4.48 for *Response*. Their mean per-case action-event F1 reaches 0.71 and 0.61 on the 50 evaluated rollouts.

Instantiation on the Qwen3.5 backbone. Qwen3.5 inherits Qwen3-Next's hybrid decoder, whose linear-attention layers use Gated DeltaNet (Qwen Team, 2026; Yang et al., 2025). At position j, the recurrent state follows

$$\widetilde{S}_j = \alpha_j \widetilde{S}_{j-1} \left(I - \beta_j k_j k_j^\top \right) + \beta_j v_j k_j^\top, \qquad h_j = \widetilde{S}_j q_j, \tag{11}$$

where q_j, k_j, and v_j are the query, key, and value vectors. The data-dependent coefficients $\alpha_j = \exp(g_j) \in (0, 1)$ and $\beta_j = \sigma(b_j) \in (0, 1)$ control decay and write strength.

At each micro-turn, the teacher-forced *Inner* and *Response* targets occupy one physical sequence in that order. A block-causal mask lets each branch attend to the shared prefix and its own earlier tokens, while blocking attention between the two current branches. Both completed outputs enter the state exposed to the next micro-turn. Full-attention layers construct the two KV increments under this mask. Linear-attention layers fork the shared recurrent state, process both targets independently, and apply Equation 5 before the next micro-turn. Training and inference use the same token order, mask, position IDs, and state transition. Inference replaces teacher-forced targets with tokens generated autoregressively by each branch.

State merging and positions. Both branches represent simultaneous decisions from one shared prefix. We set $\gamma = 0.5$ as a symmetric heuristic that assigns neither role prior weight. The resulting mean $(S_t^I + S_t^R)/2$ is invariant to branch order and generally differs from either serial composition. Both LoRA routes are trained under this rule, which is applied unchanged at inference. For three-axis MRoPE, both branches start at the first position after the prefix to encode a common temporal origin, following position-sharing conventions for parallel decoding and aligned multimodal streams (Chen et al., 2025; Xueyun et al., 2026). The next micro-turn advances by $\max(|o_t^I|, |o_t^R|)$, placing its first position after the longer branch span.

Merge-weight parameterization. We compare two merge-weight settings under the same SFT configuration (Table 7). Fixed uses $\gamma = 0.5$ and corresponds to the reported SyncPro-SFT model. Learned per branch optimizes separate coefficients for *Inner* and *Response*.

Effect of merge weights. Learning separate branch weights reduces StreamingBench-PO Acc@2s from 38.8% to 38.0% and OVO-FAR mean F1 from 32.37% to 30.71% (Table 7). The OVO-FAR results show a task-dependent trade-off: REC F1 increases by 3.14 points, while CRR and SSR F1 decrease by 5.45 and 2.66 points. The REC gain comes from higher precision, whereas both precision and recall decline on CRR and SSR. Fixed symmetric merging thus performs better overall than independent branch weighting under the evaluated SFT configuration.

Reinforcement learning. RL starts from the SFT checkpoint and updates only the two branch-specific LoRA adapters. The reward contains only the timing term. In pilot runs, adding a content term increased response frequency and verbosity without improving conditional content accuracy. We use AdamW with FP32 master weights, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, learning rate 10^{-6}, zero weight decay, and gradient clipping at 1.0. GRPO uses a clipping radius of 0.2 and a zero KL coefficient. RL makes one pass over 800 unique examples, with 200 from each of four behavior groups: single-response trajectories with short or long post-response silence, and multi-response trajectories with sparse or dense response events. Each of the 50 updates samples 16 queries without replacement, four per group, and draws $G = 4$ rollouts per query, yielding 64 trajectories. The reported model is the checkpoint after the final update.

Table 7 | **State-merge weights.** (a) StreamingBench-PO Acc@2s. (b) OVO-FAR precision, recall, and F1 within two seconds. Fixed uses $\gamma = 0.5$. Learned per branch optimizes separate coefficients for *Inner* and *Response*. Values are percentages.

