UNDERSTANDING STREAMING MEMORY FOR PROGRESSIVE CONTEXT EXPANSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding requires models to reason over continuously growing history under strict online computation budgets. Existing memory systems rely on increasingly complex compression, organization, and retrieval mechanisms, yet their tightly coupled designs obscure the factors that truly govern performance and efficiency. We systematically revisit streaming memory, disentangle its key design choices, and derive principles for effective memory utilization. Based on these insights, we propose TRACE (**T**riggered **R**etrieval and **A**daptive **C**ontext **E**xpansion), a training-free progressive memory framework that optimizes Recent, Short, and Long memory levels and expands historical access according to predictive uncertainty. TRACE starts from low-cost recent context and introduces additional memory only when needed. On OVO-Bench, TRACE improves Qwen3.5-9B by 9.13 percentage points overall, with gains of 10.88 points on real-time perception and 6.81 points on backward tracing, while reducing per-query latency from 19.10 s to 1.86 s. Consistent gains on StreamingBench and across multiple VLM backbones demonstrate the effectiveness of selective and progressive memory access for balancing long-horizon reasoning and efficiency.

1 INTRODUCTION

Recent advances in Large Vision-Language Models (LVLMs) have substantially improved video perception and reasoning (Bai et al., 2025; Wang et al., 2025a). Emerging applications such as embodied intelligence and real-time interaction, furthermore, require models to process continuously growing video streams and answer queries at arbitrary future times (Chen et al., 2024; Niu et al., 2025; Qian et al., 2025). In such streaming settings, retaining and processing the entire history is computationally unsustainable and may also degrade inference by introducing excessive irrelevant context (Zhang et al., 2025; 2026a; Shen et al., 2026). Structured memory has therefore become a promising mechanism for compactly maintaining long-term video context through compression, organization, and retrieval (Di et al., 2025; Zeng et al., 2026; Zhang et al., 2026a).

Prior work typically addresses growing video history through tightly coupled memory systems, jointly designing how historical information is represented, compressed, organized, and retrieved, including implicit KV-based memory and explicit event-centric structures (Song et al., 2024; Zhang et al., 2025; Xie et al., 2026; Di et al., 2025). This coupling design makes it difficult to identify which components actually drive performance and efficiency, leaving memory optimization empirical and hindering principled improvements (Shen et al., 2026; Zhang et al., 2026a; Xie et al., 2026). We therefore revisit the design of streaming memory along three dimensions: *how historical information should be accessed* (**Access**), *what information should be exposed to the model once relevant history is accessed* (**Information**), and *when additional memory should be invoked* (**Invocation**).

Through a systematic study along these three dimensions, we obtain three design principles that guide the design of effective streaming memory. **(1) Access.** Under a bounded context budget, maintaining a fixed form of historical information is sufficient by default, with additional retrieved information introduced only when needed. **(2) Information.** Memory should preserve visual and temporal information directly, without introducing additional semantic compression. **(3) Invocation.** Memory should be invoked only when the current context is insufficient, predictive uncertainty provides a lightweight signal for determining when additional memory is needed.

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.

Building on these principles, we propose TRACE (**T**riggered **R**etrieval and **A**daptive **C**ontext **E**xpansion), a training-free progressive memory framework that expands historical access on demand. Rather than persistently loading the full memory, TRACE starts from a small recent context and progressively expands historical access based on predictive uncertainty. Specifically, when the recent context is uncertain, TRACE augments the context with a low-cost short-term window and a global summary. Only when the expanded context remains insufficient does TRACE retrieve long-term memory and expose the retrieved visual evidence with its temporal structure. This staged escalation reserves expensive retrieval and visual-context expansion for queries that require them, retaining access to long-term evidence while avoiding unnecessary computation.

Experimentally, we evaluate TRACE on OVO-Bench (Niu et al., 2025) across various VLM families. As a fully training-free framework, TRACE achieves better performance than its vanilla counterparts and other baselines, while substantially reducing inference cost, which is critical for continuous streaming deployment. Furthermore, we observe consistent gains on StreamingBench (Lin et al., 2026), suggesting robustness across benchmarks and model families.

Overall, our contributions are summarized as follows:

- We revisit streaming video memory through three fundamental questions and disentangle its design space into three dimensions, i.e., *access*, *information*, and *invocation*. Through systematic analysis of the key choices within each dimension, we distill practical principles for effective streaming memory design.
- Guided by these principles, we propose TRACE, an uncertainty-driven progressive memory framework that starts from recent visual context and expands memory access when needed. TRACE selects a balanced strategy for historical retrieval, establishes an efficient evidence-exposure paradigm, and enables historical evidence to enter the model context selectively and progressively rather than keeping memory persistently loaded.
- We validate TRACE across multiple VLM families on both OVO-Bench and StreamingBench. While training-free, TRACE improves question-answering performance while substantially reducing the computational overhead and inference latency associated with memory access, and demonstrates robustness across models and benchmarks.

