WATCHWEAVE: CONCURRENT PROACTIVE PERCEPTION AND TRANSACTIONAL ACTION OVER STREAMING VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video assistants must sustain user requests across ongoing observations, intervening questions, and instruction changes. Timely responses require tracking both visual events and the changing requests and reminder histories associated with them. We introduce WatchWeave, a runtime for concurrent proactive perception over a shared video stream. It maintains independently editable, versioned per-request records (Watches), combines event-phase and transition evidence with reminder history, and recovers weak event cues through relative changes in the model's binary prediction scores. It checks proposed updates against current state and revalidates requests before releasing generated responses. On all 3,738 IPIBench episodes, WatchWeave achieves Overall scores of 30.58% and 34.54% with Qwen3-VL-8B and Qwen3-VL-32B, respectively. The 8B configuration exceeds published same-backbone IPI-Agent by 2.93 percentage points, with gains in repeated monitoring and multi-task management. On all 250 OmniInteract samples, native speech achieves an overall Interaction-Aware Quality-Timeliness F1 (IA-QTF1) score of 0.420 under video-time evaluation, extending the assessment to continuous multimodal interaction.

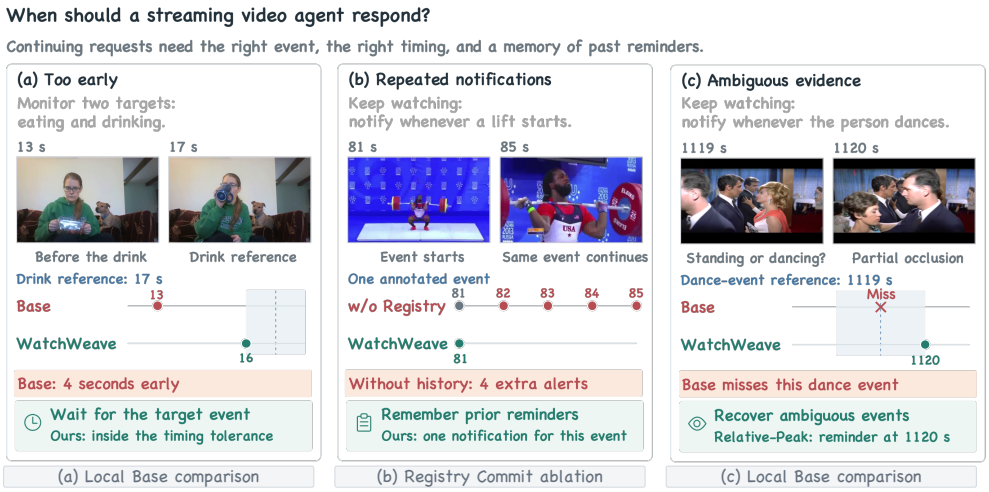


Figure 1: **Premature, repeated, and missed notifications in streaming video.** Selected 8B IPIBench traces compare WatchWeave with a local reference baseline in (a,c) and its Registry Commit ablation in (b).

1 INTRODUCTION

Video-language interaction is moving from completed clips to unfolding streams (Chen et al., 2024; Wang et al., 2025c; Qian et al., 2025), where delegated monitoring requires deciding when to respond (Wang et al., 2025b; Azad et al., 2026). Instructions must persist until fulfilled, revised, or cancelled,

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.

even across intervening questions (Li et al., 2026b). We consider multiple requests sharing one stream: "notify me when this person eats, and separately when they drink" establishes two independently completed or edited targets.

Three sources of change complicate monitoring. Intervening questions and edits require retaining unfinished targets; IPIBench reports weakened monitoring after such questions (Li et al., 2026b). Fluctuating visual evidence can obscure event boundaries: a cup approaching the mouth resembles drinking, and brief actions may be only partly observed. Isolated predictions can produce premature, duplicate, or missed reminders (Figure 1). Finally, requests or reminder histories may change during inference: cancellation can invalidate an otherwise correct answer before it is ready. Valid notification thus depends on event evidence, request requirements, prior notifications, and execution-time state.

IPI-Agent addresses task retention through structured proactive memory and supports additions, edits, cancellation, and parallel monitoring. Its temporal gate suppresses or recovers responses using thresholded changes in task–video similarity (Li et al., 2026b, Secs. 4.2–4.3). Such changes measure visual support, but do not by themselves determine whether an occurrence has already been reported or a request has changed since inference began. We study this as *concurrent proactive perception*: maintaining multiple simultaneously active, independently editable requests over a shared stream and deciding when each warrants a valid response.

WatchWeave links perceptual evidence to request-specific notification history and execution-time validity through three mechanisms (Figure 2). **Persistent Intent Weaving** maintains each target as a versioned *Watch* with its own instructions and history. **Occurrence-Centric Perception** separates event progress from reminder eligibility, combining evidence and request history to handle new and repeated notifications, with selective recovery for weak visual evidence. **Transactional Agency** checks proposed updates against current state and revalidates requests before releasing generated responses.

On all 3,738 IPIBench episodes, WatchWeave obtains Overall scores of 30.58% and 34.54% with Qwen3-VL-8B and Qwen3-VL-32B, respectively. Against published same-backbone IPI-Agent, 8B improves Overall by 2.93 percentage points, Repeated Proactiveness by 5.66, and Multi-task Management by 5.20, supporting sustained monitoring and concurrent request handling. Component removals examine the roles of event tracking, persistence, and recovery. Under video-time evaluation on all 250 OmniInteract samples, native speech reaches IA-QTF1 scores of 0.420 overall and 0.165 on continuous-response 1QnA.

2 Related Work

Streaming video perception and response timing. Video representation methods improve the evidence available for understanding (Pan et al., 2026; Yuan et al., 2025), while streaming architectures support online processing (Chen et al., 2024; Wu et al., 2024; Zhang et al., 2025a; Huang et al., 2025; Di et al., 2025; Xu et al., 2026; Yuan et al., 2026). Proactive models further address response timing and anticipation (Li et al., 2025; Qian et al., 2025; Wang et al., 2025a;c; 2026; Xia et al., 2025; Yang et al., 2025a; Azad et al., 2026; Ma et al., 2026). Our focus is whether this evidence still warrants a notification under the request's current state and reminder history.

Persistent requests and valid execution. Visual feedback and egocentric assistants connect observations to user needs (Fu et al., 2025; Kim et al., 2025; Zhang et al., 2025b). IPI-Agent uses persistent memory and similarity-change gating (Li et al., 2026b); ProAct and ProEvent organize actions through task graphs and mutable timetables, respectively (Zhu et al., 2026; Li et al., 2026a). Complementary work studies concurrent-state consistency and pre-commit intent validation (Herlihy & Wing, 1990; Lin et al., 2026b). WatchWeave integrates visual event tracking, per-request reminder history, and execution-time validation in one streaming runtime. This connects the perception question of what happened with the interaction question of which outstanding request should receive a response.

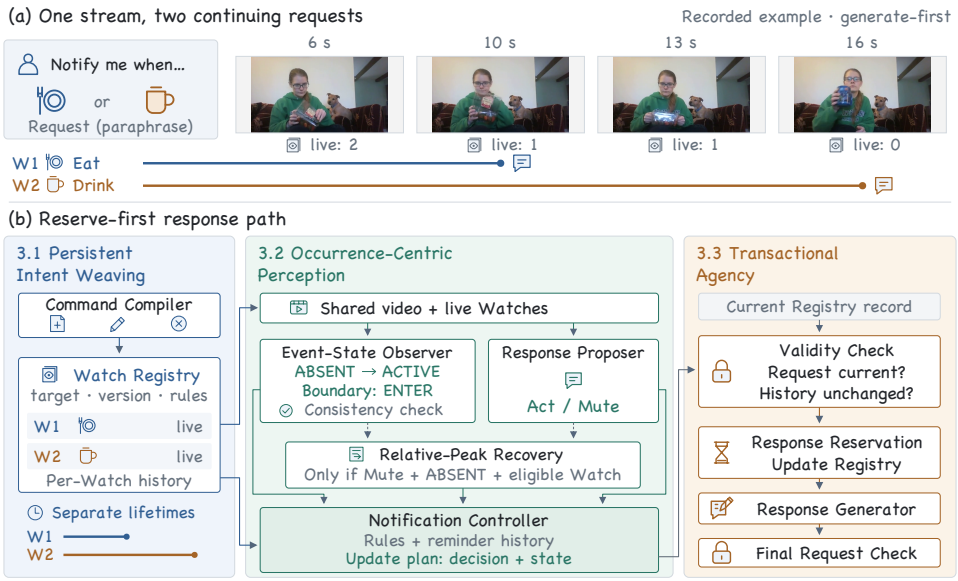


Figure 2: **From continuing requests to valid responses.** (a) IPIBench Multi-task Management 10 with Qwen3-VL-32B: eating/drinking Watches complete independently at 10/16 s via the generate-first path. (b) Reserve-first architecture (Sections 3.1–3.3): evidence and reminder history determine a plan, which is checked against current Registry state before reservation, generation, and final request validation. Solid arrows carry evidence/plans; dashed arrows mark conditional recovery.

Evaluation and multimodal interaction. Benchmarks assess temporal and streaming video understanding (Cai et al., 2024; Lin et al., 2026c; Niu et al., 2025; Xun et al., 2025; Yang et al., 2025b; Xiong et al., 2025; Wang et al., 2025b) and mobile/multimodal interaction (Lu et al., 2026b; Wang et al., 2025d). Always-on models (Défossez et al., 2024; Lu et al., 2026a; Xu et al., 2025; Cui et al., 2026) and full-duplex benchmarks (Lin et al., 2025b;a; 2026a) address spoken interaction.