(a) StreamingBench-PO.

Merge weights	Acc@2s
Fixed	38.8
Learned per branch	38.0

(b) OVO-FAR.

Merge weights	CRR (P/R/F1)	SSR (P/R/F1)	REC (P/R/F1)	Avg. F1
Fixed	18.09/62.50/28.06	13.14/57.06/21.36	33.12/85.14/47.69	32.37
Learned per branch	13.91/60.42/22.61	11.51/49.84/18.70	36.32/84.63/50.83	30.71

Streaming inference. We implement a Qwen3.5 streaming runtime with Hugging Face Transformers and PyTorch FlexAttention (Qwen Team, 2026; Dong et al., 2025). Videos are sampled at 1 fps, and each frame is encoded once into at most 128 visual tokens. At each micro-turn, the runtime forks the shared state, applies both LoRA routes, and batches their autoregressive decoding. Both branches use greedy decoding with a limit of 256 generated tokens per branch and micro-turn. Their states are merged before the next frame, while only active *Response* content is streamed to the user during decoding.

For adaptive inference, action tokens determine activation directly. No response threshold is tuned on the evaluation sets. Benchmark decisions are indexed by sampled frames, which the runner processes in order without dropping or coalescing. The runtime reuses the KV cache and evicts frame tokens after their state update. Full-attention KV memory still grows with retained action and text tokens. Evicting frame tokens removes the dominant per-frame contribution and limits persistent growth to compact branch outputs retained across micro-turns.

Scheduling semantics. The scheduling comparison executes one fixed SyncPro model in two ways. Concurrent execution batches both branches from the shared state. The *Inner*-first schedule completes *Inner* before decoding *Response* from the same state, without introducing cross-branch conditioning. The two schedules therefore preserve branch outputs and the merged transition while changing only execution order. Time to the first *Response* token includes preceding *Inner* computation under the sequential schedule. Task scores remain frame-indexed, while Appendix B reports wall-clock service times for one stream under the stated runtime configuration.

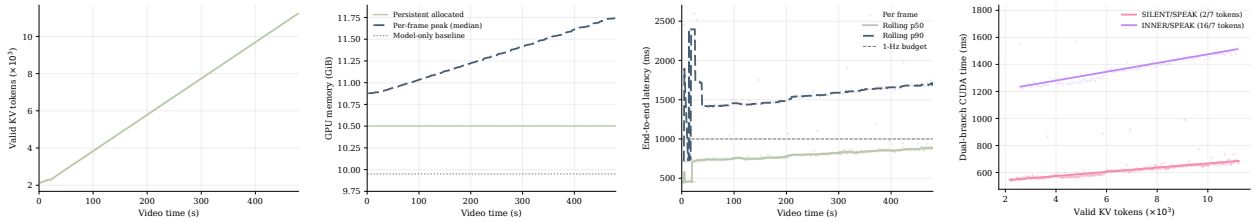
Adapted baselines. For the proactive comparisons in Tables 1 and 2, we evaluate VST and ThinkStream from their released implementations and checkpoints at 2 fps, following each method's inference protocol (Guan et al., 2026; Liu et al., 2026b). Their emitted answers are treated as response events, while steps without an answer remain silent. The broader task results in Tables 8 and 9 use the listed frame rates. Other streaming-model values follow the Em-Garde compilation (Zheng et al., 2026), except MMDuet's StreamingBench-PO and VideoChat3's ProactiveVideoQA scores from their original reports (Wang et al., 2025b; Li et al., 2026b). Their frame rates and model sizes follow the reported sources. Controlled formulation comparisons are confined to our four variants under the common 1-fps protocol and matched decoding budget.

