EVISTREAM: EVIDENCE-ALIGNED AUDIO-VISUAL DECISIONS OVER PARTIALLY OBSERVABLE VIDEO STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) could enable assistants that continuously monitor video streams and respond when a user-specified condition is met. Yet recognizing the current scene is insufficient for deciding when to respond: some events depend on earlier observations, while spoken evidence becomes available only after transcription. Under bounded context, a streaming assistant must preserve relevant task progress and determine whether the available evidence supports a response. We introduce **EviStream**, a training-free framework for evidence-aligned audio-visual decisions over partially observable video streams. Its central idea is to manage observed evidence, persistent task progress, and temporary model judgments separately. A bounded KV cache retains recent visual and transcript evidence, while a temporary inference branch produces compact, evidence-indexed observations. A persistent state tracks established prerequisites and consumed observations to determine whether to wait or trigger a notification. Incremental speech transcripts are incorporated as they become available, and answers are generated from the frozen triggering context. Using frozen Qwen3-VL-32B-Instruct, EviStream achieves a joint timing-and-content score of **25.3** on all 774 StreamArena Pro questions, compared with 4.7 for direct window monitoring with the same backbone. Removing persistent state, audio, or sliding KV reduces the score to 10.3, 22.1, and 14.9, respectively. These results support the effectiveness of coordinating evidence retention and task progress for proactive streaming responses without task-specific fine-tuning.

1 INTRODUCTION

Streaming assistants receive observations continuously but must respond selectively. A user may request a notification when a condition is met, without supplying another prompt at the event. Under bounded context, the system must organize available evidence, preserve relevant task progress, and decide when to commit a response. Figure 1 illustrates two reasons that a current image alone is insufficient: a person's return depends on observed departure, while a spoken price becomes usable only after its transcript arrives. These cases motivate explicit management of observations and decisions throughout the stream.

This problem has two interacting sources of difficulty. First, observations are partial. Sampled frames omit intervening events; occlusion is not evidence of absence; speech becomes usable only after recognition. Second, observations and interpretations have different lifetimes. A recent frame is directly inspectable evidence. A previously confirmed departure is a historical state. A generated hypothesis is neither a new frame nor a new utterance. Treating these objects uniformly obscures which information supports an irreversible notification.

Streaming models reuse computation through compact key–value (KV) caches. StreamingLLM preserves attention sinks (Xiao et al., 2024), while StreamingVLM and ReKV reuse visual KV (Xu et al., 2026; Di et al., 2025). Yet SimpleStream's recent-frame baseline can match elaborate streaming methods on existing benchmarks (Shen et al., 2026), motivating evaluation that distinguishes current-scene recognition from sustained monitoring. Active-response systems learn when to speak or combine memory with response triggering (Liu et al., 2026; Azad et al., 2026; Xie et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

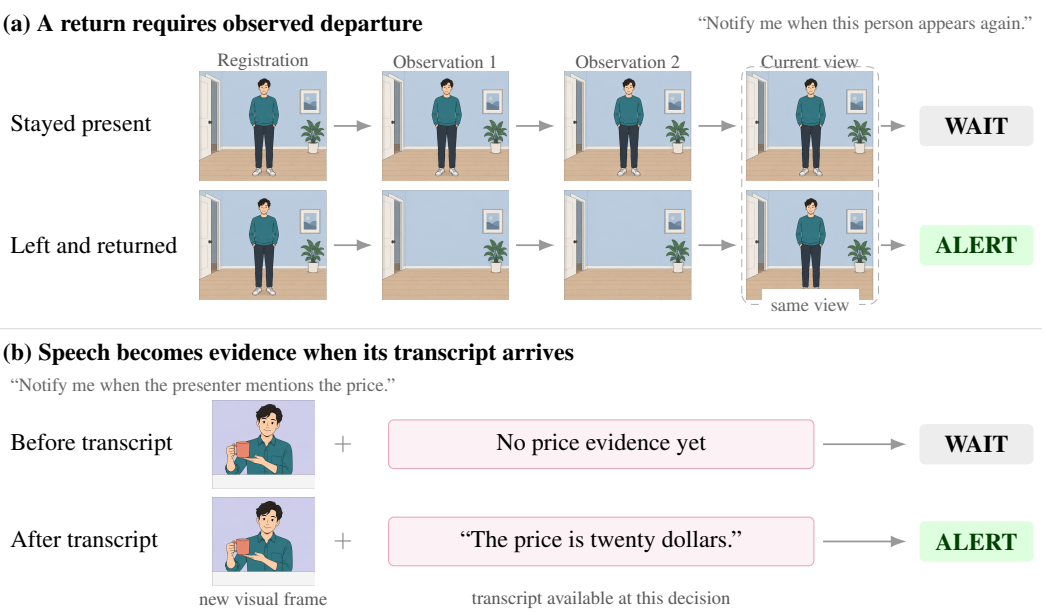

Figure 1: **Task state and arrived speech complement the current view.** (a) Reappearance requires prior departure, established by two absence observations at least 0.5 s apart. (b) Each decision includes a new visual frame; arrived price evidence changes the response despite a similar view. Both panels use illustrative scenes and assume correct recognition.

These advances motivate an explicit interface: *what should persist after a model judgment, and what must be established before a response is emitted?* A bounded cache alone does not specify that interface; an unconstrained textual summary does not enforce it.

EviStream separates the streaming loop into three roles. A persistent evidence cache stores the task prefix and recent observations. A temporary branch interprets unconsumed frames in the available audio-visual context and emits target-and-phase labels. A compact decision state records established progress and consumed observations; a deterministic reducer then produces a wait, state update, or alert. Query/output KV is released after decoding, while reduced state persists outside the cache. Earlier interpretation remains available as explicit, fallible state without being reinserted as observed evidence.

This interface also separates notification from answer generation. On an alert, the system notifies and answers from the frozen dispatch context. Evidence source, dispatch, notification, and answer completion remain distinct timestamps.

Our contribution is an explicit *observation-to-state-to-action interface*: persistent task progress and once-per-source-time frame consumption accompany disposable query/output KV. Three shared policies—event, change, and reappearance—specify the required progress, while causal audio admission incorporates asynchronous speech. Historical prerequisites are one application of the interface. We instantiate it with frozen Qwen3-VL-32B-Instruct (Bai et al., 2025) and streaming Paraformer (Gao et al., 2022), without task-specific training.