3 WatchWeave

Problem formulation. At observation step t, the system receives the video and interaction context X_t available so far. Given evolving user requests and their notification histories, it must decide when and how to respond while retaining unfinished tasks across questions and edits. A request specifies what to detect, when to notify, and what response to provide. Several requests may refer to the same scene but have different completion times and reminder rules. The system therefore tracks progress per request: answering an immediate question leaves ongoing monitoring intact, while completing one target leaves the others active. WatchWeave represents each request as a *Watch* and proceeds in three stages (Figure 2(b)): maintain the requests, combine visual evidence with reminder history, and execute the resulting response plans.

3.1 Persistent Intent Weaving

An immediate question should not replace an unfinished monitoring task, and editing one target should not reset another's history. The *Command Compiler* therefore separates a compound request into independently maintained *Watches*. Figure 2(a) creates one for eating and one for drinking. Watch i at revision r is

$$W_i^r = (p_i, q_i, a_i), \tag{1}$$

where p_i describes what to detect, q_i specifies when and how often to notify, and a_i describes the requested response. The rules q_i distinguish action onset, state entry, and condition presence, and specify once-only or repeated reminders with confirmation and timing requirements. For example, "tell me when drinking starts" asks for an action onset, whereas "keep reminding me while the pan

Table 1: **Two kinds of event evidence.** (a) A phase describes one observation. (b) A boundary describes a change between two observations. The observer predicts both; the runtime checks their agreement. Non-active means ABSENT, PREPARE, or POST; UNKNOWN is unresolved.

(a) Event phases: what is happening now?

ABSENT	Target is inactive, with no preparation or completion cues.
PREPARE	Preparation for the target action, before it starts.
ACTIVE	Target action, state, or condition currently holds.
POST	Action has ended, with visible completion or aftermath.
UNKNOWN	Evidence is insufficient to assign a phase.

(b) Boundary labels: what changed between observations?

Label	Consistent phase pair	Meaning
ENTER	Non-active → ACTIVE	Target becomes active.
EXIT	ACTIVE → non-active	Target stops being active.
STAY_ACTIVE	ACTIVE → ACTIVE	Target remains active.
STAY_INACTIVE	Non-active → non-active	Neither observation is active.
UNKNOWN	Unresolved	Change cannot be determined.

is smoking" calls for repeated guidance during a continuing condition. Separating the visual target from the reminder rule lets the same event evidence serve these different requests.

The *Watch Registry* $\mathcal{R}$ stores W_i^r and state $z_{i,t}$: event evidence, confirmation counts, reminder allocations, and monitoring status. Instruction edits advance r; observations and reminders update $z_{i,t}$. Adding, editing, or cancelling a target updates its record; immediate questions leave monitored tasks intact. Watches share video context but retain independent histories (Figure 2(a)). Appendix B.1 maps this representation to the implementation.

3.2 Occurrence-Centric Perception

A drinking event may continue after its reminder. Deciding whether to notify again therefore requires both visual evidence and reminder history. Figure 2(b) combines three evidence sources for this decision.

Event evidence and response proposals. The *Event-State Observer* supplies $e_{i,t}$: event phases for individual observations and *boundary labels* for adjacent pairs. A phase describes what is happening; a boundary describes whether the target started, stopped, or continued (Table 1).

Uninterrupted drinking labelled ACTIVE→PREPARE→ACTIVE could produce a spurious onset. The observer therefore also judges boundaries directly from paired views. Entry requires agreement: a known non-active phase followed by ACTIVE must coincide with ENTER. ACTIVE indicates presence; STAY_ACTIVE, continuation. Conflicts remain unconfirmed, and UNKNOWN marks insufficient visual evidence. Observation identities prevent duplicate entries across overlapping windows (Appendix B.2).

PREPARE distinguishes a cup approaching the mouth from drinking and supports an auxiliary boundary query; sustained preparation helps confirm its ENTER result. POST captures aftermath, such as the cup moving away. A transition from pre-action evidence to consistent POST can support short-action recovery when the active phase fell between samples.

The *Response Proposer* reads video context and Watches, returning $g_{i,t} \in \{\texttt{Act}, \texttt{Mute}\}$: propose a response or make no proposal. Temporally confirmed Act proposals support missed entries and continuing conditions. ENTER follows the observer's boundary checks, while Act confirmation accumulates response proposals over successive observations. The Notification Controller combines these complementary inputs with the Watch's reminder history.

Relative-Peak Recovery. Categorical No answers can remain unchanged while support for a brief event rises and falls. *Relative-Peak Recovery* retains this variation to propose candidates. To expose small objects and local action cues, nine overlapping crops are enlarged into one 3×3 mosaic of

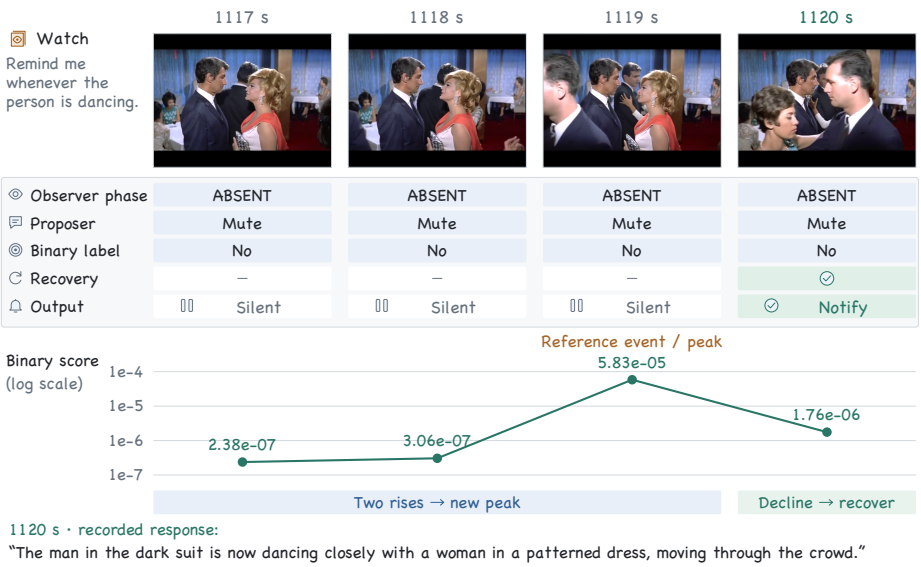


Figure 3: **Recorded Relative-Peak recovery.** IPIBench Repeated Proactiveness 204 with Qwen3-VL-8B. All probes answer No with observer ABSENT and proposer Mute; the score decline at 1120 s triggers recovery after the 1119 s peak. Scores are shown on a log scale.

the current frame. One binary query to the same VLM asks whether any tile satisfies the Watch condition, returning Yes/No and token scores. Probes selectively target supported Watches with Mute proposals and ABSENT observations, using one additional forward pass per admitted frame (Appendix B.3). Let n index consecutive admitted probes, $\xi_{i,n}$ their first-token Yes score normalized over reported Yes/No forms, and $\hat{y}_{i,n}$ the emitted token. $H_{i,n}$ contains the latest at most eight probe indices, including n. With at least four probes, $\kappa_{i,n}$ flags a candidate:

$$\kappa_{i,n} = [\hat{y}_{i,n-3:n} = \mathrm{No}^4] \wedge [\xi_{i,n-3} < \xi_{i,n-2} < \xi_{i,n-1}] \\ \wedge [\xi_{i,n} < \xi_{i,n-1}] \wedge \left[\xi_{i,n-1} > \max_{\substack{j \in H_{i,n} \\ j \le n-2}} \xi_{i,j} \right]. \tag{2}$$

Here No^4 means four consecutive No outputs, and j indexes retained probes. Two rises to a history maximum followed by a decline identify a candidate at probe n, one probe after the peak. This within-history comparison recovers evidence hidden by categorical answers without a fixed score threshold. The decline confirms that support has passed a local peak, placing the decision at the current probe rather than retrospectively at the peak. Histories follow individual Watches and reset across task or monitoring-cycle changes, so the comparison follows one continuing target. The resulting recovery evidence $\lambda_{i,t}$ retains its source and observation pair alongside the observer's predictions. In Figure 3, two score rises peak at the 1119 s reference event; the 1120 s decline triggers a recovery candidate despite unchanged No outputs. Appendix B.3 specifies probe admission, history resets, and invocation counts.

History-aware notification. The *Notification Controller* turns the three evidence sources into a notification proposal by applying the Watch's rules and history:

$$(z'_{i,t}, d_{i,t}) = F\big(W_i^r, z_{i,t}; g_{i,t}, e_{i,t}, \lambda_{i,t}\big). \tag{3}$$

Here F is the deterministic notification rule, $z'_{i,t}$ the proposed next state, and $d_{i,t} \in \{0, 1\}$ the notification decision. Here $d_{i,t} = 1$ requests a notification. Silent steps with $d_{i,t} = 0$ still accumulate evidence and restore reminder availability, allowing the system to progress between responses. The resulting update plan $\pi_{i,t} = (W_i^r, z_{i,t}, z'_{i,t}, d_{i,t})$ records the before-state and proposed update for Section 3.3 to check and apply.

Confirmed ENTER supports onset-sensitive requests; confirmed Act and recovery candidates compensate for incomplete entry evidence or support continuing conditions. The controller checks each



Figure 4: **Event tracking and repeated notification.** IPIBench Repeated Proactiveness 154 (32B): a reminder at 76 s, suppressed repeats at 77–78 s, restored eligibility at 101 s, and a new reminder at 105 s. Checks show predicted entry support; reminder states are post-decision. Insets magnify original pixels.

against timing rules and reminder availability. Allocating a reminder completes a once-only Watch or consumes the current reminder opportunity of a recurring Watch.

Another reminder becomes available after the configured interval and either sustained Mute proposals or consistent ABSENT/POST evidence after a Mute. Because Mute concerns the response decision, repeated guidance can resume during a continuing ACTIVE event. Figure 4 shows this distinction on a recurring pan-monitoring request: positive proposals at 77–78 s remain silent after the 76 s reminder. Mute, ABSENT, and the elapsed interval restore eligibility at 101 s; the next notification occurs at 105 s. Appendix B.2 lists the confirmation thresholds and notification decision order.