2 Rethinking Memory In Streaming Video

Despite increasingly complex memory designs (Di et al., 2025; Zeng et al., 2026; Zhang et al., 2026a), SimpleStream (Shen et al., 2026) shows that a simple sliding window over recent frames can outperform several memory-based approaches. This finding suggests that improvements in streaming video understanding may not primarily stem from increasingly sophisticated memory mechanisms, but from more fundamental design factors. However, existing methods tightly couple multiple memory components within a single system, making it difficult to isolate the factors that actually drive performance. This motivates our systematic analysis of streaming memory.

Setup. We focus on the real-time and backward settings of OVO-Bench, with NExT-GQA (Xiao et al., 2024) providing complementary retrieval evaluation. Qwen3.5-9B (Qwen Team, 2026) serves as the default backbone. All evaluations follow the standard protocols. We use the most recent 8 frames sampled at 2 FPS as the no-memory baseline. All evaluations follow the standard protocols.

2.1 Access: How Should Historical Information Be Accessed?

In streaming settings, the available context budget is often highly constrained, making historical access fundamentally a trade-off between temporal coverage and visual details(Doorenbos et al., 2026; Shen et al., 2024; Chen et al., 2026). Existing memory mechanisms make different trade-offs between these two aspects. *Sparse Buffer* samples a longer temporal range than the local baseline at a lower rate, preserving direct visual observations but still providing limited temporal coverage, *Summary* provides global access to the accumulated history with broad temporal coverage but reduced information granularity due to compression, and *Retrieval* provides selective access to long-term history by selecting query-relevant historical segments but depends on whether gold evidence is successfully retrieved (Chatterjee et al., 2025; Liang et al., 2026). We empirically compare these paradigms in Table 1 and the detailed setup is provided in Appendix C.

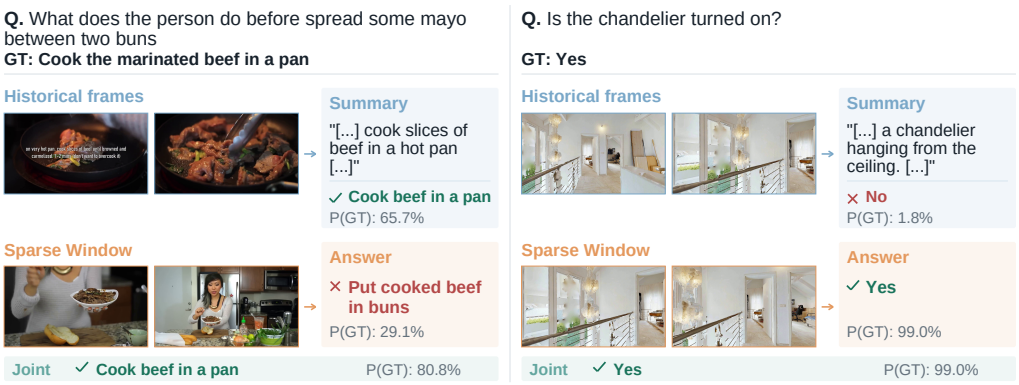


Figure 1: Case Comparison. The Sparse Buffer misses the answer-relevant moment (left), while the Summary loses critical visual details (right).

Table 2: Evidence presentation accuracy. Full results in Appendix B.

Frames	Captions	Time		Acc. (%)
		Implicit	Explicit	
–	–	–	–	42.47
–	✓	–	–	44.49
✓	–	–	–	52.58
✓	✓	–	–	51.24
✓	–	✓	–	53.48
✓	–	–	✓	**54.38**

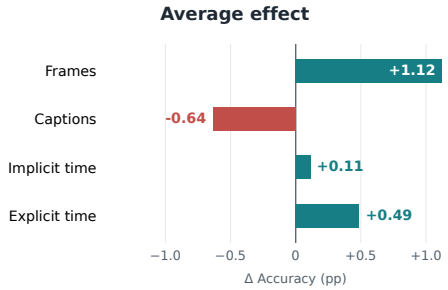


Figure 2: Factor effects averaged over paired settings differing only in the corresponding factor.

The results reveal a clear distinction between memory that can be maintained persistently and memory that should be accessed selectively. Among fixed configurations, combining *Sparse Buffer* and *Summary* achieves the best overall performance, substantially outperforming either paradigm alone. This suggests that recent high-fidelity observations and a compressed, broad-coverage history provide complementary information, as further illustrated in Fig. 1. Nevertheless, $W + S$ does not fully cover the information required for historical reasoning. Introducing Retrieval as an additional oracle choice yields a clear improvement, revealing that the bounded context still leaves an information gap. This is expected: Sparse Buffer is limited in temporal coverage, while Summary loses fine-grained visual details through compression, leaving some queries dependent on earlier fine-grained visual evidence.

Table 1: Results in different memory mechanisms under the same context budget. V:Vanilla Baseline. B: Sparse Buffer; S: Summary; R: Retrieval.

Configuration	EPM ↑	ASI ↑	Overall ↑
Individual paradigms			
V	52.86	60.81	55.51
B	55.56	64.19	58.43
S	56.90	64.19	59.33
R	55.22	51.35	53.93
Fixed combinations			
$B + S$	58.59	72.30	63.15
$S + R$	47.47	54.73	49.89
$B + S + R$	58.25	63.51	60.00
Oracle selection			
$\{B, S\}$	64.65	71.62	66.97
$\{B, S, R\}$	75.08	82.43	77.53
$\{B + S, R\}$	72.73	79.05	74.83

Retrieval hurts when persistently added to $W+S$, but remains beneficial under oracle selection. This contrast suggests that Retrieval is valuable not as a persistent augmentation to the bounded context, but as a complementary access mechanism for queries that require additional historical evidence.