On StreamArena Pro (Zhang et al., 2026), credit requires correct content with a triggering observation inside a narrow event-relative window. Across all 774 questions, EviStream scores 25.3 under the trigger-frame metric, versus 4.7 for a direct window-monitoring control using the same frozen backbone. The published StreamMind system scores 11.6 with a different backbone. Removing persistent state, audio, or sliding KV lowers our score to 10.3, 22.1, or 14.9. These ablations test the complete state procedure, speech pathway, and cache reuse; horizon and task slices locate their gains and remaining challenges.

2 PROBLEM FORMULATION

At stream time t_0, a user registers a question or notification condition q. The system observes sampled video frames and stable automatic speech recognition (ASR) text.

An evidence record is

$$e_i = (\mathrm{id}_i, m_i, s_i, a_i, r_i, x_i), \qquad m_i \in \{V, T\}, \quad s_i \le a_i, \tag{1}$$

where s_i is source time, a_i is availability time on the stream-aligned runtime clock, r_i is the revision, and x_i is a frame or transcript block. Source time denotes where evidence came from; availability time denotes when inference could use it. For the current ASR wrapper, transcript source time is the endpoint of the consumed audio block, not a word-level alignment.

At dispatch time d_k, the retained evidence $\mathcal{R}_k$ is processed with compact task state S_k to produce action u_k and the next state:

$$\mathcal{R}_k \subseteq \{e_i : a_i \le d_k\}, \qquad (u_k, S_{k+1}) = F(q, \mathcal{R}_k, S_k), \quad u_k \in \{\textsc{wait}, \textsc{update}, \textsc{alert}\}. \tag{2}$$

A state update does not notify the user. The first alert stops monitoring and initiates an optional answer. For an emitted alert, n_k is the local timestamp read after decoding and passed to commit, before validation and the notification callback; it does not measure external user delivery. When the answer starts and completes successfully, its start b_k and completion c_k satisfy

$$d_k \le n_k \le b_k \le c_k. \tag{3}$$

The event's cited source time may precede all four. Relative notification error is $n_k - t^\star$ for reference event time $t^\star$; answering overhead is $c_k - n_k$. These quantities measure different delays.

Partial observability refers to sampling, limited retained context, uncertain recognition, and delayed evidence. We do not assume access to unobserved events. Nor does dispatch-time causality imply source-time isolation: all retained observations available at d_k may influence a label assigned to an earlier frame. The model is instructed to interpret each frame at its own source time, but the implementation does not impose a separate visibility mask for each frame. Semantic correctness and temporal correctness must therefore be evaluated in addition to checking causal availability.

3 EVISTREAM

3.1 FOUR OBJECTS WITH DIFFERENT LIFETIMES

Figure 2 summarizes the architecture. The fixed prefix P contains the original question, a compact task specification, and a reference image available at registration. The dynamic evidence stream contains recent visual records R and stable transcript records T. The task state S contains confirmed progress, a small unconfirmed candidate, a source-time cursor, and evidence identifiers. The temporary branch E contains the current evidence index, a compact rendering of S, the observation query, and generated labels.

Only P, R, and T are retained as evidence KV. The decision state is stored separately and rendered into the next temporary query. Query and label KV are discarded when E closes; decoded text remains available to the reducer and logs. Each state transition is therefore explicit and linked to supporting observations. State remains a fallible interpretation: an erroneous label can produce an erroneous transition.

The implementation retains at most four dynamic frames within a twelve-second source-time window, plus a fixed registration reference. Stable transcripts occupy at most twenty seconds and 512 tokens. The total model-context budget is 4096 tokens, including the fixed prefix, evidence, temporary query, and reserved decision output. Eviction removes whole records under these limits; Appendix B specifies the order and overflow handling. The bounds cover retained payload, model context, and compact decision state; historical identifier bookkeeping and persisted logs can grow over time.

3.2 INCREMENTAL KV AND TEMPORARY OBSERVATION DECODING

New admissible evidence is encoded into the persistent KV cache (Figure 2b). Ordinary eviction removes expired or over-budget evidence blocks while preserving the fixed prefix and surviving

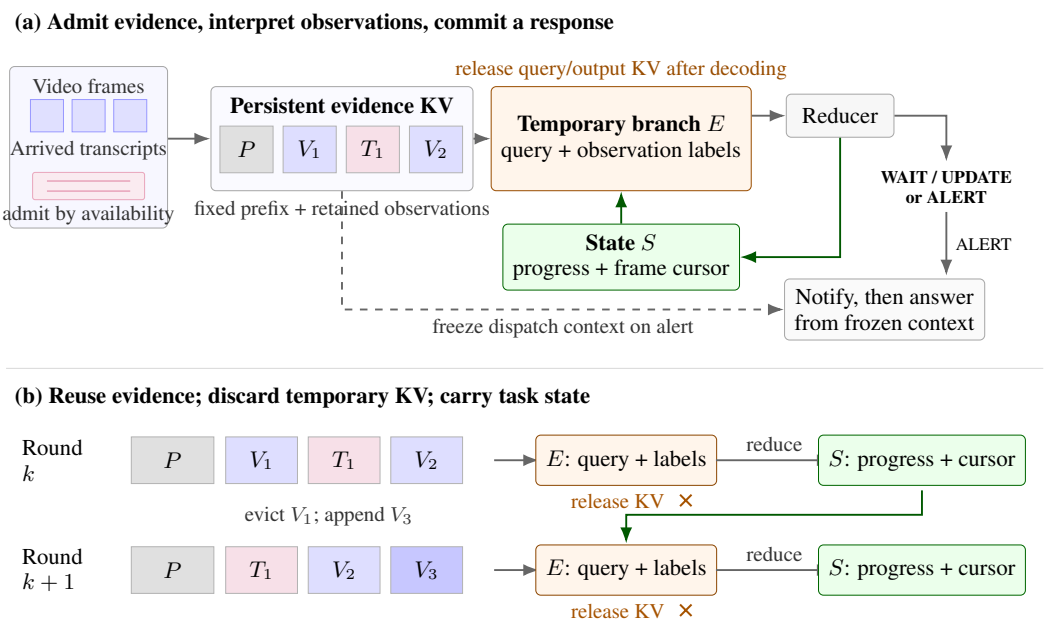

Figure 2: **Evidence, temporary inference, and task state have distinct lifetimes.** (a) Arrived evidence updates KV. E decodes labels for unconsumed visual slots; with none, decoding is skipped. The reducer updates S or alerts, and answers use frozen dispatch context. (b) Evidence KV is reused, query/output KV is released, and S persists. Cache strips are schematic; arrows denote dependencies, not latency.

cached tensors. Surviving representations may indirectly encode earlier observations that they attended to. Thus, eviction limits directly retained context without guaranteeing complete erasure of historical influence.