3.3 Transactional Agency

While a response plan is being computed, the user may cancel a request or another accepted plan may update its reminder history. For recurring Watches with a structured plan, Figure 2(b) uses a reserve-first path. The *Validity Check* verifies that the request revision and observation remain current. The Registry applies the plan only if its current state also matches $z_{i,t}$. Comparing and updating that state is one atomic operation, *compare-and-swap* (CAS), for each Watch. These checks address different changes: a request revision detects an instruction edit, whereas CAS detects a history update even when the instruction is unchanged. For example, two plans may refer to the same drinking request, but the first accepted reminder changes the state expected by the second plan.

For an accepted notification, *Response Reservation* writes $z'_{i,t}$, updates the reminder count and time, and creates a pending response record, claiming the opportunity before generation. Replays return the same record; other plans based on the old $z_{i,t}$ fail the state comparison. A new reservation starts the *Response Generator*. Writing $\rho_{i,t}$ for the reservation created from $\pi_{i,t}$, the reservation-to-generation sequence is

$$\rho_{i,t} = \mathrm{ReserveCAS}(\mathcal{R}, \pi_{i,t}), \qquad y_{i,t} = G(X_t, W_i^r, \rho_{i,t}). \tag{4}$$

Here $\mathrm{ReserveCAS}$ compares and atomically updates the Registry, and G generates candidate answer $y_{i,t}$ from the reserved context. The *Final Request Check* verifies the request and observation again after generation. If the user cancelled or changed the request meanwhile, the answer is withheld; otherwise it is attached to its reservation for release. The reservation identifies the reminder being answered; the final check determines whether that answer still belongs to the current request. Figure 6 illustrates replay and cancellation; Appendix B.4 gives the algorithm.

One-shot Watches and cases without a valid structured plan use a *generate-first path*: content generation precedes the current-request check and Registry update. Appendix B.4 details both paths and their record-handling behavior.

4 EXPERIMENTS

4.1 EVALUATION SETUP

Models. WatchWeave uses Qwen3-VL-8B/32B-Instruct (Bai et al., 2025); WatchWeave-Omni uses Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025) (30B total, 3B active parameters). Baselines come from the benchmark publications.

IPIBench. IPIBench (Li et al., 2026b) contains 3,738 episodes covering proactive monitoring, task management, and reactive/proactive interleaving. Following its official protocol (Li et al., 2026b, Sec. 5.1), we use one frame per second (FPS), the released polling windows, and the latest 16 frames. Decoding uses temperature zero with thinking disabled. Overall averages the three category scores using the released task metrics and subtype aggregation (Appendix A).

OmniInteract. OmniInteract (Lu et al., 2026c) contains 250 samples. Its 1Q1A tasks cover proactive, nested, and real-time interactions, while 1QnA requires multiple time-grounded responses to a continuing instruction. We sample video at 2 FPS with a 16-frame context. Qwen3-Omni interprets user speech, and the resulting commands enter the controller at the end of each speech segment. We evaluate paired text and native-audio outputs from the same control decisions. Native audio uses Qwen3-Omni's Talker; text uses a fixed reading-rate timeline. Both use the released Interaction-Aware Quality-Timeliness F1 (IA-QTF1) evaluator and GPT-4o judge protocol (Lu et al., 2026c, Sec. 3.3 and App. A.4). Responses are placed on the video timeline using recorded decision times and speech playback durations; local speech-synthesis computation time is excluded. Appendix A gives the detailed input and timing settings.

4.2 MANAGING LIVE USER COMMITMENTS

WatchWeave obtains 30.58 Overall with 8B and 34.54 with 32B (Table 2). The 8B result exceeds published IPI-Agent with the same backbone by 2.93 percentage points (pp), improving all three category means. Repeated Proactiveness increases from 15.09 to 20.75, and Multi-task Management from 11.82 to 17.02, corresponding to repeated event monitoring and concurrent request handling. The Interleaved category also rises from 27.18 to 29.66, extending the improvement to episodes that combine monitoring with reactive questions. The 32B configuration reaches 34.36 on Understanding and 53.70 on Cancellation, compared with 25.15 and 46.76 for 8B.

4.3 STREAMING MULTIMODAL INTERACTION

Under the same controller decisions, WatchWeave-Omni scores 0.420 All Global with native speech and 0.427 with text (Table 3). On 1QnA, the respective scores are 0.165 and 0.205, versus the highest published score of 0.052. Native speech also reaches 0.600 on Nested and 0.492 on 1Q1A. The text–speech gap is 0.007 overall and 0.040 on 1QnA. Because the two outputs share control decisions, this comparison captures the effect of response realization and its timing under the evaluation protocol.

4.4 UNDERSTANDING THE COMPONENTS

Table 4 evaluates four component removals on all 3,738 episodes, with the same Full reference as Table 2. Removing a component also disables its dependent mechanisms.

Event tracking. Removing Event-State/Recognition lowers Overall by 2.86 pp (8B) and 2.00 pp (32B). Repeated Proactiveness falls from 20.75 to 12.26 and from 25.00 to 16.51, respectively, supporting event tracking for repeated notification.

Persistent task state. Removing Registry Commit retains readable Watch snapshots but stops persisting state updates. Overall falls by 1.71 pp (8B) and 2.61 pp (32B). Multi-task Management drops from 17.02 to 8.27 and from 20.09 to 9.93, respectively, linking task-state persistence to concurrent request handling.

Table 2: **IPIBench: complete category- and task-level comparison** (%, higher is better). Baselines: Li et al. (2026b, Tables 2, 4). Avg.: category mean; Overall: mean of three categories. Tim./Und./Rep.: Timing/Understanding/Repeated Proactiveness; Can./Mod./Mul.: Cancellation/Modification/Multi-task; R2P/RuP/RaP: Reactive-to/under/after-Proactive. –/NR: unreported; N/A: human. Size: nominal model scale. Gray text: closed-source; blue rows: WatchWeave. Bold/underline: best/second-best per column, excluding closed-source and human rows; ties share marks.

Method / model	Size	Monitoring				Management				Interleaved				Overall
		Tim.	Und.	Rep.	Avg.	Can.	Mod.	Mul.	Avg.	R2P	RuP	RaP	Avg.	
Human reference														
Human Level	N/A	98.00	94.00	92.00	94.67	98.00	92.00	98.00	96.00	96.00	96.00	98.00	96.67	95.78
Offline models (reported)														
Gemini 3 Pro	NR	43.49	24.90	18.40	28.93	19.44	21.84	19.39	20.22	24.76	17.68	26.69	23.04	24.06
Gemini 2.5 Pro	NR	45.36	25.26	13.21	27.94	24.54	20.81	16.78	20.71	21.17	17.54	28.24	22.32	23.66
GPT-5.4	NR	54.64	23.52	8.96	29.04	38.43	24.50	7.57	23.50	27.04	15.06	29.28	23.79	25.44
GPT-4o	NR	50.59	25.99	13.68	30.09	11.57	31.88	14.18	19.21	27.04	17.68	26.17	23.63	24.31
LLaVA-OneVision	7B	12.14	8.66	0.00	6.93	1.39	1.01	0.47	0.96	3.91	0.00	3.89	2.60	3.50
InternVL3	8B	32.51	22.31	3.77	19.53	6.94	6.04	3.55	5.51	7.17	0.00	9.33	5.50	10.18
Qwen3-VL	8B	43.12	24.51	4.25	23.96	1.39	17.45	9.22	9.35	24.43	19.75	23.05	22.41	18.57
Qwen3.5-Plus	397B	35.25	24.35	7.89	22.50	12.96	31.54	10.87	18.46	**32.25**	22.01	30.57	28.28	23.08
GLM-4.6V	106B	50.43	27.59	10.85	29.62	15.28	24.16	9.69	16.38	22.15	6.77	**39.37**	22.76	22.92
Online models (reported)														
VideoLLM-online	8B	14.63	0.14	1.82	5.53	7.87	1.68	6.62	5.39	0.00	1.10	1.55	0.88	3.93
Dispider	7B	18.72	0.65	1.82	7.06	–	–	–	–	–	–	–	–	–
Flash-VStream	7B	5.91	0.00	0.00	1.97	2.31	0.00	0.00	0.77	0.00	0.00	0.78	0.26	1.00
IPI-Agent (reported)														
Gemini 3 Pro	NR	56.27	35.67	28.30	40.08	51.85	35.23	29.08	38.72	30.62	22.38	29.53	27.51	35.44
GPT-5.4	NR	57.20	24.30	10.85	30.78	48.61	24.83	14.66	29.37	28.01	19.34	33.42	26.92	29.02
Qwen3.5-Plus	397B	52.80	33.18	18.42	34.80	52.78	**33.22**	**25.53**	**37.18**	**32.25**	22.63	35.95	30.28	34.09
Qwen3-VL	8B	46.62	25.37	15.09	29.03	44.91	23.49	11.82	26.74	27.36	23.62	30.57	27.18	27.65
Ours														
WatchWeave	8B	51.02	25.15	20.75	32.31	46.76	25.50	17.02	29.76	29.97	28.45	30.57	29.66	30.58
WatchWeave	32B	**57.21**	**34.36**	**25.00**	**38.85**	**53.70**	29.19	20.09	34.33	28.66	**30.52**	32.12	**30.44**	**34.54**

Table 3: **OmniInteract results on all 250 samples.** IA-QTF1 fractions (higher is better); text and native audio share control decisions. WatchWeave uses video-time scoring (Appendix A). Gray text: closed-source; blue rows: WatchWeave. Bold/underline: best/second-best per column across both output modes, excluding closed-source systems.