This leads to our first design principle: **Under a bounded context budget, prioritize complementary local context and access long-term history only on demand.**

2.2 INFORMATION: WHAT INFORMATION DOES THE MODEL NEED?

Beyond how historical information is accessed, we further consider an often-overlooked dimension: among the relevant history already available, what information is actually useful for effective reasoning? This leads to our second question: *What information does the model need?*

Specifically, we consider two aspects of information presentation: *modality*, i.e., whether the evidence is visual frames, textual descriptions, or their combination, and *temporal structure*, i.e., how temporal relations are preserved or explicitly encoded. Moreover, to isolate the effect of information presentation from the availability of relevant evidence, we first conduct experiments under the *Golden Evidence* setting, where the ground-truth relevant history is provided, and then validate the observed trends under matched and full retrieval settings.

To quantify their contributions, we conduct controlled ablations over different forms of historical evidence, isolating the effects of visual frames, captions, and temporal cues. As shown in Fig. 2 and Tab. 2, visual frames provide the strongest positive contribution, whereas captions have a negative effect. Both implicit temporal ordering and explicit temporal cues are beneficial, but their effects are complementary rather than simply additive: each becomes more valuable when the other is absent. These results indicate that useful historical evidence should retain visual details and make temporal grounding explicit. Taken together, these findings lead to our second design principle: **Memory should provide visual details and explicit temporal grounding.**

2.3 INVOCATION: WHEN IS ADDITIONAL MEMORY NEEDED?

Finally, we investigate whether a streaming model always benefits from memory. We consider three sources of historical context: Recent Context (RC, a baseline using only the recent 8 frames), Sparse Memory (SM, Sparse Buffer with Summary), and Retrieved Memory (RM, retrieved long-term historical evidence). These sources naturally form a hierarchy of historical access, with progressively increasing costs: RC is already available as part of the local context, SM introduces additional memory tokens, and RM further incurs retrieval overhead and adds retrieved tokens to the model input. Despite this progressively broader access, our results in Tab. 3 show that persistently incorporating more memory does not consistently improve performance. For many queries, recent context alone is already sufficient, while additional history may introduce redundant or irrelevant evidence.

However, the limited benefit of persistent memory does not imply that additional memory is generally unnecessary. The oracle analysis in Tab. 3 instead shows substantial gains from providing memory selectively, suggesting that memory demand is highly query-dependent: some queries can be resolved from recent context alone, while others benefit from progressively broader historical access. This shifts the problem from whether to use memory to when the context should be expanded and how far. Memory access should therefore be selective and progressive, expanding from low-cost recent context to increasingly expensive historical access only as needed.

Table 3: Accuracy (%) and mean QA latency of different memory configurations and oracle combinations. †: estimated latency weighted by oracle selection. All Three refers to RC, RC + SM, and RC + SM + RM.

Configuration	Backward Acc. (%)	Realtime Acc. (%)	Overall Acc. (%)	Latency ↓ (ms/query)
RC	50.24	79.69	67.03	179.2
RC + SM	52.93	77.06	66.69	274.2
RC + RM	50.08	74.91	64.24	364.0
RC + SM + RM	52.14	74.79	65.05	478.7
Oracle: RC / (RC + SM)	59.75	82.56	72.75	184.6†
Oracle: All Three	**65.13**	**84.35**	**76.09**	194.6†

Given this query-dependent need for history, a natural approach is self-assessment: asking the VLM to judge whether its current evidence is sufficient. However, explicit self-assessment proves unreliable. As shown in Tab. 4, when directly asked whether the available information is sufficient, the model exhibits strong overconfidence and systematically overestimates evidence sufficiency. Verbalized confidence is therefore an unreliable signal for additional memory. Importantly, this failure does not imply that the model lacks internal signals related to evidence sufficiency. We further perform five-fold MLP probing over intermediate representations. As shown in Tab. 4, even when the probe only observes the model states associated with the current query and Recent Context, it can still partially distinguish whether the available evidence is sufficient for answering the query. This suggests that the need for additional memory is implicitly encoded in the model's internal representations, motivating us to seek a lightweight, inference-time signal for evidence sufficiency.

Table 4: Evidence-sufficiency estimation. Suff. rate: percentage predicted sufficient; QA Acc.: QA accuracy; Suff. B. Acc.: balanced accuracy for classifying evidence as sufficient or insufficient.