Logical positions increase between rebuilds and survive ordinary eviction. Approaching the position boundary or accepting a transcript correction triggers a rebuild from P and retained records. Because survivors can retain indirect history, recomputation changes the inference procedure as well as its cost. Appendix B specifies the update order.

At dispatch, E opens a private cache container over the retained context. Its query asks for a label for each previously unconsumed visual slot in source-time order. A registration slot is included until its anchor has been checked. The branch initially shares the retained history tensors; query and output appends belong to the private branch. Greedy decoding is constrained to the observation grammar, and the container is released after decoding even if validation fails. Persistent KV thus excludes query and output tokens.

For a dynamic visual slot j, the allowed label is

$$o_j \in \{\texttt{unknown}, \texttt{other}\} \cup \{\texttt{match}/p \mid p \in \Phi_q\}, \tag{4}$$

where Φ_q is the task's phase vocabulary, listed in Appendix A.1, and $\texttt{match}/p$ denotes the literal label for phase p. Such a label asserts that the observation supports that phase for the requested target. In particular, `match/absent` requires evidence of the known target's departure or absence, not merely a failure to recognize it. Identity and phase are judged by the same model. `unknown` covers insufficient evidence; `other` identifies a different target. The separate registration slot also permits `match/base`, which can establish the anchor and initial presence but cannot itself trigger an alert. Outputs are indexed labels, without explanations or model-authored alarm commands.

Transcripts inform visual judgments without creating independent observation slots. A dispatch with no unconsumed visual slot updates context without semantic decoding; late text does not reopen consumed frames. Thus, transcript arrival alone does not supply a new decision opportunity.

3.3 FROM OBSERVATIONS TO PERSISTENT PROGRESS

A lightweight question-only compiler selects a shared event, change, or reappearance policy through hand-written English phrase rules, with a generic event fallback. These policies express predicate satisfaction, change relative to a baseline, and return after departure. The compiler does not consult reference answers or event timestamps; target attributes remain in the original question. Phase labels and speech hints guide the VLM's interpretation, while confirmation rules govern commitment. Appendix B specifies the rules separately from the evidence/cache interface.

The reducer validates that all requested slots appear exactly once and follow the allowed vocabulary. It then processes their labels in source-time order. A cursor prevents previously consumed frames from counting again, and distinct records with the same source time count as one observation. The first supported alert wins within a batch; later labels are validated but do not advance that alert's state. Invalid observations produce no alert and preserve the previous state.

Table 1 specifies the transitions. Confirmed departure can remain in S after its supporting frames expire, enabling a later reappearance alert. One ambiguous view cannot establish departure. Change uses the registration anchor; event conditions can fire on one matched phase.

The state records evidence identifiers, source times, and revisions for confirmed transitions. Historical provenance can outlive the associated pixels; it is not a substitute for a retained image. Ordinary `unknown` or `other` labels clear an unconfirmed candidate while preserving confirmed history. A non-change phase clears a change candidate; an observed presence clears an unconfirmed departure. Accepted transcript correction is a stronger event: the runtime conservatively resets semantic state and rebuilds retained context instead of assuming prior interpretation is still valid.

3.4 CAUSAL AUDIO ARRIVAL AND COALESCED DISPATCH

Speech and visual observations become available on different clocks: a spoken cue is usable only after transcription. Our evidence interface connects streaming Paraformer (Gao et al., 2022) to the bounded cache and decision lifecycle. Each text block carries an identifier, its consumed-audio endpoint, and its availability time after recognition. The endpoint is not an exact word timestamp. The interface preserves committed recognizer text and excludes uncommitted hypotheses. Our adapter receives 16 kHz mono PCM in 200 ms chunks and buffers them into 600 ms working chunks before invoking the existing streaming recognizer.

A bounded inbox coalesces visual and transcript arrivals over 120 ms before dispatch. Either modality can make a batch eligible, with a one-second idle-wait fallback. The temporary branch interprets unconsumed visual observations with available transcripts; the reducer records progress in S. Transcript-only batches update context without creating visual observation slots. These buffering settings do not bound ASR, queueing, or model inference time.

Decision submission checks the evidence version used during inference, rejecting stale decisions after a correction or deletion. The evaluated Paraformer path emits immutable blocks; Appendix B describes the runtime's revision support. Coalescing combines arrivals into fewer dispatch opportunities; its separate effect on latency and accuracy is not isolated by the audio ablation.

Table 1: **Implemented transition rules.** Confirmation is conditional on the task family, rather than a fixed delay imposed on every event. All dynamic observations follow registration.

Family	Condition for notification
Event	One target-matched `met` label, or `caption` for a speech-dependent condition.
Change	An identifiable registration anchor, followed by two `changed` observations at source times at least 0.5 s apart.
Reappearance	Prior presence, then two `absent` observations at source times at least 0.5 s apart, then a later `present` observation.

3.5 NOTIFY FIRST, ANSWER FROM THE TRIGGERING SNAPSHOT

On an alert, the reducer commits the task state and the runtime emits the notification. It then stores a detached snapshot of the triggering dispatch KV, retained records, and state. Answer generation uses this snapshot and the original question, with an output cap of 128 tokens subject to remaining context capacity. It receives no evidence that arrives after the triggering dispatch. The snapshot includes the entire retained dispatch context, not only the cited evidence identifiers or the source-time prefix ending at the cited frame.

The answer request asks for a direct response, acknowledging insufficient context when necessary. Answer failure cannot retract an emitted notification; the two outcomes are recorded separately, so a plausible answer cannot erase an early or incorrect alert.