Data provenance and generative tools. All examples in SyncPro-189K are constructed from the official training splits of the source datasets, while StreamingBench, OVO-Bench, and ProactiveVideoQA are reserved for evaluation. For the UCF-Crime source also used by ProactiveVideoQA, the training and held-out test videos have disjoint IDs (Wang et al., 2025a). To preserve the streaming setting, each *Inner* or *Response* target at timestamp t is generated from the query, preceding trajectory entries, and a 1-fps sample of $\mathcal{V}_{\leq t}$. The complete video is used only for trajectory-level auditing of event coverage and trigger timing, and candidates selected for refinement are regenerated under the same prefix constraint. Gemini 3.1 Pro Preview is used for proactive query generation and judging, Qwen-VL-Max-Latest for dual-stream trajectory synthesis, and GPT-5.4 for targeted refinement. Appendix E details the interfaces and acceptance criteria for each stage.

B. Long-Stream Runtime Profile

We profile SyncPro-SFT in BF16 over 480 seconds of a StreamingBench video sampled at 1 fps. The source resolution is 1920×1080. Each sampled frame is resized to 480×256 and encoded as 120 visual tokens. The run uses one video stream, greedy decoding with at most 64 tokens per branch, a $W = 1$ step mask, and a

16,384-token preallocated streaming cache. The mask evicts prior visual KV while retaining the system prompt, query, action tokens, and generated text. End-to-end step time includes image preprocessing, visual encoding, shared prefill, batched dual-branch decoding, state merging, and cache eviction. It excludes disk I/O and video decoding. We omit the first two steps affected by compilation. The trace measures per-step service time and excludes queueing across concurrent streams, isolating the execution cost of one video stream.



(a) Retained KV length. (b) GPU memory. (c) End-to-end latency. (d) Branch latency.

Figure 6 | **Long-stream memory and latency of dual-branch inference.** We profile a 480-s stream at 1 fps, excluding the first two compilation-affected steps. Fixed-capacity cache allocation keeps persistent memory stable, while the growing retained prefix increases transient peak memory and decoding cost. Panel (d) fixes branch output lengths to isolate how retained KV context length affects decoding latency for each branch activation pattern.

Figure 6 shows that retained full-attention KV grows from 2,126 to 11,220 tokens. Persistent allocation changes by only 1.1 MiB, while the per-frame peak rises by 0.88 GiB to 11.75 GiB. Across 478 measured steps, mean and median end-to-end latency are 985.3 and 827.2 ms. The p90 is 1,596.7 ms, and 74.9% of frames meet the 1-s budget. Panel 6d fixes branch output lengths within each action pair. Dual-branch CUDA latency increases by 15.6 and 32.4 ms per 1k retained KV tokens when only *Response* is active and when both branches are active, respectively. In this workload, cache preallocation stabilizes persistent allocation, while retained text increases transient memory and long-stream latency over the full 480-s stream.

C. Detailed Benchmark Results

Tables 8 and 9 provide task-level results and frame metadata for the broader benchmark suites (Lin et al., 2026a; Niu et al., 2025). Streaming baselines use the compilation of Em-Garde, while VST and ThinkStream are our 2-fps evaluations (Zheng et al., 2026; Guan et al., 2026; Liu et al., 2026b).

On StreamingBench, RL improves SFT on six of ten tasks. The largest gains occur on EU and CR, at 1.89 and 1.56 points, while the other four task scores remain unchanged. The resulting overall score rises from 70.44 to 70.84 after trajectory-level optimization.

On OVO-Bench, both variants attain the strongest online FPD score of 76.24, while SyncPro-SFT leads online models on ASI at 68.24. RL maintains average accuracy on the Real-Time Visual Perception subset (61.17 vs. 61.13 for SFT) and improves STU by 1.12 points. OCR, ACR, FPD, and OJR remain unchanged. The backward-tracing average shifts slightly from 43.95 to 43.62.

Table 8 | **StreamingBench Real-Time Visual Understanding subset.** Bold and underlined values mark the best and second-best results among open-source online models.