Evidence	Method	Suff. rate(%)	QA Acc.(%) ↑	Suff. B. Acc.(%) ↑
RC	Always sufficient	100.00	67.03	50.00
	self-assessment	98.02	66.69	50.38
	MLP probe	67.51	**77.18**	**74.00**
RC + SM	Always sufficient	100.00	66.69	50.00
	self-assessment	98.84	66.49	50.21
	MLP probe	66.83	**74.39**	**71.12**
RC + SM + RM	Always sufficient	100.00	65.05	50.00
	self-assessment	96.39	64.71	50.82
	MLP probe	65.12	**73.77**	**71.14**

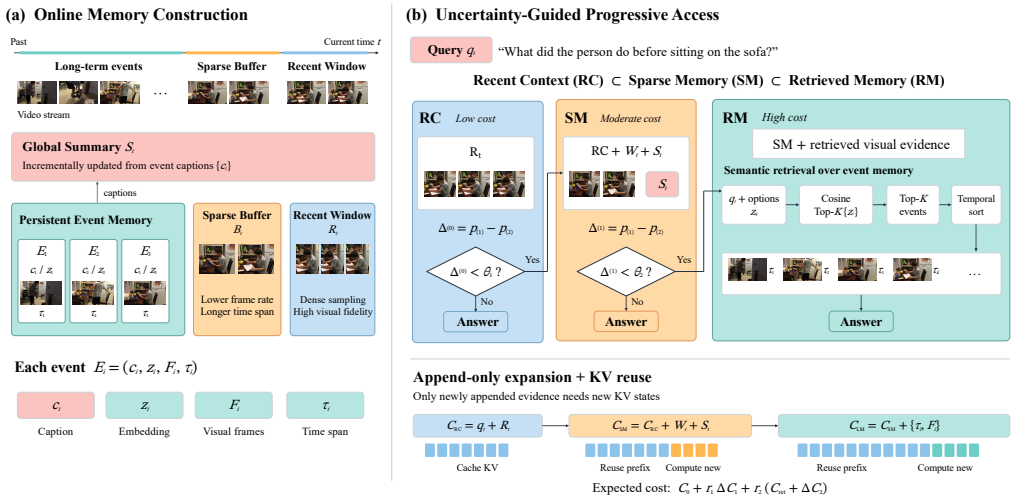


Figure 3: TRACE: Triggered Retrieval and Adaptive Context Expansion

These observations lead to our third design principle: **Memory should be accessed progressively and only when internal model signals indicate that the current evidence is insufficient.**

3 METHOD

As analyzed in the previous section, the key to effective streaming memory is not to continuously provide the model with more history, but to maintain historical information efficiently and determine when and which evidence should be exposed to the model. Guided by the three principles derived above, in this section, we propose TRACE (**T**riggered **R**etrieval and **A**daptive **C**ontext **E**xpansion).

3.1 FRAMEWORK OVERVIEW

As shown in Fig. 3, the proposed TRACE continuously maintains a structured memory state as the video stream evolves and progressively expands access to historical evidence when a query arrives. At time t, TRACE continuously maintain a memory state $\mathcal{M}_t$ that evolves with the incoming video stream, as detailed in Sec. 3.2. To selectively access the maintained memory state $\mathcal{M}_t$ at query time, TRACE defines three context levels with progressively broader historical coverage: **Recent Context (RC)**, which contains only recent observations; **Sparse Memory (SM)**, which augments RC with sparsely short-term history and a compact summary; and **Retrieved Memory (RM)**, which further incorporates evidence retrieved from long-term memory, as detailed in Sec. 3.3. When a query arrives, VLM starts from RC and progressively expands to SM and RM only when the current context is insufficient, using predictive uncertainty as a proxy for evidence sufficiency. Together, these three levels enable TRACE to access broader historical evidence only when needed.

3.2 Online Memory Construction

Let $\mathcal{V}_{1:t}$ denote the video stream observed up to time t. Guided by the first two principles, as new visual observations arrive, TRACE continuously updates memory state $\mathcal{M}_t$ in an query–independent online context, maintaining bounded sparse short-term history for efficient access while preserving fine-grained visual and temporal evidence for later retrieval. Specifically, $\mathcal{M}_t$ contains four components: a densely sampled recent window $\mathcal{R}_t$, a sparsely sampled buffer $\mathcal{B}_t$, a persistent event memory $\mathcal{E}_i$, and an incremental global summary S_t.

Recent Window and **Sparse Buffer.** TRACE maintains two visual streams with distinct roles. The recent window $\mathcal{R}_t$ retains densely sampled observations from a short temporal horizon and serves as the default visual context when a query arrives. In contrast, the sparse buffer $\mathcal{B}_t$ uses a lower sampling rate to cover a longer short-term horizon. Under a bounded visual budget, $\mathcal{R}_t$ prioritizes local visual detail, while $\mathcal{B}_t$ extends temporal coverage for subsequent memory access.

Persistent event memory. For history beyond the Sparse Buffer, TRACE groups every n consecutive frames into an event. Each event is formulated as

$$\mathcal{E}_i = (c_i, z_i, \mathcal{F}_i, \tau_i), \qquad \text{where } i = 1, \ldots, N_t,$$

where i indexes completed events in chronological order up to time t, and N_t denotes the number of completed events. Here, c_i is an event-level caption generated by the VLM, $z_i = f_{\text{emb}}(c_i)$ is its retrieval embedding produced by the embedding model, $\mathcal{F}_i$ denotes the retained visual frames, and τ_i specifies the temporal span of the event. The caption and its embedding provide a compact index over historical events, while the associated visual frames and temporal information are preserved as evidence for downstream reasoning. This design avoids irreversibly discarding fine-grained details through purely semantic compression.