4 EXPERIMENTS

4.1 EVALUATION PROTOCOL

All experiments reported here use the complete StreamArena Pro task set (Zhang et al., 2026): 774 questions across 237 source videos. Advance instructions and annotated registration/event times enable joint evaluation of autonomous response timing and content. The complete method and three ablations share questions, registration times, reference frames, stopping boundaries, and frozen Qwen3-VL-32B-Instruct. Video sampling is nominally 1 fps; the complete method uses sliding KV, persistent S, and causal audio scheduling with incremental Paraformer transcripts.

Monitoring horizons divide the set into 259 questions at most 30 seconds (L1), 422 between 30 seconds and four minutes (L2), and 93 beyond four minutes (L3). Prespecified question-based slices comprise 127 speech-dependent and 647 other questions, plus 27 change/reappearance questions overlapping that partition. All subgroup results come from the same full-set runs.

For EviStream and its component ablations, we use the official StreamArena Pro metric and Gemini 3.1 Pro evaluator, with temperature zero and the released proactive-event judging prompt. Credit requires correct response content and a triggering observation within $[-0.5, +2.0]$ s of the annotated event. EviStream assigns the trigger the cited visual slot's source time; that slot is interpreted using the retained dispatch context (Section 2). Answer-generation time is excluded, so this score does not measure end-to-end notification latency. All 774 questions remain in the denominator, including failed or invalid runs. Appendix C specifies the evaluator and separate local-clock diagnostics.

4.2 COMPLETE METHOD AND COMPONENT ABLATIONS

We test three interventions, each over all 774 questions. *Without persistent decision state S*, each dispatch resets both semantic progress and the consumed-frame cursor, then reconstructs progress from the current window. Retained frames can therefore be judged again. *Without audio*, the system disables audio decoding, ASR, transcript evidence, and audio-triggered dispatch while preserving the visual decision path. *Window KV recomputation* rebuilds KV from the fixed prefix and currently surviving evidence at each dispatch, retaining the same state policy and context limits.

Table 2: **Component ablations on all 774 StreamArena Pro questions.** L1/L2/L3 and Avg report the official score using the official Gemini evaluator; strata contain 259/422/93 questions and Avg is question-weighted. No notification (%) counts runs with no emitted notification by observation end, separately from the official score.

System	L1	L2	L3	Avg	ΔAvg	No notification
EviStream	38.2	21.3	7.5	**25.3**	–	27.0
Without persistent S	15.8	7.3	8.6	10.3	−15.0	63.3
Without audio	32.0	19.0	8.6	22.1	−3.2	37.0
Window KV recomputation	22.8	12.3	4.3	14.9	−10.4	40.7

The complete system satisfies both trigger-timing and content requirements on 196/774 questions (score 25.3; Table 2). A response whose triggering observation lies outside the accepted window

Table 3: **Prespecified slices of the same 774-question runs.** Cells show correct counts (scores); denominators appear in the row labels. Speech and other questions partition the full set; the 27 change/reappearance questions overlap that partition. Small state-slice differences are descriptive.

Slice	Complete	No S	No audio	Recompute
Change/reappearance (27)	3 (11.1)	1 (3.7)	3 (11.1)	2 (7.4)
Speech-dependent (127)	22 (17.3)	5 (3.9)	6 (4.7)	11 (8.7)
Other questions (647)	174 (26.9)	75 (11.6)	165 (25.5)	104 (16.1)

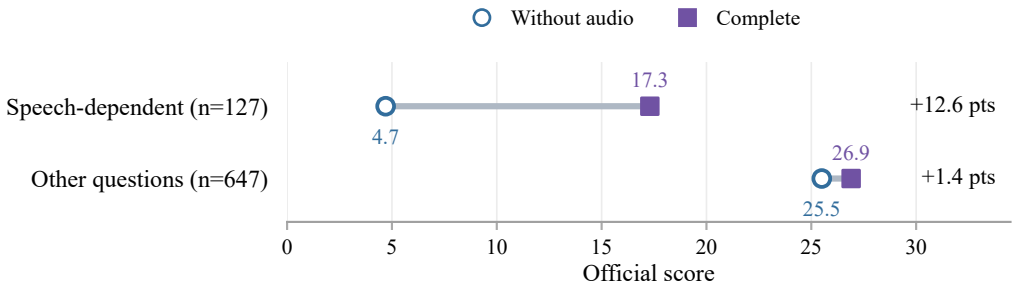

Figure 3: **Audio gains concentrate on speech-dependent questions.** Complete and audio-removal runs share the frozen backbone, questions, and visual protocol; n counts questions. Values reproduce Table 3, and gaps use its displayed scores. The comparison concerns the entire audio subsystem.

earns no joint credit, even when its content is correct. Each intervention lowers the full-set score. Table 3 breaks down the same runs by question group.

Persistent state improves full-set response quality. Removing S yields the largest decrease, from 25.3 to 10.3 (-15.0 points), while raising the no-notification rate from 27.0% to 63.3%. Resetting both semantic progress and the consumed-frame cursor changes which frames are judged again, potentially altering labels and processing time. This supports persistent progress and consumption history as an integrated decision procedure. On the 27 change/reappearance questions, the complete method meets the joint criterion on 3 questions versus 1 without S; this small difference is descriptive and does not isolate intermediate-state memory.

Audio helps most when the condition depends on speech. Figure 3 shows that on speech-dependent questions, the complete method scores 17.3 versus 4.7 without audio, compared with 26.9 versus 25.5 elsewhere. The larger difference on speech-dependent tasks supports using arrived transcripts to judge the requested event. Removing audio lowers the full-set score from 25.3 to 22.1 (-3.2 points). The intervention measures the audio subsystem jointly: recognition, transcript context, scheduling, and their processing cost.

Sliding KV supports response quality under bounded context. Window recomputation lowers the score from 25.3 to 14.9 (-10.4 points). Despite identical retention limits, surviving KV can carry indirect history from evicted observations. Rebuilding changes both the available representations and the workload, potentially affecting dispatch and notification times; this comparison does not separate their contributions. The result supports the evaluated cache procedure's response quality. Normalized GPU cost and end-to-end latency were not measured, so it does not quantify a speedup.