Method	Size	Frames	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Overall
Human	–	–	89.50	92.00	93.60	91.50	95.70	92.50	88.00	88.80	89.70	91.30	91.50
Proprietary models													
Gemini 1.5 Pro	–	1 fps	79.00	80.50	83.50	79.70	80.00	84.70	77.80	64.20	72.00	48.70	75.70
GPT-4o	–	64	77.10	80.50	83.90	76.50	70.20	83.80	66.70	62.20	69.10	49.20	73.30
Claude 3.5 Sonnet	–	20	73.30	80.50	84.10	82.00	75.40	79.50	61.10	61.80	69.30	43.10	72.40
Offline models													
Video-LLaMA2	7B	32	55.90	55.50	57.40	58.20	52.80	43.60	39.80	42.70	45.60	35.20	49.50
VILA-1.5	8B	14	53.70	49.20	71.00	56.90	53.40	53.90	54.60	48.80	50.10	17.60	52.30
Video-CCAM	14B	96	56.40	57.80	65.30	62.80	64.60	51.40	42.60	48.00	49.60	31.60	54.00
LongVA	7B	128	70.00	63.30	61.20	70.90	62.70	59.50	61.10	53.70	54.70	34.70	60.00
InternVL2	8B	16	68.10	60.90	69.40	77.10	67.70	62.90	59.30	53.30	55.00	56.50	63.70
Kangaroo	7B	64	71.10	84.40	70.70	73.20	67.10	61.70	56.50	55.70	62.00	38.90	64.60
LLaVA-NeXT-Video	32B	64	78.20	70.30	73.80	76.80	63.40	69.80	57.40	56.10	64.30	38.90	67.00
MiniCPM-V2.6	8B	32	71.90	71.10	77.90	75.80	64.60	65.70	70.40	56.10	62.30	53.40	67.40
LLaVA-OneVision	7B	32	80.40	74.20	76.00	80.70	72.70	71.70	67.60	65.50	65.70	45.10	71.10
Qwen2.5-VL	7B	1 fps	78.30	80.50	78.90	80.50	76.70	78.50	79.60	63.40	66.20	53.20	73.70
Streaming models													
Flash-VStream	7B	–	25.90	43.60	24.90	23.90	27.30	13.10	18.50	25.20	23.90	48.70	23.20
VideoLLM-online	8B	2 fps	39.10	40.10	34.50	31.10	46.00	32.40	31.50	34.20	42.50	27.90	36.00
Dispider	7B	1 fps	74.90	75.50	74.10	73.10	74.40	59.90	76.10	62.90	62.20	45.80	67.60
StreamAgent	7B	1 fps	79.60	78.30	79.20	75.90	74.70	76.90	82.90	66.30	73.70	**55.40**	74.30
StreamForest	7B	1 fps	83.10	**82.80**	82.70	84.30	77.50	78.20	76.90	69.10	**75.60**	54.40	77.30
Em-Garde	7B	2 fps	80.40	74.20	**89.90**	84.00	**78.90**	82.20	**87.00**	71.10	68.80	42.50	76.70
Reasoning-enabled models													
VST (Guan et al., 2026)	7B	2 fps	**85.40**	82.00	86.40	**89.10**	74.20	**87.20**	82.40	**73.10**	73.90	47.30	**79.50**
ThinkStream (Liu et al., 2026b)	3B	2 fps	83.74	70.31	76.03	82.69	76.73	78.19	76.85	70.73	74.43	45.21	75.00
Ours													
SyncPro-RL	4B	1 fps	66.94	81.25	77.29	80.77	70.44	76.32	75.00	60.16	67.90	52.13	70.84
SyncPro-SFT	4B	1 fps	66.67	79.69	77.29	80.45	68.55	76.32	75.00	59.76	67.90	51.06	70.44

Table 9 | **OVO-Bench task accuracy.** Bold and underlined values mark the best and second-best results among open-source online models.