Incremental global summary. In addition, TRACE maintains a rolling global summary S_t over completed events. After every K new events are finalized, TRACE updates the global summary by merging their captions with the previous summary. The update is defined as

$$S_t = f_{\text{sum}}\Big(S_{t^-}, \{c_i\}_{i=N_t-K+1}^{N_t}\Big),$$

where S_{t^-} denotes the latest summary before the current update and f_{sum} is the VLM-based summarization function. In this way, S_t maintains a compact semantic overview of the accumulated video history without revisiting the full stream, complementing the fine-grained visual and temporal evidence retained in event memory with coarse global context.

Overall, TRACE constructs $\mathcal{M}_t = \Big\{\mathcal{R}_t,\ \mathcal{B}_t,\ \{\mathcal{E}_i\}_{i=1}^{N_t},\ S_t\Big\}$ by jointly maintaining sparse short-term visual history, an incrementally updated global summary, and persistent event memory. These representations capture complementary temporal scales and information granularity, providing a compact yet structured memory state that can be continuously updated as the video stream evolves.

3.3 Uncertainty-Guided Progressive Memory Access

Hierarchical Context Levels. Following the principles established in Sec. 2, TRACE constructs three nested query-time contexts with progressively broader historical coverage. Given a query q_t, $\mathcal{X}_t^{(0)}$ contains only the most recent visual observations, i.e., Recent Window $\mathcal{R}_t$. $\mathcal{X}_t^{(1)}$ extends $\mathcal{X}_t^{(0)}$ with a longer, sparsely sampled visual window and a global summary, providing complementary short-term visual detail and broader temporal coverage within a bounded context, i.e., the combination of Sparse Buffer $\mathcal{B}_t$ and Summary S_t. $\mathcal{X}_t^{(2)}$ further incorporates query-relevant long-term events $\{\mathcal{E}_i\}_{i\in\mathcal{I}_t}$, where $\mathcal{I}_t$ indexes the retrieved by matching the query embedding and event embeddings. For each retrieved event, TRACE exposes its retrieved vistual frames $\mathcal{F}_i$ together with the corresponding temporal span τ_i. Formally, we construct the nested hierarchy memory with increasing historical coverage and access cost as

$$\mathcal{X}_t^{(1)} = \mathcal{X}_t^{(0)} \oplus \mathcal{B}_t \oplus \mathcal{S}_t, \qquad \mathcal{X}_t^{(2)} = \mathcal{X}_t^{(1)} \oplus \{\mathcal{E}_i\}_{i\in\mathcal{I}_t}. \tag{1}$$

We refer to $\mathcal{X}_t^{(0)}$, $\mathcal{X}_t^{(1)}$, and $\mathcal{X}_t^{(2)}$ as Recent Context (RC), Sparse Memory (SM), and Retrieved Memory (RM), respectively.

Uncertainty-Guided Context Expansion. Following our third principle, TRACE does not expose the full memory at once. Instead, it progressively expands the accessible history with hierarchical context levels according to whether the current prediction is sufficiently confident. Given a query q_t, TRACE starts from the lowest-cost RC and expands to SM and RM only when the current evidence is insufficient.

Specifically, we use the margin between the two most probable tokens as an internal confidence signal, which is formulated as

$$\Delta^{(\ell)} = p^{(\ell)}_{(1)} - p^{(\ell)}_{(2)}, \qquad \ell \in \{\mathrm{RC}, \mathrm{SM}\}, \tag{2}$$

where $p^{(\ell)}_{(1)}$ and $p^{(\ell)}_{(2)}$ denote the largest and second-largest probabilities under context level ℓ, respectively. A smaller $\Delta^{(\ell)}$ indicates a less decisive prediction under the currently available evidence and therefore suggests that additional context may be needed (Schuster et al., 2022). TRACE first prefills RC with query and computes $\Delta^{(\mathrm{RC})}$ based on the prediction of VLM. If $\Delta^{(\mathrm{RC})} \geq \theta_1$, the model answers directly; otherwise, the context is expanded to SM. After prefilling the expanded context, TRACE recomputes the confidence as $\Delta^{(\mathrm{SM})}$. If $\Delta^{(\mathrm{SM})} \geq \theta_2$, the model answers from SM; otherwise, long-term retrieval is triggered by comparing the query embedding with the event-caption embeddings $\{z_i\}$ and using the top-K most relevant events. In this way, TRACE progressively introduces additional historical evidence only when the current context is insufficient, while invoking the more costly long-term retrieval only after both RC and SM fail to provide a sufficiently confident prediction.

3.4 Efficient Incremental Inference

The progressive memory access described above not only avoids unnecessary historical access, but also induces naturally nested contexts. TRACE therefore implements context escalation as incremental evidence expansion rather than independent full inference at each memory level.

Incremental context expansion. During expansion, the nested structure of the proposed memory allows TRACE to reuse the KV cache of the existing context and compute new KV states only for the appended evidence, avoiding redundant encoding of shared context across memory levels.