4.3 Published reference and deployed controls

Same-backbone and external controls. **Window-Qwen** uses the same frozen Qwen3-VL-32B-Instruct checkpoint as EviStream. Each decision recomputes the registration reference, up to four recent frames, and twenty seconds of causally available Paraformer transcripts, without persistent task state or evidence KV reuse. It directly predicts WAIT or ALERT on a nominal one-second polling schedule, subject to inference completion. Both Qwen systems use H100/BF16 with Transformers,

Table 4: **System comparison.** Joint timing-and-content scores. Rows above the separator are our experiments, including MiniCPM-o-4.5. Below the separator, StreamMind is a published result quoted from Zhang et al. (2026). Window-Qwen shares our backbone; JoyAI uses visual input only.

System	L1	L2	L3	Avg
EviStream	**38.2**	**21.3**	7.5	**25.3**
Window-Qwen	6.9	3.6	3.2	4.7
JoyAI-VL-Interaction	0.0	0.0	0.0	0.0
MiniCPM-o-4.5	6.3	7.4	4.6	6.7
StreamMind (Zhang et al., 2026)	16.6	9.5	7.5	11.6

without vLLM. Decision output caps are 128 tokens for Window-Qwen and 64 for EviStream; these are maximum lengths, not actual token counts.

We also evaluate **JoyAI-VL-Interaction** (Yao et al., 2026), a visual-only control using its native silence/response interface, nominal 1 fps input, and four-frame history, and **MiniCPM-o-4.5**, an external audio-visual model.

EviStream, Window-Qwen, and JoyAI cover the same 774 questions. Their scores use the first alert's cited visual-frame source time with the $[-0.5, +2.0]$ s window and the same content-adjudication rule. Response coverage remains a separate diagnostic under each system's native interface. Appendix C.1 summarizes the control settings.

EviStream scores 25.3 versus 4.7 for same-backbone Window-Qwen, a 20.6-point difference between complete procedures. State, cache reuse, scheduling, and decision format change together; the preceding ablations examine component interventions. Our external-model runs yield 6.7 for MiniCPM-o-4.5 and 0.0 for JoyAI, with different backbones and modality interfaces.

StreamMind's published score is 11.6 with Qwen3.5-397B-A17B. Its implementation was unavailable for reproduction at evaluation time, so we quote the paper's result. EviStream's 13.7-point system-level advantage is concentrated in L1/L2; L3 ties this reference.

Response coverage under the deployed procedures. Window-Qwen meets the joint criterion on 36/774 questions; JoyAI has no successes, with alerts tending to cite frames before the event. Their no-notification counts are 659/774 (85.1%) and 618/774 (79.8%), respectively, versus 209/774 (27.0%) for EviStream. Coverage is distinct from joint timing-and-content accuracy.

4.4 Response analysis

Monitoring horizon and unresolved notifications. The complete method alerts on 73.0% of questions; the remaining 27.0% have no notification by observation end. Emitted notifications may still be early, late, or incorrect, so this rate is not the complement of accuracy. Joint performance falls from 38.2 at L1 to 7.5 at L3. Removing audio or S yields 8.6 at L3: overall component gains do not extend uniformly to long monitoring horizons. Appendix D illustrates the distinct clocks in one recorded speech-dependent case.

5 Related Work

Streaming context and KV memory. StreamingLLM preserves attention sinks (Xiao et al., 2024); StreamingVLM uses short visual and longer textual windows (Xu et al., 2026). ReKV retains video KV for retrieval, while MuKV compresses multiple visual scales (Di et al., 2025; Xiao et al., 2026). Strong recent-frame baselines (Shen et al., 2026) and memory-budget analyses (Tian, 2026) motivate examining what should persist. EviStream separates retained evidence KV, disposable query/output KV, and reduced decision state.

Event memory and interaction. StreamChat combines hierarchical memory and parallel scheduling for multi-round dialogue (Xiong et al., 2025). StreamForest stores event trees, and OASIS retrieves hierarchical events on demand (Zeng et al., 2025; Liang et al., 2026). EviStream instead

specifies how fallible observation labels become persistent task progress and a consumed-frame cursor alongside recent evidence.

Deciding when to respond. ThinkStream couples reasoning with silence and response actions (Liu et al., 2026); StreamReady learns readiness, and StreamOV connects evidence-guided memory to triggering (Azad et al., 2026; Xie et al., 2026). JoyAI-VL-Interaction chooses silence, response, or delegation during visual input (Yao et al., 2026). Our frozen VLM supplies restricted observation labels to an external reducer that enforces explicit task-family transitions.

Audio-visual streaming. Omni-Streaming Thinking verifies pending claims against audio-visual evidence (Du et al., 2026); MiniCPM-o 4.5 supports proactive, full-duplex interaction (Cui et al., 2026). EviStream admits causally available Paraformer transcripts (Gao et al., 2022) into Qwen3-VL (Bai et al., 2025), without native acoustic tokens or speech output. Its audio ablation concerns the ASR-and-dispatch subsystem.

Streaming evaluation. StreamingBench includes proactive output (Lin et al., 2026); OVO-Bench tests clue revelation, recurring events, and steps (Niu et al., 2025); SVBench emphasizes temporal multi-turn dialogue (Yang et al., 2025). StreamArena evaluates continuous, interactive monitoring and reports StreamMind (Zhang et al., 2026). These benchmarks have different response opportunities and clocks. We evaluate EviStream on all StreamArena Pro questions with its official metric and Gemini pipeline; published StreamMind results provide a system-level reference, not a matched control.

6 Limitations

We evaluate one frozen backbone on StreamArena Pro. The compiler and response policies use hand-written English rules; transfer across paraphrases, languages, and datasets remains to be evaluated. Long-horizon and change/reappearance performance is discussed in Section 4.

Visual recognition and ASR errors can affect subsequent state updates, and transcript-only audio excludes nonverbal cues. Official trigger-time scores do not characterize end-to-end notification latency; latency distributions and answer-delivery time require separate evaluation.