Method / model	Size	Nested	Real-time	Proactive	1Q1A	1QnA	All
Published audio-visual systems							
Gemini 2.5 Flash Live	NR	0.398	0.553	0.121	0.428	0.028	0.344
Qwen3.5-Omni Flash Realtime	NR	0.379	0.524	0.108	0.401	0.023	0.323
AURA	8B	0.596	0.376	0.549	0.467	0.052	0.363
MiniCPM-o 4.5	9B	0.599	0.337	**0.607**	0.456	0.015	0.368
Ours — Native audio							
WatchWeave-Omni	30A3B	0.600	**0.431**	0.517	**0.492**	0.165	0.420
Ours — Text							
WatchWeave-Omni	30A3B	**0.627**	0.424	0.492	0.488	**0.205**	**0.427**

Response generation. Replacing the Response Generator with deterministic fallback answers produces the largest Overall drops: 3.88 pp (8B) and 4.14 pp (32B). Understanding falls from 25.15 and 34.36 to 0.38 for both sizes, whereas Management decreases by 1.06 and 0.88 pp: response content contributes beyond trigger timing.

Table 4: **WatchWeave component ablations.** All runs use 3,738 episodes (90,445 frames); Full matches Table 2. Scores: %; Δ: pp. Mon./Mgmt./Inter.: category means; Rep./Multi/Und.: Repeated Proactiveness/Multi-task Management/Understanding. †Registry removal also disables dependent modules. Blue shading marks Full; bold numbers are column maxima within each size block.

Size	Configuration	Category means			Selected tasks			Overall	Δ
		Mon.	Mgmt.	Inter.	Rep.	Multi	Und.		
8B	**WatchWeave (full)**	**32.31**	**29.76**	**29.66**	**20.75**	**17.02**	25.15	**30.58**	–
	w/o Event-State/Recognition	29.31	27.95	25.90	12.26	15.13	27.13	27.72	−2.86
	w/o Registry Commit†	30.16	27.25	29.20	12.26	8.27	27.40	28.87	−1.71
	w/o Response Generator	22.98	28.70	28.42	17.92	14.42	0.38	26.70	−3.88
	w/o Relative-Peak	31.69	29.20	28.40	16.51	14.66	**27.64**	29.76	−0.82
32B	**WatchWeave (full)**	**38.85**	**34.33**	**30.44**	25.00	**20.09**	34.36	**34.54**	–
	w/o Event-State/Recognition	35.46	32.28	29.89	16.51	17.97	33.88	32.54	−2.00
	w/o Registry Commit†	35.67	30.06	30.05	15.57	9.93	**34.48**	31.93	−2.61
	w/o Response Generator	27.48	33.46	30.27	25.00	18.44	0.38	30.40	−4.14
	w/o Relative-Peak	38.70	33.01	29.97	**25.94**	18.91	33.21	33.89	−0.65

Table 5: **Relative-Peak temporal-confirmation sensitivity.** Overall (%) on all 3,738 IPIBench episodes. Shading marks the default R2D1 setting; bold marks the best score per model. Details are in Appendix C.1.

	8B			32B		
R	$D=0$	$D=1$	$D=2$	$D=0$	$D=1$	$D=2$
1	29.36	29.30	29.37	33.69	**34.61**	34.58
2	29.57	**30.58**	29.30	34.10	34.54	33.76
3	29.58	29.43	29.46	34.16	33.82	34.47

Relative-peak recovery. Removing Relative-Peak Recovery gives 29.76/33.89 Overall (8B/32B), $-0.82/-0.65$ pp relative to Full. At 8B, Full raises Repeated Proactiveness from 16.51 to 20.75, while at 32B it raises Understanding from 33.21 to 34.36. The gains vary by task: the OFF variant has higher 8B Understanding and 32B Repeated Proactiveness, whereas Full has the higher aggregate score at both sizes. Appendix B.3 specifies the disabled branch and retained recovery routes.

Temporal confirmation length. Table 5 varies consecutive strict score rises $R \in \{1, 2, 3\}$ and declines after a new maximum $D \in \{0, 1, 2\}$ on all IPIBench episodes at both model sizes (Appendix C.1). R2D1 has the highest reported 8B score. At 32B, R1D1 and R2D1 score 34.61 and 34.54, respectively.

5 Conclusion

WatchWeave supports multiple editable requests over streaming video by combining persistent Watches, occurrence-aware perception, and transactional execution. It retains unfinished tasks across interactions, uses event evidence and reminder history to decide when to respond, and checks that generated answers still match current requests. On IPIBench, the 8B system improves repeated monitoring and multi-task management over published same-backbone IPI-Agent. OmniInteract further evaluates this approach with native speech under video-time scoring. Together, these results demonstrate the value of integrating persistent intent with history-aware notification.

AI USE STATEMENT

Generative AI tools assisted manuscript writing and polishing, literature retrieval and discovery, and preparation of figures and verification scripts. The authors are responsible for reviewing and approving all AI-assisted content.

REPRODUCIBILITY STATEMENT

Section A specifies the inference and scoring protocols; Section B.2 specifies the Notification Controller's confirmation and reminder rules. Appendix B provides component interfaces, policy settings, and execution pseudocode; Appendix D identifies the episodes used in the qualitative comparisons. The complete code will be publicly released upon acceptance.

REFERENCES

Shehreen Azad, Vibhav Vineet, and Yogesh S Rawat. StreamReady: Learning What to Answer and When in Long Streaming Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40494–40504, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Azad_StreamReady_Learning_What_to_Answer_and_When_in_Long_Streaming_CVPR_2026_paper.html.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report, 2025. URL https://arxiv.org/abs/2511.21631.

Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. TemporalBench: Benchmarking Fine-grained Temporal Understanding for Multimodal Video Models. *arXiv preprint arXiv:2410.10818*, 2024. URL https://arxiv.org/abs/2410.10818.

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 (CVPR)*, pp. 18407–18418, June 2024. URL https://arxiv.org/abs/2406.11816.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, Yuyang Wen, Huiping Liu, Rongkang Wang, Yaqi Zhang, Hongliang Wei, Chi Chen, You Li, Kechen Fang, Jie Zhou, Yuxuan Li, Guoyang Zeng, Chaojun Xiao, Yankai Lin, Xu Han, Maosong Sun, Zhiyuan Liu, and Yuan Yao. MiniCPM-o 4.5: Towards Real-Time Full-Duplex Omni-Modal Interaction, 2026. URL https://arxiv.org/abs/2604.27393.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming Video Question-Answering with In-context Video KV-Cache Retrieval. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 42115–42127, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/67a9b444cbcd647572c88194619f72d5-Paper-Conference.pdf.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024. URL https://arxiv.org/abs/2410.00037.

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, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Fu_ViSpeak_Visual_Instruction_Feedback_in_Streaming_Videos_ICCV_2025_paper.html.

Maurice P. Herlihy and Jeannette M. Wing. Linearizability: a correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463–492, July 1990. doi: 10.1145/78969.78972. URL https://doi.org/10.1145/78969.78972.

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.

Junhyeok Kim, Min Soo Kim, Jiwan Chung, Jungbin Cho, Jisoo Kim, Sungwoong Kim, Gyeongbo Sim, and Youngjae Yu. EgoSpeak: Learning When to Speak for Egocentric Conversational Agents in the Wild. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2990–3005, 2025. URL https://aclanthology.org/2025.findings-naacl.163/.

Guanzhen Li, Liangming Pan, and Leye Wang. ProEvent: An Event-centric Benchmark for Proactive Agents, 2026a. URL https://arxiv.org/abs/2607.17701.

Jinzhao Li, Yinuo Chen, Wenxuan Song, Yijia Lei, Yichi Zhang, Honglei Yan, Panwang Pan, and Miao Liu. IPIBench: Evaluating Interactive Proactive Intelligence of MLLMs under Continuous Streams, 2026b. URL https://arxiv.org/abs/2605.27074.

Wei Li, Bing Hu, Rui Shao, Leyang Shen, and Liqiang Nie. LION-FS: Fast & Slow Video-Language Thinker as Online Video Assistant. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3240–3251, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Li_LION-FS_Fast__Slow_Video-Language_Thinker_as_Online_Video_Assistant_CVPR_2025_paper.html.

Guan-Ting Lin, Shih-Yun Shan Kuan, Jiatong Shi, Kai-Wei Chang, Siddhant Arora, Shinji Watanabe, and Hung-yi Lee. Full-Duplex-Bench-v2: A Multi-Turn Evaluation Framework for Duplex Dialogue Systems with an Automated Examiner. *arXiv preprint arXiv:2510.07838*, 2025a. URL https://arxiv.org/abs/2510.07838.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H. Liu, and Hung-yi Lee. Full-Duplex-Bench: A Benchmark to Evaluate Full-duplex Spoken Dialogue Models on Turn-taking Capabilities. *arXiv preprint arXiv:2503.04721*, 2025b. URL https://arxiv.org/abs/2503.04721.

Guan-Ting Lin, Chen Chen, Zhehuai Chen, and Hung-yi Lee. Full-Duplex-Bench-v3: Benchmarking Tool Use for Full-Duplex Voice Agents Under Real-World Disfluency, 2026a. URL https://arxiv.org/abs/2604.04847.

Junda Lin, Zhaomeng Zhou, Zhi Zheng, Shuochen Liu, Tong Xu, Yong Chen, and Enhong Chen. VIGIL: Defending LLM Agents Against Tool-Stream Injection via Verify-Before-Commit. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9764–9785, San Diego, California, United States, jul 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.443. URL https://aclanthology.org/2026.acl-long.443/.

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, May 2026c. doi: 10.1109/icassp55912.2026.11463959. URL http://dx.doi.org/10.1109/ICASSP55912.2026.11463959.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. AURA: Always-On Understanding and Real-Time Assistance via Video Streams. *arXiv preprint arXiv:2604.04184*, 2026a. URL https://arxiv.org/abs/2604.04184.

Xudong Lu, Huankang Guan, Yang Bo, Jinpeng Chen, Xintong Guo, Shuhan Li, Fang Liu, Peiwen Sun, Xueying Li, Wei Zhang, Xue Yang, Rui Liu, and Hongsheng Li. PhoStream: Benchmarking Real-World Streaming for Omnimodal Assistants in Mobile Scenarios. *arXiv preprint arXiv:2601.22575*, 2026b. URL https://arxiv.org/abs/2601.22575.