Expected query-time computation. This design benefits from both conditional memory access and incremental computation. Let C_0 denote the cost of the initial RC inference, ΔC_1 and ΔC_2 the incremental costs of expanding to SM and RM, and C_{ret} the cost of long-term retrieval. If r_1 and r_2 denote the fractions of queries reaching SM and RM, respectively, with $0 \leq r_2 \leq r_1 \leq 1$, the expected query-time cost is

$$\mathbb{E}[C_{\mathrm{TRACE}}] = C_0 + r_1 \Delta C_1 + r_2 \left(C_{\mathrm{ret}} + \Delta C_2\right). \tag{3}$$

Thus, most queries incur only the low-cost RC inference, while additional memory-access costs are paid only by queries for which the current evidence is insufficient. KV reuse further reduces the cost of escalation to the incremental computation required by newly introduced evidence.

4 Experiment

4.1 Experiment Setup

Benchmarks and Streaming Protocol. We primarily evaluate TRACE on OVO-Bench (Niu et al., 2025), which contains 1,640 questions across 12 tasks. Focusing on online perception and historical memory access, we use the *Backward Tracing* and *Real-Time Visual Perception* subsets. Following the official streaming protocol, each query can access only video content observed before its timestamp, and performance is reported using the official metrics. We additionally evaluate cross-dataset generalization on StreamingBench under its *full-context* setting.

Backbones and Baselines. We use Qwen3.5-9B as the default backbone and further evaluate Qwen3-VL-8B to assess generalization across Video-LLMs. We compare TRACE with representative streaming video understanding methods: SimpleStream, which relies only on recent

Table 5: Results on the Backward and Realtime subsets of OVO-Bench. Best and second-best results are shown in **bold** and underlined, respectively.

Method	Input Frames	Backward ↑				Realtime ↑							Overall ↑ (micro)	Efficiency‡ ↓	
		EPM	ASI	HLD	Avg. (micro)	OCR	ACR	ATR	STU	FPD	OJR	Avg. (micro)		FLOPs (T/query)	Latency (s/query)
Qwen3-VL-8B (vanilla)	0.5(per sec)	53.20	67.57	21.51	47.23	81.88	66.06	78.45	56.74	70.30	66.85	69.30	59.81	538.82	16.48
Qwen3.5-9B (vanilla)	0.5(per sec)	59.60	**68.92**	26.88	52.14	85.23	65.14	82.76	55.06	**80.20**	70.65	72.04	63.49	325.64	19.10
SimpleStream (Qwen3.5-9B) (Shen et al., 2026)	4	57.91	58.78	44.09	54.04	95.30	77.98	**87.93**	73.03	76.24	**87.50**	**83.27**	70.71	38.94	0.59
SimpleStream (Qwen3.5-9B) (Shen et al., 2026)	16	65.32	64.86	38.17	57.21	93.96	77.06	81.03	69.66	79.21	79.35	79.81	70.10	155.18	1.79
SAVEMem (Qwen3.5-9B) (Wu et al., 2026)	1(per sec)	56.90	63.51	16.67	46.59	94.63	80.73	84.48	72.47	78.22	78.80	81.24	66.35	190.59	1.99
OASIS (Qwen3-VL-8B) (Liang et al., 2026)	8–48	53.20	64.86	50.00	54.99	91.95	80.73	81.90	66.29	66.34	78.26	77.54	67.85	29.35	6.13
OASIS (Qwen3.5-9B) (Liang et al., 2026)	8–48	60.94	65.54	44.62	57.21	93.29	77.98	80.17	67.42	70.30	75.54	77.30	68.66	191.11	13.84
WeaveTime (LLaVA-OV-7B) (Zhang et al., 2026b)	1 (per sec)	60.61	60.81	10.75	45.96	78.52	66.97	74.14	48.88	73.27	73.37	68.34	58.72	3.24	0.43
TRACE (Qwen3-VL-8B)	8–32	56.57	66.89	**45.16**	55.63	95.97	**87.16**	81.90	**73.60**	73.27	82.61	82.44	70.91	92.19	1.31
TRACE (Qwen3.5-9B)	8–32	**67.34**	64.19	41.40	**58.95**	**96.64**	81.65	83.62	**73.60**	77.23	84.24	82.92	**72.62**	116.39	1.86

‡ FLOPs and latency measure per-query inference only, excluding the cost of Memory/cache construction and online updates. The corresponding memory-maintenance overhead is evaluated separately in the Appendix.

visual context; OASIS, which constructs an Event Tree and retrieves historical events via tool-based access; SAVEMEM, which adopts hierarchical memory organization and adaptive retrieval; and WEAVETIME, which combines temporally aware training with uncertainty-triggered historical KV retrieval. Together, these baselines cover representative paradigms including recent-context inference, structured persistent memory, adaptive retrieval, and on-demand historical access.

Implementation Details. We use the same VLM for online memory construction and downstream question answering. The video stream is sampled online at 1 FPS, with every 16 frames grouped into an event chunk containing a textual description and eight representative frames. The global Summary is updated every four chunks. Recent Context (RC) contains the latest 8 frames sampled at 2 FPS, while Sparse Memory (SM) additionally includes 8 frames sampled at 1 FPS. Retrieved Memory (RM) retrieves the top-2 event chunks, whose textual representations are indexed using Qwen-Embedding-0.6B. TRACE uses the top-1 - top-2 probability margin as the routing signal for progressive memory access, with the expansion threshold fixed at 0.3.