7 Conclusion

EviStream separates retained evidence, persistent decision state, and temporary model judgments for causal streaming responses. On all 774 StreamArena Pro questions, it scores 25.3 under the trigger-frame metric, versus 4.7 for same-backbone direct window monitoring. The published StreamMind system scores 11.6 with a different backbone. Removing decision state, audio, or sliding KV lowers the score to 10.3, 22.1, or 14.9. These results support integrated evidence, state, and response management under the benchmark protocol, without task-specific fine-tuning.

Reproducibility Statement

Appendix A specifies the inference configuration and state transitions. The experiment protocol uses a fixed 774-question Pro manifest, separates inference inputs from evaluation labels, and preserves failures in the scoring denominator. The complete system and three ablations are reported using the official StreamArena Pro metric and Gemini evaluation pipeline. Published external results are identified separately from our runs.

AI Assistance Disclosure

AI tools assisted with literature search, method and implementation development, test and analysis scripts, manuscript drafting and revision, and conceptual illustration assets.

REFERENCES

Shehreen Azad, Vibhav Vineet, and Yogesh Singh 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*, pp. 40494–40504, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL Technical Report. arXiv preprint arXiv:2511.21631, 2025. URL https://arxiv.org/abs/2511.21631.

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. 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 *International Conference on Learning Representations (ICLR)*, pp. 42115–42127, 2025.

Enjun Du, Siyi Liu, Ziyu Zheng, Jingyu Li, Yiwen Guo, Yongqi Zhang, and Difan Zou. Omni-Streaming Thinking. arXiv preprint arXiv:2609.15128, 2026. URL https://arxiv.org/abs/2609.15128.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. Paraformer: Fast and Accurate Parallel Transformer for Non-autoregressive End-to-End Speech Recognition. In *Interspeech 2022*, pp. 2063–2067, 2022. doi: 10.21437/Interspeech.2022-9996.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. OASIS: On-Demand Hierarchical Event Memory for Streaming Video Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2821–2831, 2026.

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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 12147–12151, 2026. doi: 10.1109/ICASSP55912.2026.11463959.

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. In *Computer Vision – ECCV 2026*, volume 17038 of *Lecture Notes in Computer Science*, pp. 21–38. Springer, 2026. doi: 10.1007/978-3-032-37447-9_2.

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*, pp. 18902–18913, 2025.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A Simple Baseline for Streaming Video Understanding. arXiv preprint arXiv:2604.02317, 2026. URL https://arxiv.org/abs/2604.02317.

Yixian Tian. How Well Can Your Video Model Remember? Measuring Memory-Budget Trade-offs in Long Video Understanding. arXiv preprint arXiv:2606.20726, 2026. URL https://arxiv.org/abs/2606.20726.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Junbin Xiao, Jiajun Chen, Tianxiang Sun, Xun Yang, and Angela Yao. MuKV: Multi-Grained KV Cache Compression for Long Streaming Video Question-Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11381–11391, 2026.

Ming Xie, Zizheng Huang, Xudong Tan, Chao Wang, Xiangyu Zeng, Wenxiao Wu, Tao Chen, Limin Wang, and Yanwei Fu. StreamOV: Streaming Omni-Video Understanding via Evidence-Guided Memory and Response Triggering. arXiv preprint arXiv:2605.25621, 2026. URL https://arxiv.org/abs/2605.25621.

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. In *International Conference on Learning Representations (ICLR)*, pp. 69332–69351, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-Time Understanding for Infinite Video Streams. In *International Conference on Learning Representations (ICLR)*, pp. 61463–61475, 2026.

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. SVBench: A Benchmark with Temporal Multi-Turn Dialogues for Streaming Video Understanding. In *International Conference on Learning Representations (ICLR)*, pp. 42522–42556, 2025.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Xiangyu Zeng, Yifei Li, Haowen Hou, Zheming Liang, Congcong Wang, Kaiwen Tuo, et al. 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.

Xichen Zhang, Guankai Li, Yinghao Zhu, Shijian Wang, Sitong Wu, Shaozuo Yu, Meng Chu, Yuan Lu, and Jiaya Jia. StreamArena: Toward Continuous, Interactive, and Long-Horizon Agentic Streaming Video Understanding. arXiv preprint arXiv:2608.05703, 2026. URL https://arxiv.org/abs/2608.05703.

A Implementation Details

Table 5: Frozen inference configuration.

Component	Setting
Vision-language backbone	Qwen3-VL-32B-Instruct, frozen
Inference hardware / precision	NVIDIA H100 / BF16
Inference backend	PyTorch; Transformers 5.2.0; eager attention
Decision / answer decoding	Greedy; decision output grammar-constrained
Dynamic visual context	At most 4 frames, source age at most 12 s
Frame representation	448×448 pixels, nominal sampling 1 fps
Registration reference	Fixed frame available at task registration
Transcript context	At most 20 s and 512 tokens
Fixed prefix text cap	1024 tokens
State rendering cap	64 tokens
Temporary query / decision output caps	256 / 64 tokens
Total model-context cap	4096 tokens
Answer output cap	128 tokens, reduced if context capacity is smaller
Audio input	16 kHz mono PCM, 200 ms ingress chunks
ASR execution	Paraformer on CPU, 4 threads
ASR working chunk	600 ms; chunk setting $(0, 10, 5)$
ASR look-back	4 encoder / 1 decoder chunks
Arrival coalescing / idle-wait setting	120 ms / 1 s
Inbox capacity	128 records, 32768 text characters across active versions
Change / absence confirmation	2 observations; source-time separation ≥ 0.5 s

A.1 Observation vocabulary and initialization

Each dynamic slot permits `unknown`, `other`, or `match/phase`. Event tasks use {`pending`, `met`}; speech-dependent event tasks use {`pending`, `caption`}; change tasks use {`same`, `pending`, `changed`}; reappearance tasks use {`present`, `absent`}. A `caption` label means arrived transcript context supports the entire requested speech condition. It is not a label emitted on a transcript-only slot. An `absent` label requires affirmative evidence concerning the known target's absence or departure; occlusion, ambiguous identity, and sampling gaps map to `unknown`.

Initially the anchor is unchecked, the presence/departure/candidate records are empty, and the source cursor precedes the stream. Registration permits `unknown`, `other`, or `match/base`. A matched base establishes the fixed anchor; for reappearance it also establishes prior presence. An unknown/other base marks the anchor as checked but unidentified. Change then lacks its required anchor; reappearance can still establish prior presence from a later matched presence. Event tasks do not require an anchor. Ordinary later observations do not replace the registration baseline. Accepted transcript correction resets state, including the anchor-check flag.