Xudong Lu, Xueying Li, Annan Wang, Yang Bo, Jinpeng Chen, Zengliang Li, Nianzu Yang, Rui Liu, Xue Yang, Jingwen Hou, and Hongsheng Li. OmniInteract: Benchmarking Real-World Streaming Interaction for Real-Time Omnimodal Assistants, 2026c. URL https://arxiv.org/abs/2605.26485.

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 Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44139–44153, San Diego, California, United States, jul 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.2042. URL https://aclanthology.org/2026.acl-long.2042/.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How Far is Your Video-LLMs from Real-World Online Video Understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Niu_OVO-Bench_How_Far_is_Your_Video-LLMs_from_Real-World_Online_Video_CVPR_2025_paper.html.

Hewen Pan, Cong Wei, Dashuang Liang, Zepeng Huang, Pengfei Gao, Ziqi Zhou, Lulu Xue, Pengfei Yan, Xiaoming Wei, Minghui Li, and Shengshan Hu. UFVideo: Towards Unified Fine-Grained Video Cooperative Understanding with Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31833–31844, 2026. URL https://arxiv.org/abs/2512.11336.

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 (CVPR)*, pp. 24045–24055, June 2025. URL https://arxiv.org/abs/2501.03218.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-ASR Technical Report, 2026. URL https://arxiv.org/abs/2601.21337.

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, 2025a. URL https://arxiv.org/abs/2505.05467.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. ProactiveVideoQA: A Comprehensive Benchmark Evaluating Proactive Interactions in Video Large Language Models, 2025b. URL https://arxiv.org/abs/2507.09313.

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, 2025c. 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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 23257–23270, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/281970ef377580ab9ec3c7d3781110d4-Abstract-Conference.html.

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, 2025d. URL https://arxiv.org/abs/2503.22952.

Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. VideoLLM-MoD: Efficient Video-Language Streaming with Mixture-of-Depths Vision Computation. *Advances in Neural Information Processing Systems*, 37:109922–109947, 2024. URL https://arxiv.org/abs/2408.16730.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming Video Instruction Tuning. *arXiv preprint arXiv:2512.21334*, 2025. URL https://arxiv.org/abs/2512.21334.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming Video Understanding and Multi-round Interaction with Memory-enhanced Knowledge. *arXiv preprint arXiv:2501.13468*, 2025. URL https://arxiv.org/abs/2501.13468.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-Omni Technical Report, 2025. URL https://arxiv.org/abs/2509.17765.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-Time Understanding for Infinite Video Streams. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61463–61475, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6445dd88ebb9a6a3afa0b126ad87fe41-Paper-Conference.pdf.

Shuhang Xun, Sicheng Tao, Jungang Li, Yibo Shi, Zhixin Lin, Zhanhui Zhu, Yibo Yan, Hanqian Li, Linghao Zhang, Shikang Wang, Yixin Liu, Hanbo Zhang, Ying Ma, and Xuming Hu. RTV-Bench: Benchmarking MLLM Continuous Perception, Understanding and Reasoning through Real-Time Video. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2505.02064.

Haolin Yang, Feilong Tang, Lingxiao Zhao, Xinlin Zhuang, Yifan Lu, Xiang An, Ming Hu, Xiaofeng Zhang, Abdalla Swikir, Junjun He, Zongyuan Ge, Muhammad Haris Khan, and Imran Razzak. StreamAgent: Towards Anticipatory Agents for Streaming Video Understanding, 2025a. URL https://arxiv.org/abs/2508.01875.

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. SVBench: A Benchmark with Temporal Multi-Turn Dialogues for Streaming Video Understanding. *arXiv preprint arXiv:2502.10810*, 2025b. URL https://arxiv.org/abs/2502.10810.

Chao Yuan, Yang Yang, Yehui Yang, and Zach Cheng. DATE: Dynamic Absolute Time Enhancement for Long Video Understanding. *arXiv preprint arXiv:2509.09263*, 2025. URL https://arxiv.org/abs/2509.09263.

Chao Yuan, Shimin Chen, Minliang Lin, Limeng Qiao, Guanglu Wan, and Lin Ma. UniComp: Rethinking Video Compression Through Informational Uniqueness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18609–18618, 2026. URL https://arxiv.org/abs/2512.03575.

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, 2025a. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_Flash-VStream_Efficient_Real-Time_Understanding_for_Long_Video_Streams_ICCV_2025_paper.html.

Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. Proactive Assistant Dialogue Generation from Streaming Egocentric Videos. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12044–12068, 2025b. URL https://aclanthology.org/2025.emnlp-main.605/.

Xiaomeng Zhu, Fengming Zhu, Weijie Zhou, Ye Tian, Zhenlin Hu, Yufei Huang, Yuchun Guo, Xinyu Wu, Zhengyou Zhang, Fangzhen Lin, and Xuantang Xiong. ProAct: A Benchmark and Multimodal Framework for Structure-Aware Proactive Response, 2026. URL https://arxiv.org/abs/2602.03430.

A Evaluation Protocols and Reproducibility Details

Table 6 lists the samples and inputs used for benchmark, component, and speech/text evaluations.

Table 6: Evaluation samples and input schedules.

Evaluation	Episodes	Sampling	Interpretation
IPI full, each scale	3,738	1 FPS/16 frames	Full public collection; causal replay.
IPI component, each arm	3,738	1 FPS/16 frames	Full-collection interventions; predictions paired by episode ID.
Omni speech/text	250	2 FPS/16 frames	Paired text/native decisions; video-time replay with fixed-rate text or rendered speech.

IPIBench aggregation. We follow the official task metrics and released evaluator (Li et al., 2026b). Each of its three category scores averages three task scores. Timing and Understanding first average their released subtype scores. We retain this nesting:

$$\text{Overall} = \frac{1}{3}\sum_{c=1}^{3}\left(\frac{1}{3}\sum_{j\in\mathcal{T}_c}\text{TaskScore}_j\right). \tag{5}$$

Here c indexes the three categories, $\mathcal{T}_c$ is the set of three tasks in category c, and TaskScore_j is the released score for task j. Thus Overall weights categories and tasks equally, rather than weighting them by their episode counts. Following the official protocol (Li et al., 2026b, Sec. 5.1), proactive polling advances in 1-second steps from four seconds before to four seconds after each reference trigger. A first trigger is timely within one second of the reference; repeated and multi-turn tasks require all specified triggers and interactions to pass. Reference annotations schedule evaluation but are not part of model prompts.

Model and input configuration. The IPI models are Qwen3-VL-8B-Instruct and Qwen3-VL-32B-Instruct. Inputs use the native visual processor and at most the most recent 16 sampled frames, with temperature zero and thinking disabled. Empty active-Watch periods skip model calls. Separate role prompts implement the Command Compiler, Response Proposer, Event-State Observer, and Response Generator.

OmniInteract inputs and response timing. Video is sampled at 2 FPS with the latest 16 frames retained. Qwen3-Omni decodes speech segments using its native audio input; the resulting commands are released at each segment's end, without an external input-ASR model. Text and native-speech evaluation share these commands and the controller's recorded response decisions.

The text timeline uses 4 Chinese characters or 2.6 Latin-script words per second, with a minimum utterance duration of 0.4 seconds. Native speech is rendered from the response text by Qwen3-Omni's Talker as 24 kHz audio. Its start anchors come from the recorded response timeline; playback follows the actual waveform duration on a single channel, waits through user speech, and can be interrupted by a new user utterance. Local Talker synthesis time is not added to these anchors. The resulting audio is transcribed by Qwen3-ASR-1.7B and aligned with Qwen3-ForcedAligner-0.6B (Shi et al., 2026) before matching and IA-QTF1 scoring with the official evaluation code and GPT-4o judge protocol (Lu et al., 2026c, Sec. 3.3 and App. A.4). The evaluator measures response timing on the assembled media timeline. The released adapters construct this timeline differently: MiniCPM-o places generated audio at input-chunk offsets while logging computation time separately; Gemini uses the audio available during paced input streaming.

Model-size convention. Size denotes the nominal checkpoint or backbone scale, in billions of parameters. NR means that the cited benchmark/model sources do not report the evaluated service's size; N/A applies to the human reference. Explicit 7B/8B model suffixes follow the IPIBench source. The additional scales follow official model cards: Qwen3.5-Plus corresponds to the 397B family with 17B active parameters; GLM-4.6V is 106B; MiniCPM-o 4.5 is 9B. Our Qwen3-Omni checkpoint is designated 30B-A3B (30B total, 3B active), displayed as 30A3B for WatchWeave-Omni in the tables. Dispider's 7B refers to its final response LLM, following its original Table 2. AURA releases its 8B backbone; the scale excludes its separate ASR and TTS modules.

B Additional Method Details

B.1 Watch Interfaces and Recorded Compilation

A recorded compilation and edit. Figure 8 shows Multi-task Management 10, which contains one compound eating-or-drinking request. Its recorded compiler output has two once-only Watches, with instructions "the person is eating some food" and "the person is taking a drink from a cup". Both are live before the first response; after 9 s only the drinking Watch remains. Figure 9 shows Task Modification 154, with an update at 53 s from adding ingredients to pouring liquid into a jar, retaining the same Watch identity.

Model and runtime interfaces. Table 7 summarizes the same modules as Figure 2. In the implementation, the rules q_i in Equation 1 are stored as an event type (action onset, state entry, or a sustained condition), a reminder mode, and repetition settings.

Table 7: Component interfaces for the reserve-first path in Figure 2(b).