4.2 MAIN RESULTS

Results on OVO-Bench. On real-time perception, TRACE improves Qwen3.5-9B from 72.04% to 82.92% (+10.88 points). These tasks require accurate perception of the current scene despite an ever-growing history, suggesting that selectively exposing historical evidence helps preserve focus on recent observations while avoiding interference from irrelevant context. On backward tracing, TRACE improves accuracy from 52.14% to 58.95% (+6.81 points), showing that progressive context expansion can recover earlier evidence when recent context alone is insufficient. Together, these results show that TRACE improves historical recall without sacrificing real-time perception, maintaining strong performance across both regimes through query-adaptive memory access.

Overall accuracy increases from 63.49% to 72.62% (+9.13 points), while per-query inference latency decreases from 19.10,s to 2.31,s, corresponding to an 87.9% reduction. Notably, this efficiency is achieved while improving both real-time and backward reasoning, rather than optimizing one at the expense of the other. By avoiding persistent historical context and expanding memory only when needed, TRACE achieves a favorable accuracy–latency trade-off for streaming inference.

Results on StreamingBench. We further evaluate TRACE on StreamingBench to examine its generalization beyond OVO-Bench. As shown in Tab. 6, TRACE consistently improves performance across all four task categories. The gain is particularly pronounced on MCU, where highly similar consecutive video segments can make it difficult for the model to identify the evidence relevant to the

Table 6: Results on StreamingBench.

Model	Frames	Real-Time All	ACU	MCU	SQA
Qwen3.5-9B	0.5fps	83.84	53.20	46.40	40.40
SimpleStream	16	**86.16**	64.40	56.80	54.80
SAVEMem	256	82.88	63.20	62.00	44.40
OASIS	8–48	82.64	66.80	55.20	42.00
TRACE	8–32	85.60	**71.20**	**62.40**	**61.20**

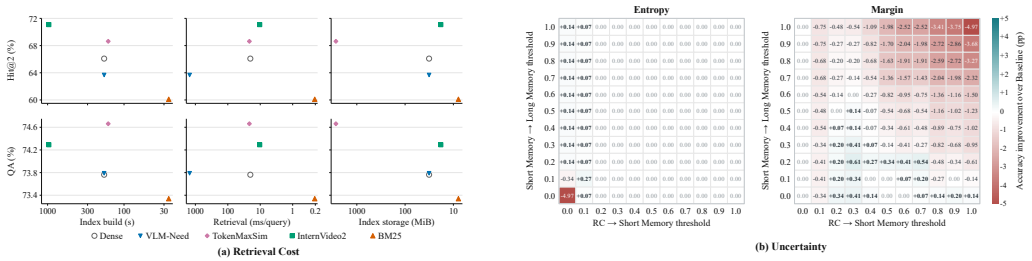


Figure 4: **Retrieval and uncertainty ablations.**

current query. By progressively exposing additional history only when the current prediction is uncertain, TRACE reduces unnecessary historical interference while retaining access to complementary evidence when needed. Improvements on SQA further suggest that this strategy remains effective under more complex long-context reasoning scenarios.

4.3 Ablation Study

Generalization across Backbones. To verify that the effectiveness of TRACE is not specific to the Qwen model family, we further evaluate it with GLM as an additional backbone. As reported in Tab. 7, TRACE consistently improves performance on the non-Qwen backbone, suggesting that progressive memory access is not tied to a particular model architecture.

Table 7: Generalization in other backbone.

Backbone	Method	Backward	Realtime	Overall
GLM-4.6V-Flash	Vanilla	52.14	68.22	61.31
	TRACE	60.86	77.06	70.10

Effect of Retrieval Paradigm. We compare five representative retrieval paradigms. As shown in Fig. 4(a), more sophisticated methods can improve retrieval on specific tasks, but introduce additional overhead without consistent gains in downstream QA. Moreover, retrieval quality alone does not determine evidence utility, which also depends on how the retrieved content is represented and organized. These results motivate TRACE to balance retrieval effectiveness with online efficiency.

Table 8: Ablation of progressive memory access.

Access Strategy	Backward	Realtime	Overall
RC only	54.83	**82.92**	70.84
RC → SM	56.89	82.44	71.46
SM → RM	58.00	78.85	69.89
RC → RM	54.20	82.44	70.30
RC → {SM, RM}	55.31	81.48	70.23
RC → SM → RM	**58.95**	**82.92**	**72.62**

Uncertainty Metric and Threshold Selection. We next study the uncertainty signal for triggering context expansion. While entropy is widely used, its long-tailed distribution can limit discrimination near the decision boundary. In contrast, the top-1–top-2 margin, $\Delta = p_{(1)} - p_{(2)}$, directly captures ambiguity between the two most competitive predictions. As shown in Fig. 4(b), the margin provides clearer separation between queries that are sufficiently supported by the current context and those requiring additional memory. TRACE also remains robust across a reasonable range of routing thresholds, suggesting that its gains are not sensitive to a narrowly tuned operating point.

Effect of Progressive Memory Access. We examine the contribution of the multi-stage memory access strategy by comparing TRACE with its single-stage variants. As shown in Tab. 8, both Sparse Memory and Retrieved Memory contribute to the final performance. Their gains arise from being selectively activated for different queries, supporting the progressive design of TRACE.