A reappearance trace. Suppose the reference identifies the requested person. A matched presence preserves the prior-presence prerequisite. One unknown frame clears an unconfirmed absence candidate; it does not establish departure. Two later matched absence labels at source times at least 0.5 s apart establish departure. A subsequent matched presence completes the condition. If the person remains present throughout, no departure is established and no reappearance alert follows. This schematic illustrates the reducer under the stated observation labels.

Provenance and correction. The reducer stores compact identifiers, source times, and revision numbers for prerequisites. Current alert evidence identifiers refer to retained records; historical prerequisites are carried as provenance. They need not all remain in the visual window. The revision interface can reset the semantic state and reconstruct the cache when committed transcript evidence is corrected. Ordinary eviction and transcript correction therefore have different semantics.

Answer budgeting. After the alert, the answer query is evaluated against the frozen dispatch context. The permitted generation length is the minimum of the requested cap, remaining total context

capacity, and remaining position capacity. If no answer token fits, the answer is recorded as unavailable rather than silently admitting additional evidence or enlarging the budget. Notification validity and answer success remain separately observable.

B Compiler and Inference Interface

Ordered task routing. Routing lowercases the question, replaces double-quoted spans with a placeholder, keeps the text after the first `when` or `after` when present, and stops before `tell/told/remind/notify me` or a comma followed by `saying`. This separates the requested event from the wording of the eventual notification. The original question is preserved in the fixed prefix. On the resulting condition text, the first matching rule below selects the family:

1. A word beginning with `appear` and the word `again`, in either order: reappearance.
2. `taken away` or the word `removed`: change relative to the original object/location; reaching, lifting, and partial removal are pending.
3. `material` together with a word beginning with `chang`: change in the specified region; selection or preparation is pending.
4. `next` and `product`: change relative to the registration product; incidental visibility or announcement alone is pending. The task description requests transcript support when introduction requires speech.
5. `next part`, `next section`, or `part ends`: change in discourse progress, with transcript support required. A pause or a slide switch alone is pending.
6. Otherwise: generic event, with action-specific hints where applicable.

The generic-event speech flag matches `say/says`, `saying`, `mention/mentions`, `speaking`, `explaining`, a word beginning with `introduc`, or the phrases `class starts/part ends`. A speech event uses `caption` as its completion phase; a non-speech event uses `met`. Event hints require actual eating rather than food preparation, actual cooking rather than ingredients merely being visible, and simultaneous holding of the requested number/type of objects. Ordinal wording requests explicit ordinal evidence; the program does not initialize a global repetition count. Conjunctive wording requires all requested conditions on the specified target. These hints guide recognition; the reducer does not independently verify their semantics.

Prefix and query. The fixed text concatenates the observation rules, compiled task description, unchanged question, registration source second, index-mapping instructions, and allowed dynamic labels. A fixed registration image is assigned slot 0. Dynamic visual records are indexed from 1 in source-time/identifier order; transcript records are listed as audio-context identifiers. Only indices in `judge` require output. The temporary query, before the backbone's ordinary user-role chat template is applied, has this structure:

```
t=<absolute time> S=<compact state>
e=v1=<visual ID>,a=<transcript ID>,...  judge=<indices>
```

The compact state contains a phase (`watch`, `seen`, `away`, or `done`), anchor status (`unread`, `unknown`, or `known`), and any retained presence, departure, or candidate source time. Times in the temporary query are absolute video seconds. For example, a hypothetical reappearance query can request `judge=0,1` and receive `0:match/base;1:match/present`. It initializes the anchor/presence but does not establish departure. This string is a serialization example, not a model result.

The constrained decoder emits every requested `index:label` in order, separated by semicolons. The single token string `unknown` denotes unknown for every requested slot. The protocol prohibits explanations, future predictions, state-transition commands, and alarm commands. Its semantic instructions require matching the named target and modifiers, distinguishing actual actions from preparation, treating uncertain identity or occlusion as unknown, and avoiding retrospective relabeling of a frame merely because later context resolves the event. Speech must be supported by arrived transcript context, not inferred from lips, gestures, slide text, or unrelated speech. The current state

is explicitly described as fallible history. These instructions define the model's task; they are not semantic correctness guarantees.

Retention order. At dispatch, source-age expiration is evaluated relative to the current video-aligned runtime clock d_k: a visual record older than twelve seconds or a transcript older than twenty seconds expires. Among eligible pending and retained visual records, the latest four by source time and identifier are selected. Records are appended in availability order; before each new visual block is appended, excess oldest visual blocks are evicted. This allows the first new frame in a batch to attend to the evidence still retained at its own append step.

The whole-block fitting loop checks total tokens, decision-output reserve, and transcript tokens. When the transcript cap is exceeded, eviction candidates are transcripts only; otherwise all retained records are candidates. It removes the oldest candidate by source/availability time. An incoming block participates in eviction and can be logged as skipped. The fixed prefix is not evicted; if the prefix/query/reserve alone cannot fit, execution fails. Rebuilds start from the fixed prefix and reappend surviving records in availability order. Before an append, a rebuild occurs if the next logical position plus incoming length and the total-context reserve reaches the backend position limit; a second check covers query and output positions before decoding. Ordinary eviction preserves survivors' logical positions.

State commit and answer. The batch's evidence generation is checked before committing a decoded decision. Stale decisions are discarded. Valid observations are reduced in source order, at most once per source time; the first alert ends state consumption, while the complete returned sequence is still validated. On alert, the notification is emitted before the detached dispatch snapshot is used for answer generation. The answer prompt supplies the original question, alert evidence identifiers, state, and retained-evidence metadata, and requests one or two direct sentences without a reasoning trace. It instructs the model to acknowledge insufficient evidence. No subsequent observation is added to that answer context.