Method	Size	Frames	Real-Time Visual Perception							Backward Tracing			
			OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.
Proprietary models													
Gemini 1.5 Pro	–	–	85.90	67.00	79.30	58.40	63.40	62.00	69.30	58.60	76.40	52.60	62.50
GPT-4o	–	–	69.80	64.20	71.60	51.10	70.30	59.80	64.50	57.90	75.70	48.70	60.80
Offline models													
LLaVA-Video	7B	64	69.80	59.60	66.40	50.60	72.30	61.40	63.50	51.20	64.20	9.70	41.70
LLaVA-OneVision	7B	64	67.10	58.70	69.80	49.40	71.30	60.30	62.80	52.50	58.80	23.70	45.00
Qwen2-VL	7B	64	69.10	53.20	63.80	50.60	66.30	60.90	60.70	44.40	66.90	34.40	48.60
InternVL-V2	8B	64	68.50	58.70	69.00	44.90	67.30	56.00	60.70	43.10	61.50	27.40	44.00
LongVU	7B	1 fps	55.70	49.50	59.50	48.30	68.30	63.00	57.40	43.10	66.20	9.10	39.50
Streaming models													
Flash-VStream	7B	1 fps	25.50	32.10	29.30	33.70	29.70	28.80	29.90	36.40	33.80	5.90	25.40
VideoLLM-online	8B	2 fps	8.10	23.90	12.10	14.00	45.50	21.20	20.80	22.20	18.80	12.20	17.70
Dispider	7B	1 fps	57.70	49.50	62.10	44.90	61.40	51.60	54.50	48.50	55.40	4.30	36.10
TimeChat-Online	7B	1 fps	69.80	48.60	64.70	44.90	68.30	55.40	58.60	53.90	62.80	9.10	42.00
StreamAgent	7B	1 fps	71.20	53.20	63.60	53.90	67.30	58.70	61.30	54.80	58.10	25.80	41.70
Reasoning-enabled models													
VST (Guan et al., 2026)	7B	2 fps	80.50	55.10	**72.40**	**55.10**	76.20	64.10	**67.20**	**56.90**	64.90	**48.40**	**56.70**
ThinkStream (Liu et al., 2026b)	3B	2 fps	**85.23**	**64.22**	69.82	49.43	69.31	**64.13**	67.03	53.87	59.46	43.55	52.30
Ours													
SyncPro-RL	4B	1 fps	69.80	53.21	67.24	50.00	**76.24**	50.54	61.17	55.22	67.57	8.06	43.62
SyncPro-SFT	4B	1 fps	69.80	53.21	68.10	48.88	**76.24**	50.54	61.13	55.56	**68.24**	8.06	43.95

D. Analysis of Inner Update Timing

We analyze the 50 evaluated rollouts in Figure 5, which contain 79 annotated *Inner* updates. Predicted activations are matched one-to-one with reference updates within each case. Exact matching requires the same micro-turn. Tolerant matching permits a displacement of up to one or two micro-turns. We pool event counts across cases to compute precision, recall, and F1 in Table 10. Figure 5 reports the mean of per-case F1 scores.

Table 10 | **Temporal alignment of *Inner* updates on 50 rollouts.** All settings use the same 79 reference updates. Precision, recall, and F1 pool event counts across cases and are percentages. Tolerance is measured in micro-turns.

Step	Tolerance	Activations	Matches	Precision	Recall	F1
100	0	101	34	33.7	43.0	37.8
200	0	75	28	37.3	35.4	36.4
500	0	113	53	46.9	67.1	55.2
1000	0	93	56	60.2	70.9	65.1
1000	1	93	68	73.1	86.1	79.1
1000	2	93	69	74.2	87.3	80.2

Learning to select update times. From step 500 to 1000, the number of activations falls from 113 to 93 while exact matches increase from 53 to 56. Precision improves by 13.3 percentage points and recall by 3.8 points. The later checkpoint achieves better alignment with fewer activations.