Role	Available input	Output and state update
Command Compiler	User command and existing Watch context	Task targets, reminder rules, and response instructions; Registry applies additions, edits, or cancellations.
Watch Registry	Watch specification and accepted updates	Persistent per-Watch instructions, evidence, and notification history.
Event-State Observer	Sampled views, observation IDs, and Watch target	Event phases and adjacent boundary labels, checked for agreement.
Response Proposer	Sampled frames and live Watch descriptions	Per-target Act/Mute proposal.
Relative-Peak Recovery	Eligible Watch, frame crops, and score history	Supplementary recovery evidence.
Notification Controller	Watch rules, three evidence sources, and history	Proposed history update and notification decision.
Validity Check	Request/observation identifiers and the plan's before-state	Request handler rejects stale requests; Registry checks current state.
Response Reservation	Accepted notification plan	Registry atomically applies the update and reserves the response.
Response Generator	Accepted context, instructions, and new reservation	Text or native speech for the reserved response.
Final Request Check	Generated answer and request/observation identifiers	Handler withholds stale answers; attaches current answers for release.

B.2 Evidence Normalization and Notification Control

Controller state and evidence. The controller stores event evidence, confirmation counts, and reminder history in $z_{i,t}$. Recovery evidence $\lambda_{i,t}$ adds a source-identified ENTER proposal alongside the observer's phase and boundary labels. Table 1 defines these labels and their correspondence; the following rules specify confirmation and observation handling.

How PREPARE and POST affect decisions. PREPARE preserves the observed lead-in to an action. In boundary recovery, four consecutive PREPARE observations allow one matching auxiliary ENTER to complete preparation-supported confirmation; a shorter preparation history uses the ordinary ENTER-confirmation threshold. POST, like ABSENT, contributes consistent non-active evidence to the rearm counter. Under the evaluated rule, rearm also checks the notification gap and a Mute from the Response Proposer observation. The short-action recovery branch applies to action-onset Watches that have a reminder available: a transition from ABSENT, PREPARE, or UNKNOWN into POST can supply recovery evidence when ordinary entry recovery is unavailable and the recovery gap is satisfied. It timestamps that proposal at the current observation. Repeated POST observations retain the same aftermath. This handles actions whose active phase fell between sampled frames.

Agreement and observation order. For a known state pair, an UNKNOWN boundary is insufficient; a different known boundary is disputed. If either state is UNKNOWN, a known boundary is disputed and an UNKNOWN boundary is insufficient. Only an exactly matching ENTER/EXIT prediction creates a confirmed boundary claim. These claims retain source observation IDs, timestamps, and supporting view IDs. The window-level agreement is disputed if any pair is disputed, otherwise insufficient if any pair is unknown, otherwise consistent. Observation IDs and timestamps identify repeated claims across overlapping windows; repeated evidence does not count as a new event. The normalizer retains confidence values and evaluates state–boundary agreement from the labels.

Minimal evidence I/O example. Consider target index 0 bound to a Watch at task revision 3, monitoring the onset of drinking from a cup. The request maps observation indices 0 and 1 to `obs.0` at 0 s and `obs.1` at 1 s, each with its causal image view. The following illustrative model return shows the schema:

```
{"results": [{"target_index": 0, "observations": [
  {"observation_index": 0, "phase": "PREPARE", "confidence": 0.9},
  {"observation_index": 1, "phase": "ACTIVE", "confidence": 0.9}],
  "boundaries": [{"from_observation_index": 0,
    "to_observation_index": 1, "transition": "ENTER",
    "confidence": 0.8}]}]}
```

The adapter binds the indexed return to the Watch, task revision, observation IDs, timestamps, and supporting views. It records `obs.1` as the first active endpoint. The observer inspects adjacent pairs independently of its phase predictions and uses UNKNOWN when visibility is insufficient.

Notification policy. Tables 8 and 9 give the policy parameters and decision order. Consecutive Mute proposals can restore reminder availability during an ACTIVE event. ABSENT and POST contribute to the non-active rearm count; PREPARE instead supports boundary recovery.

Complementary proposals and recovery. Confirmed entries, Act proposals, and recovery evidence are alternative inputs to the controller. The Act-proposal route initially requires three consecutive Act observations; after rearm, it requires one. A non-Act observation resets this count, and an observation gap exceeding 2 s restarts it. This temporal confirmation uses ordinary proposer outputs, with a separate confirmation rule for entry evidence.

Table 8: Evaluated notification-policy settings. These settings apply when the reserve-first path is selected. Act confirmation belongs to the complementary proposal route.

Policy property	Value or behavior
Act confirmation, initial cycle	3 consecutive Act observations
Act confirmation after rearm	1 Act observation
Consecutive Mute count	3 observations; counted while semantic state is active
Non-active evidence confirmation	1 observation
Maximum confirmation gap	2 seconds between adjacent observations
Notification gap	6 seconds for standard, proposer, and semantic recovery routes
Rearm rule	After the 6-second gap: consistent ABSENT/POST with a Mute observed, or 3 consecutive Mute proposals
Confirmed-entry recovery	Enabled during ACTIVE; requires a post-rearm entry
Non-active evidence before recovery	Not required after every prior notification
Entry seen after a reminder was used	Deferred until rearm; maximum age 4 seconds
Unanchored action proposal	4 consecutive Act observations if no prior ABSENT/POST or PREPARE support
Supplementary entry evidence	Enabled; includes source-scoped relative-peak evidence (Section 3.2), no direct Registry emission
Preparation support	4 PREPARE observations support auxiliary ENTER confirmation
Short-action recovery	Enabled for eligible transitions into POST

Table 9: Decision order in the evaluated notification policy.

Step	Operation under the evaluated configuration
1. Update evidence	Reject/reuse stale or replayed observations as specified by the controller; update response-proposal, phase, and boundary counters. Consecutive confirmation gaps are at most 2 s.
2. Reopen eligibility	For a consumed cycle, after the 6 s minimum gap, rearm if the consecutive-Mute count reaches 3, or consistent ABSENT/POST support reaches 1 and a Mute from the Response Proposer has been seen. Quiet observations may occur during ACTIVE.
3. Select candidate	Use an unseen confirmed entry while ACTIVE, or a complementary Act proposal after 3 consecutive initial confirmations or 1 after rearm (4 for an unanchored action onset). These are alternative sources. Entry recovery requires an entry at/after rearm. An entry deferred while another reminder is unavailable may be restored on rearm if still consistent and no more than 4 s old.
4. Recovery alternatives	Enabled post-event recovery can propose after an action was missed during a transition to POST. Auxiliary entry recovery requires source-identified recovery evidence plus the configured preparation/ENTER confirmations. Relative-peak evidence can be accepted for a non-active pair; recovery evidence alone cannot emit.
5. Gap and ownership	Eligible candidate sources must satisfy the 6 s notification gap. Otherwise hold. The controller proposes all changes; Registry checks apply rearm and emitting transitions, consuming the latter before response generation.

Deferred entries and recovery-evidence thresholds. In the evaluated policy's deferred-entry mode, an entry is stored after the previous reminder has been used only if it is fresh, no more than 4 s old, and supported by consistent evidence with current ACTIVE, previous stable ABSENT or POST, negative support since the last notification, and claim confidence at least 0.8. A stored, unconsumed entry may be restored on rearm if evidence remains consistent and its age is at most 4 s. Auxiliary entry recovery additionally requires valid source-identified recovery evidence, current boundary ENTER, and either 4 preparation confirmations or an ENTER streak of 2 for action-onset Watches or 1 for state-entry Watches, together with the recovery gap. The source-scoped relative-peak option can supply entry evidence for a non-active semantic pair (Section 3.2), retaining the original state and boundary labels alongside the recovery evidence.

B.3 Relative-Peak Admission and Recovery

Watch-local relative-peak recovery. The auxiliary probe maintains a score history for each Watch, task revision, event cycle, and evidence source. For a frame of width w and height h, nine overlapping crops of size $w/2 \times h/2$ are placed at the left/center/right and top/center/bottom positions. Each is enlarged to $w \times h$ and tiled into one 3×3 mosaic. One query to the same VLM asks whether any tile satisfies the Watch predicate, requesting a single Yes/No token and its first-token log probabilities. This local-view probe uses one additional forward pass per eligible observation. Valid Yes/No token scores are normalized over the reported token forms and passed to the temporal rule in Equation 2.

Probe selection and history resets. A probe runs when the Response Proposer returns Mute without a candidate and the complete Event-State batch identifies exactly one eligible Watch with ABSENT evidence and no candidate. Its tracker must be awaiting a new candidate after a non-active observation, with none pending. The supported targets are a quoted-text one-shot Watch or an active persistent Watch that has already fired and been rearmed. History continues only for the same Watch, revision, notification cycle, evidence source, and consecutive frame sequence; history resets when these conditions cease to hold.

Across all 90,445 evaluated frames per model, the probe was invoked 395 times for 8B and 440 times for 32B: 0.44% and 0.49%, respectively.

Controller integration and ablation. For recurring Watches, the peak supplies an auxiliary ENTER proposal $\lambda_{i,t}$ for the adjacent observation pair, including non-active pairs. It follows the controller's event, reminder-availability, confirmation, and gap checks, then Registry application before generation. One-shot Watches use their event-tracking and generate-first path.

Figure 5 illustrates the mechanism; Figure 3 in the main text shows a recorded 8B recovery, confirmed one second after the reference event. Table 4 reports a full-collection comparison with the entire Relative-Peak branch disabled: binary probes, score histories, and both one-shot and recurring peak candidates. PREPARE/POST recovery and ordinary Act proposals remain enabled. The OFF runs score 29.76/33.89 at 8B/32B, compared with the main-table Full reference of 30.58/34.54 (differences of $-0.82/-0.65$ pp).