C Evaluation Protocol

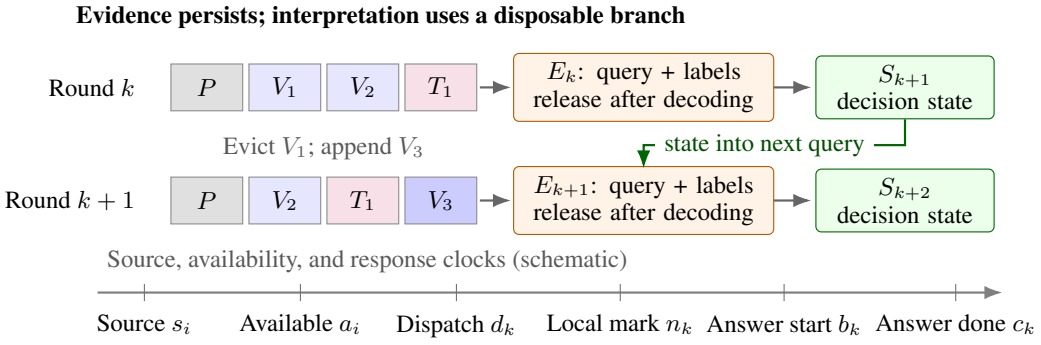

Figure 4: **Decision rounds and timestamp conventions.** Source time, evidence availability, dispatch, the local notification mark, and answer start/completion are distinct. The schematic does not measure latency or external notification delivery.

Replay clock. Let w_0 be the monotonic wall clock when replay starts at task registration t_0. The runtime uses $t(w) = t_0 + (w - w_0)$; producers wait until source data are due. Video source times, ASR availability, dispatch, notification, and answer timestamps are expressed on this video-aligned clock. Processing delay therefore contributes to notification and answer delay; replay is not advanced instantaneously after each model call. Model initialization precedes this replay interval and must be reported separately if included in total deployment cost.

Official Pro score. StreamArena defines the predicted trigger time t_i^{trig} as the video timestamp of the observation that triggers the alert, excluding subsequent response-generation latency (Zhang

et al., 2026). Let $t_i^\star$ be the annotated event time and $g_i \in \{0, 1\}$ be the Gemini content judgment. The joint outcome and score are

$$z_i^{\text{official}} = g_i \, \mathbf{1}\left[-0.5 \le t_i^{\text{trig}} - t_i^\star \le 2.0\right], \qquad \text{Score} = \frac{100}{774} \sum_{i=1}^{774} z_i^{\text{official}}. \tag{5}$$

Times are measured in seconds. All 774 questions are included; missing or empty responses, missed events, timing violations, and failed or invalid runs receive zero credit.

Gemini configuration and content criterion. We follow the released Pro judge configuration: `gemini-3.1-pro`, temperature 0, `maxOutputTokens`=10, and one vote per response, using the Gemini `generateContent` interface. The proactive-event prompt receives the question, reference trigger phrase, and model response. It accepts equivalent descriptions of the same event with the correct subject, object, and action; literal reproduction of the reference phrase is unnecessary. Incorrect facts, a different event, generic or hallucinated content, and refusals are judged incorrect. The requested output is `Correct` or `Incorrect`. The released client retries a failed request up to three attempts, with a 60-second request timeout; exhausting all attempts yields zero credit.[1]

Evaluator record fields. In the official schema, `answer_time_sec` stores the *reference event time*, not the model's answer time. The timing check uses `start_vid_ts` from the first entry of `e_proactive_msgs`; `model_answer` supplies that response's text. If `start_vid_ts` is absent, the released compatibility path uses `vid_ts`-2.0 s. This fallback does not widen the acceptance window. The trigger timestamp is distinct from the runtime's local notification mark n_i, answer start b_i, and answer completion c_i; those clocks remain separately logged.

Local notification diagnostics. For an observed notification, diagnostic error is $\delta_i = n_i - t_i^\star$: early if $\delta_i < -0.5$, within-window if $-0.5 \le \delta_i \le 2$, and late if $\delta_i > 2$. Runs ending without a notification are reported as no notification: the finite observation period cannot distinguish a later response from a permanent miss. Invalid execution is recorded separately. These timing categories do not determine semantic correctness. The no-notification column in Table 2 counts unresolved notifications at observation end over all 774 questions. For comparisons among our four arms, question identifiers, registration, reference, and stopping boundary are matched; the audio-removal arm necessarily has a different audio and transcript input trajectory.

C.1 Standalone control protocols

Window-Qwen uses the same frozen Qwen3-VL-32B-Instruct checkpoint as EviStream, reference plus four-frame visual input, and causally available Paraformer text. Each call recomputes context and returns a WAIT/ALERT decision with a 128-token cap. JoyAI-VL-Interaction uses its native silence/response interface with visual input only, nominal 1 fps feeding, and a four-frame history. These are complete deployed procedures rather than component removals from EviStream.

The joint outcome uses the visual source time cited by the first alert. Content adjudication applies the same rule across locally evaluated systems, with exact reference-phrase matches accepted automatically and other responses reviewed individually. All 774 questions remain in the denominator, including missing responses and invalid executions. Table 4 reports the control results under the trigger-frame metric. The separate coverage diagnostic counts whether each native interface emitted a notification before its observation ended. The precise serialization of control responses into the official export fields is not documented here.

[1] Released evaluator: StreamArena judge implementation.

D ILLUSTRATIVE EXECUTION TRACE

Figure 5 traces one completed question from the same 774-question main run. It connects retained evidence, temporary observation decoding, and the separately recorded notification and answer clocks.

Request (condensed): Notify me when the host says "close the front trunk and open the rear trunk."



Retained context V_{16}: source 2199.000 s



Cited frame V_{19}: source 2202.000 s

Last transcript arrival: 2201.775 s → Dispatch 2202.183 s → Local notification 2203.033 s → Answer complete 2209.552 s

Decoded label: `4:match/caption`

Notification: "The requested event has occurred."

Clock values are stream-aligned seconds. Layout is not a proportional time axis.

Figure 5: **A recorded speech-dependent decision.** StreamArena Pro video `0JPSHJUyaz0`, question 12. Both images were re-decoded from the original video at logged source times; they are not saved model-input tensors. V_{16} supplies context and V_{19} is cited by the alert. The last retained transcript block is T_{28}; its availability, dispatch, local notification, and answer completion are distinct. The selected trace illustrates execution, not an additional accuracy estimate or external delivery measurement.