Temporal tolerance. At step 1000, allowing one micro-turn of tolerance increases matches from 56 to 68. A second micro-turn adds one match. Among the 69 matches within two micro-turns, 56 are exact, eight occur one step early, four occur one step late, and one occurs two steps early. Thus, most additional matches recovered by temporal tolerance lie within one neighboring micro-turn.

Random timing with matched activation counts. We compare the step-1000 activations with random update times under the same activation count. The random baseline averages 14.3 exact matches and 33.2 matches within one micro-turn. The model obtains 56 and 68 matches under the corresponding criteria. This comparison supports temporal alignment with the annotated updates beyond that obtained from random timing at the same activation count.

E. Data-Construction Prompts

The following boxes are formatting-normalized abridgements of the execution prompts. Bracketed ellipses mark omitted explanations and few-shot examples. Each box is kept intact to prevent a prompt from spanning pages. The released artifacts will include the complete execution prompts.

A1. Dual-stream model interface (abridged)

```
ROLE
You are a streaming-video assistant performing proactive response.

INPUT
The user first provides "Query: ...". Video frames then arrive at 1 fps.

At every chunk, make two independent decisions:
  internal: <silent> or <inner>...</inner>
  external: <silent> or <speak>...</speak>
Either, both, or neither branch may activate.

INTERNAL DECISION
Record only new query-relevant evidence or state changes that must persist.
Use concise, objective, information-additive summaries. Do not repeat stored
facts. If no useful state change occurred, output <silent>.

EXTERNAL DECISION
Respond only when the query's observable condition is satisfied. The response
must be concise, query-compliant, and supported by evidence available now.
Only <speak> content is visible to the user. If no response is warranted,
output <silent>.

OUTPUT
Emit exactly one internal decision and one external decision. Do not nest tags
or place text outside them.

QUERY-TYPE STRATEGIES AND EXAMPLES
[...]
```

A2. Proactive-query construction and filtering (abridged)

```
TASK
Generate 1--5 natural proactive requests for the observed stream, then retain
only requests that satisfy every feasibility criterion.

GENERATION CRITERIA
1. The answer depends on future visual evidence.
2. A short local clip cannot resolve the request.
3. Correct behavior requires intermediate state updates.
4. The response has a precise, observable trigger before stream termination.
5. The request is visually grounded and non-speculative.
6. The query is one sentence, normally 8--18 words, and never exceeds 20.
Do not refer to a video, clip, recording, timeline, absolute time, or ending.

FILTER IF ANY CONDITION HOLDS
requires_terminal_condition | requires_audio | unanswerable |
requires_external_knowledge

OUTPUT
Return valid JSON only. For each retained request include:
query, interaction_type, reasoning_pattern, why_proactive,
why_think_is_needed, think_steps, what_to_track, response_steps.
For filtering, return KEEP/FILTER, category, evidence-based reason, and
plausible answer times for KEEP items.

TAXONOMY, SCHEMA DETAILS, AND FEW-SHOT EXAMPLES
[...]
```

A3. Causal dual-stream trajectory synthesis (abridged)

```
TASK
Observe the scene causally. Track only what matters for the query and respond
at the earliest valid moment. A trajectory may contain one or many responses.

HIDDEN-STATE RULES
1. Include at least one state update when one is needed.
2. Update on query-relevant events, not at fixed intervals.
3. Store the minimal sufficient state: counts, entity status, or milestones.
4. Do not write process narration such as "continue monitoring".
5. Within this synthesis route, a hidden state cannot share a timestamp with,
   or preview, the next response.
6. Use an observer's perspective. Never refer to a video or recording.

RESPONSE RULES
Respond when the condition first becomes supported. Do not respond early or
late. Use only evidence observed by that time. Keep the content concise and
omit clock references from the response text.

OUTPUT
Produce only a chronological sequence:
{"think": "...", "time": "MM:SS"}
{"response": "...", "time": "MM:SS"}

FEW-SHOT TRAJECTORIES
[...]
```