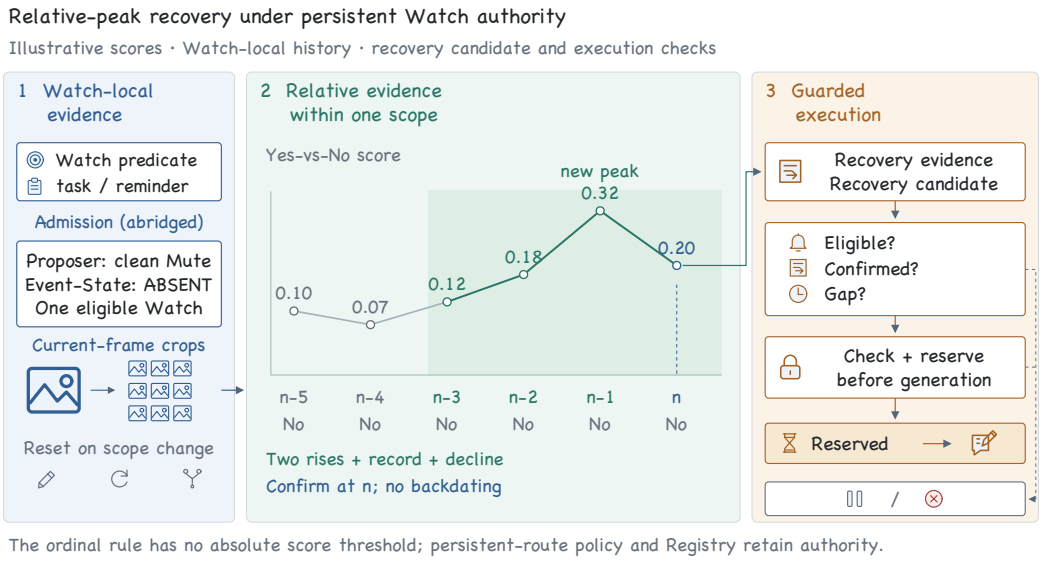


Figure 5: **Relative-Peak recovery mechanism.** Watch-local admission, relative-score history, and execution checks. The curve is illustrative; the pencil, loop, and fork denote task-edit, reminder-renewal, and evidence-source resets.

B.4 Transactional Execution and Recovery

Algorithm 1 shows the reserve-first path. The session checks the request identifiers and enters reservation without yielding to a same-loop command mutation. Transactions apply independently to each Watch; earlier successful reservations remain committed if a later CAS fails. The deployed adapter selects the generate-first path for once-only Watches and unavailable structured update plans, including recognition failure or observer shutdown. This path generates content before checking the current request and updating the Registry.

Unique response records. Within one Registry, durable writes through its update interface preserve at most one ready response record per reservation on the reserve-first path. Reservation stores its identity and expected snapshot atomically; a repeated plan reuses the record or fails the state check. Answer attachment changes that record from pending to ready, reuses an identical attached answer, and rejects a different answer. This property concerns response records; client delivery is handled by the transport layer.

Resuming pending responses. Reservation durably stores the response identity and updated reminder state. After an interruption, the low-level API can resume the existing pending record after reload. The UI leaves pending replay for this explicit recovery instead of starting a duplicate generation. If a task edit intervenes, the final request check withholds the outdated answer while retaining its reservation.

Algorithm 1 Reserve-first processing in a serialized session

Require: Registry $\mathcal{R}$, arriving commands, observations and video/task context X_t

On command: compile and apply a versioned Watch mutation
At observation t: copy monitored Watch states and request identifiers
Propose evidence from those states and X_t
Wait for the required evidence
Form the list of reserve-first update plans Π
if request generation, task revision, or current frame has changed **then**
    Discard this stale frame result
else
    $L \leftarrow [\,]$ ▷ New responses to generate; no yielding in this loop
    for each $\pi \in \Pi$ **do**
        $\rho \leftarrow \mathcal{R}.\mathrm{reserveCAS}(\pi)$ ▷ Also applies updates without a notification
        if CAS failed or ρ is replay of existing pending work **then**
            Record rejection or deferred replay; continue
        else if ρ creates new pending response work **then**
            Append ρ to L
        end if
    end for
    Generate answers for L from their committed contexts
    for each completed (ρ, y) **do**
        if request generation, task revision, and current observation still match **then**
            $u \leftarrow \mathcal{R}.\mathrm{attachAnswer}(\rho, y)$ ▷ Updated response record
            if u is newly publishable **then**
                Publish u
            end if
        end if
    end for
end if

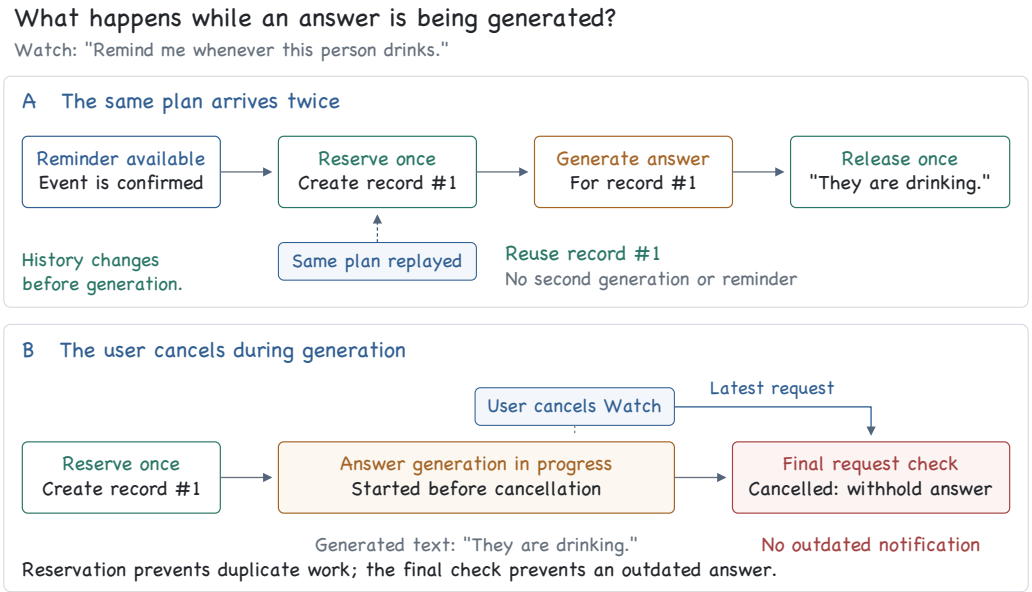


Figure 6: **Reservation and final request checking address different events.** Schematic reserve-first timelines. Top: replaying an already reserved plan reuses its record instead of starting another answer. Bottom: cancelling the Watch during generation causes the final request check to withhold the outdated answer.

C Additional Experimental Results

C.1 Relative-Peak Temporal Confirmation

The parameter scan varies the shape of the score trajectory used by Relative-Peak Recovery. R counts consecutive strict rises to a new maximum in the retained Watch-local history; D counts consecutive strict declines after that maximum. The candidate requires all $R + D + 1$ observations in this pattern to have categorical output No. Equation 2 uses the default R2D1. With $D = 0$, a candidate is proposed at the new high without waiting for a decline; $D = 1$ and $D = 2$ wait one and two eligible observations, respectively. The subsequent notification checks remain unchanged.

Table 5 covers all nine combinations of $R \in \{1, 2, 3\}$ and $D \in \{0, 1, 2\}$ on all 3,738 IPIBench episodes at both model sizes. Scores use the three-category Overall aggregation.

Among the reported scores, R2D1 is highest for 8B and R1D1 for 32B. The two settings score 34.54 and 34.61 at 32B, respectively.

C.2 End-to-End Latency During Silent Monitoring

We measure WatchWeave with remotely served Qwen3-VL-8B-Instruct on nine IPIBench episodes (246 sampled frames), using the browser's sequential 1 Hz schedule. The Response Proposer and Event-State Observer use non-streaming completion requests. Latency runs from uploading a prepared frame to receiving the complete application result, including remote model calls, local control, and client–server communication. Camera capture and JPEG encoding precede this measurement. The client uses one active session and one in-flight frame; model requests use the remote multimodal chat API with no replicated requests. The service's accelerator and serving precision were not recorded.

Table 10 reports steps that invoke perception but generate no user-facing response. Ordinary Mute excludes steps with an additional recognition/recovery check (139 frames); All Mute includes those steps (141 frames). P50 and P95 are the median and 95th-percentile latencies; ≤ 1 s is the fraction completed within one second. The real-time factor (RTF) is mean processing latency divided by the 1-second input interval.

Table 10: End-to-end latency during silent monitoring with WatchWeave (8B). Latency statistics are in seconds; RTF is dimensionless.

Path	Frames	Mean (s)	P50 (s)	P95 (s)	≤ 1 s	RTF
Ordinary Mute	139	0.660	0.544	1.303	79.9%	0.660
All Mute	141	0.740	0.548	1.308	78.7%	0.740

D Qualitative Examples of Streaming Interaction

Case selection and reading convention. The examples cover cancellation, concurrent targets, edits, intervening questions, and repeated reminders. All use Qwen3-VL-8B, including both sides of the Base comparisons. Times follow the video timeline and reference times come from benchmark annotations. Images are original sampled frames; requests are labeled paraphrases and responses are recorded text or labeled excerpts. The cancellation and interleaved-question cases use the generate-first path.

Comparison with the reference baseline. Figures 8, 12, and 13 compare the released IPIBench reference pipeline (Base) with WatchWeave using Qwen3-VL-8B on the same episodes and timing rule. Each system uses its own prompts and input configuration. The selected examples illustrate early, missed, and late Base triggers for which WatchWeave places all required responses within ± 1 s of the references. These WatchWeave traces use the generate-first path.

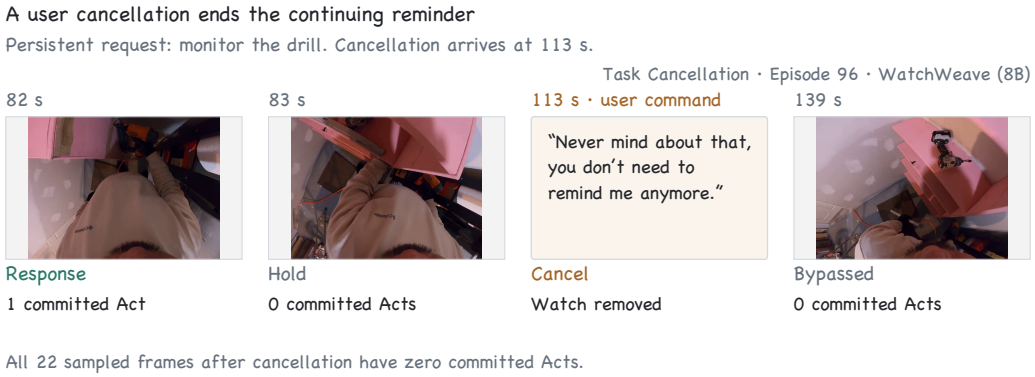


Figure 7: **Recorded cancellation in streaming video.** The IPIBench response at 82 s precedes the user's cancellation at 113 s; the later sample contains the target but causes no new response.

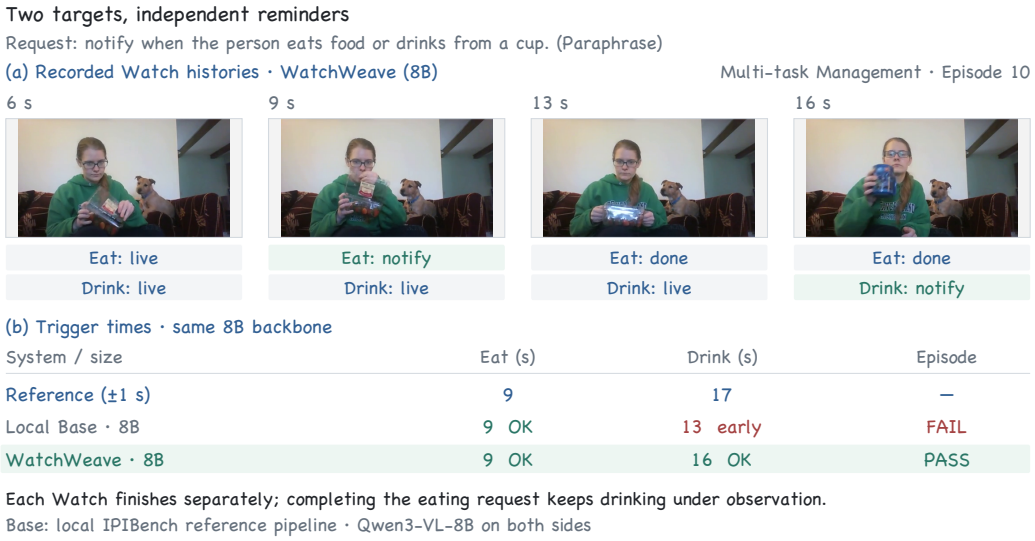


Figure 8: **Concurrent targets: independent histories and response timing.** Multi-task Management 10. (a) The eating Watch completes at 9 s; the drinking Watch remains live until 16 s. (b) With reference times 9 and 17 s, local Base (8B) triggers at 9 and 13 s, whereas WatchWeave (8B) triggers at 9 and 16 s. Both panels show the same 8B WatchWeave run. Base denotes the released IPIBench reference pipeline.

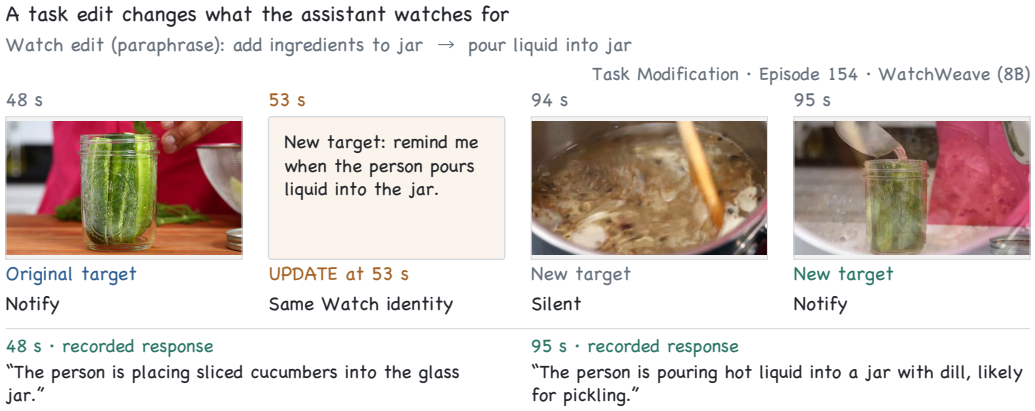


Figure 9: **A replacement request changes the monitored target.** Task Modification 154 updates the same Watch identity at 53 s, changing the target from adding ingredients to pouring liquid into a jar. The recorded responses at 48 and 95 s match the reference event times. The 53 s card marks the request edit.

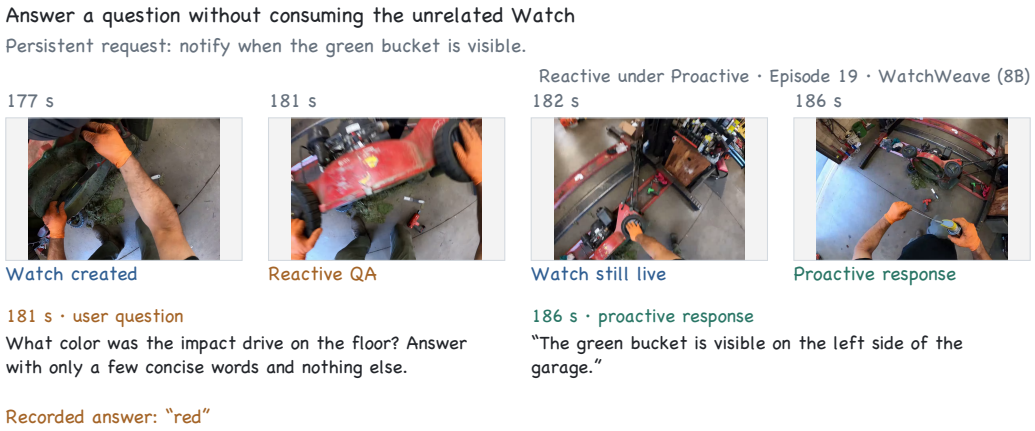


Figure 10: **An immediate question leaves an unrelated Watch active.** Reactive under Proactive 19 answers the 181 s color question with "red"; the green-bucket Watch remains live and responds at 186 s, achieving an episode score of 1.

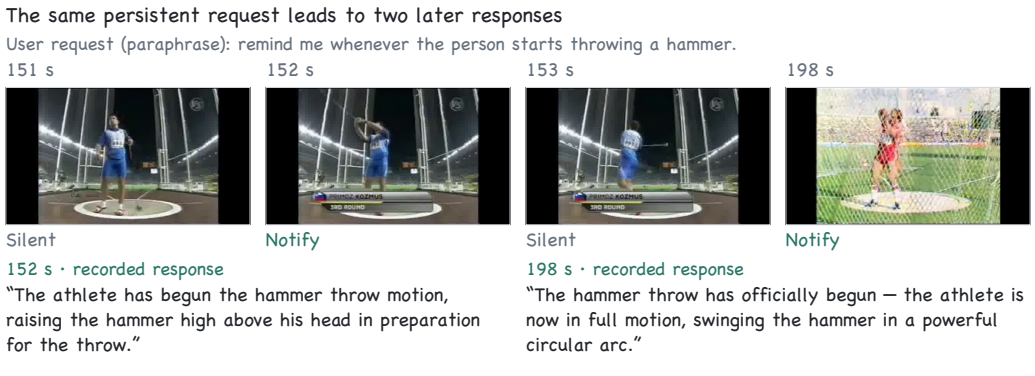


Figure 11: **Two responses to one continuing sports request.** Repeated Proactiveness 35 monitors the start of a hammer throw. The same Watch produces responses at 152 and 198 s, matching both reference times. The 151 and 153 s frames remain silent.

A changed request: the new target is missed by the base

Request (paraphrase): scoop rice; at 1378 s, change the target to removing the pot from firewood.



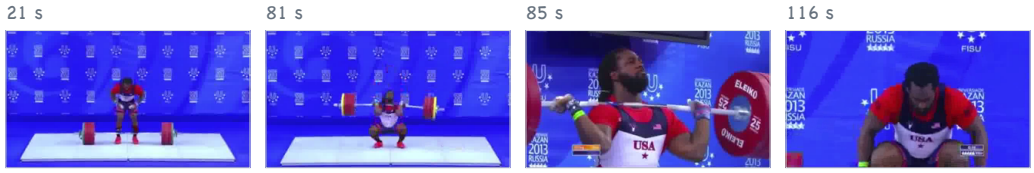
Trigger time (s)	Scoop rice	Remove pot	Episode
Reference (±1 s)	1373	1472	–
Base	1374 OK	miss	FAIL
WatchWeave (ours)	1373 OK	1473 OK	PASS

Base misses the new target. WatchWeave responds at 1473 s (reference: 1472 s).

Figure 12: **Same-backbone comparison: a new target after a changed request.** Task Modification 79 changes the request at 1378 s from scooping rice to removing the pot from firewood. The base records 1374 s and a missing second event (-1); WatchWeave records 1373 and 1473 s against references 1373 and 1472 s. In this trace, the old Watch has completed and the new request creates another Watch. The 1378 s image marks the command's media time.

Repeated events: three valid triggers from one continuing request

Request (paraphrase): remind me whenever the person starts a clean and jerk.



Trigger time (s)	Event 1	Event 2	Event 3	Episode
Reference (±1 s)	21	81	117	–
Base	miss	85 late	miss	FAIL
WatchWeave (ours)	21 OK	81 OK	116 OK	PASS

Base misses events 1 and 3; event 2 is late. All three WatchWeave triggers pass.

Figure 13: **Same-backbone comparison: repeated events under one request.** Repeated Proactiveness 2 has reference times 21, 81 and 117 s. The base records $[-1, 85, -1]$: two missed events and a four-second late trigger. WatchWeave records $[21, 81, 116]$ using the same continuing Watch, satisfying all three timing checks. Its actual outputs are three `<attention>` tokens.