A4. Response judging and full-trajectory revision (abridged)

```
EVALUATION
Given the video, query, and timestamped responses:
1. Judge each response for factual correctness and trigger timing.
2. Mark content-free repetitions or non-events for removal.
3. Find every query-relevant response opportunity that was missed.
A single-response query permits only its target event. A multi-response query
requires one response for each distinct target occurrence.

OUTPUT
Return one JSON object:
{
  "responses": [
    {"time": "...", "correct": true|false, "reason": "...",
     "proposed_time": "MM:SS|SKIP", "proposed_content": "..."}
  ],
  "missed": [{"around": "MM:SS", "reason": "..."}],
  "keep": true|false,
  "summary": "..."
}
Include proposed fields only when a correction is required.

REVISION
Re-observe the stream and regenerate the entire trajectory from scratch.
[FIX T] corrects content or timing. [REMOVE T] deletes a redundant response.
[ADD ~T] inserts a missed response. Do not patch isolated entries.

DETAILED CRITERIA AND EXAMPLES
[...]
```

A5. Inner-state effectiveness test (abridged)

```
For each target response, construct two reconstruction contexts.

A. VISUAL REFERENCE
Input: query + prior user-visible responses + frames observed since the
previous response.
Excluded: all hidden-state entries.

B. INNER-STATE CONTEXT
Input: query + prior hidden-state and response entries + only the most recent
visual window.
Excluded: earlier visual evidence.

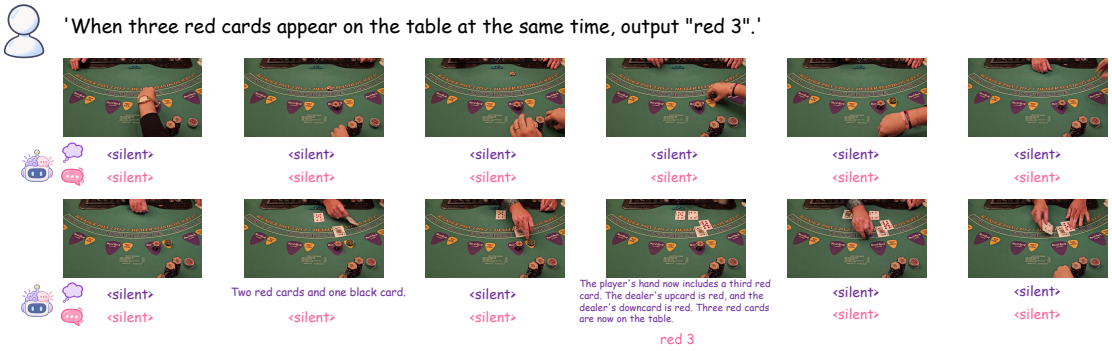
RECONSTRUCTION
Produce one short response supported at the end of the supplied context.
If the response is not yet warranted, output exactly NO_RESPONSE_YET.

SEMANTIC JUDGE
Compare prediction and reference under the query. Return:
{"correct": true|false, "reason": "<one short sentence>"}
Wording and harmless detail may differ. A missing key event, contradiction,
different subject or action, refusal, empty output, or NO_RESPONSE_YET is
incorrect when the reference is substantive.

RETENTION RULE
Keep a trajectory only if the inner-state context reconstructs every applicable
target. The visual-reference result records whether visual-history removal
preserves capability or exposes information supplied by the hidden state.
```

F. Case Studies

Figure 7 shows how SyncPro coordinates its two streams. *Inner* records relevant observations while *Response* provides the requested outputs. The branches activate together or separately with periods of silence. These behaviors illustrate how independent activation coordinates ongoing evidence tracking with selective user-facing responses.



(a) StreamingBench-PO.



(b) OVO-FAR repetition counting.



(c) ProactiveVideoQA.

Figure 7 | **Case studies.** *Inner* updates are purple and *Response* outputs are pink.