5 Conclusion

In this work, we revisited memory design for streaming video understanding by disentangling three key dimensions—**access**, **information**, and **invocation**—and distilled our analyses into practical design principles. Guided by these insights, we proposed TRACE, a training-free progressive memory framework that organizes Recent, Short, and Long memory and expands historical context according to predictive uncertainty. By relying on low-cost recent context by default and accessing additional memory only when needed, TRACE reduces unnecessary computation and historical interference while improving real-time perception, historical reasoning, and overall efficiency across VLMs.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and debugging experimental code, as well as polishing the manuscript for clarity, grammar, and readability. All AI-assisted code was reviewed, verified, and tested by the authors, and all AI-assisted text was carefully examined and revised before inclusion in the manuscript. We did not use generative AI tools for research ideation, experimental design, data analysis, result interpretation, or the generation of scientific claims. We take responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies memory construction and adaptive historical-context access for streaming video understanding. Our experiments rely on existing research benchmarks and pretrained models, and we do not collect new human-subject data or introduce a new dataset containing personally identifiable information. The proposed method is intended for general video understanding research and does not target sensitive decision-making applications. As with general-purpose video understanding systems, potential concerns related to privacy, bias in pretrained models and benchmark data, and inappropriate deployment may arise when applying such systems in real-world settings. We encourage responsible use consistent with the licenses, intended uses, and ethical guidelines associated with the underlying models and datasets.

REPRODUCIBILITY STATEMENT

We provide the information necessary to reproduce our experiments throughout the main paper and appendix. The method section specifies the memory construction, evidence representation, retrieval procedure, and uncertainty-guided progressive access mechanism used in TRACE. The experimental section describes the evaluated benchmarks, backbone models, baselines, streaming protocols, and evaluation metrics. The appendix further provides detailed analysis settings, prompts, implementation configurations, hyperparameters, and additional ablation results. Together, these materials document the experimental protocol and design choices required to reproduce the reported results.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023. URL https://arxiv.org/abs/2310.11511.

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

Dibyadip Chatterjee, Edoardo Remelli, Yale Song, Bugra Tekin, Abhay Mittal, Bharat Bhatnagar, Necati Cihan Camgoz, Shreyas Hampali, Eric Sauser, Shugao Ma, et al. Streaming videollms for real-time procedural video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22586–22598, 2025.

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

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

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

Lars Doorenbos, Federico Spurio, and Juergen Gall. Video panels for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31293–31303, 2026.

Haonan Ge, Yiwei Wang, Hang Wu, and Yujun Cai. What should a streaming video model remember?, 2026. URL https://arxiv.org/abs/2606.16353.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13504–13514, June 2024.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. Oasis: On-demand hierarchical event memory for streaming video reasoning. *arXiv preprint arXiv:2604.17052*, 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 *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151. IEEE, 2026.

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

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 119336–119360. Curran Associates, Inc., 2024. doi: 10.52202/079017-3792. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d7ce06e9293c3d8e6cb3f80b4157f875-Paper-Conference.pdf.

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

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

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323, June 2024.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

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

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18221–18232. IEEE, 2024.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXX*, pp. 58–76, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72988-1. doi: 10.1007/978-3-031-72989-8_4. URL https://doi.org/10.1007/978-3-031-72989-8_4.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025a.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2025b.

Hang Wu, Sherin Mary Mathews, Yujun Cai, Ming-Hsuan Yang, and Yiwei Wang. Semantic-aware adaptive visual memory for streaming video understanding. *arXiv preprint arXiv:2605.07897*, 2026.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13204–13214. IEEE, 2024.

Junlin Xie, Quanlong Zheng, Ruifei Zhang, Kuo Wang, Yanhao Zhang, Jinguo Luo, Haonan Lu, Xiang Wan, and Guanbin Li. Streamrag: Enhancing real-time video understanding with retrieval augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38870–38879, 2026.

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

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering, 2024. URL https://arxiv.org/abs/2312.17235.

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

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

Yulin Zhang, Cheng Shi, and Sibei Yang. Weavetime: Stream from earlier frames into emergent memory in videollms. *arXiv preprint arXiv:2602.22142*, 2026b.

A Related Work

Streaming Video Understanding and Memory. Recent video large language models have extended from offline understanding to continuous streaming scenarios with causal observations and growing history. VideoLLM-online (Chen et al., 2024) formulates temporally aligned online video interaction, while VideoStreaming (Qian et al., 2024) introduces memory-propagated streaming encoding and adaptive memory selection for arbitrarily long streams. Subsequent works develop explicit memory mechanisms for historical information. ReKV (Di et al., 2025) retrieves query-relevant historical KV caches; Flash-VStream (Zhang et al., 2025) combines compact contextual and augmentation memories; StreamForest (Zeng et al., 2026) organizes history into persistent hierarchical event memory; and ProVideLLM (Chatterjee et al., 2025) combines short-term visual context with compressed long-term textual memory. Recent approaches further improve efficiency through hierarchical KV organization (Zhang et al., 2026a) or online token compression and retrieval (Chen et al., 2026). These methods extend temporal coverage but typically couple representation, compression, retrieval, and invocation within a unified memory design. In contrast, SimpleStream (Shen et al., 2026) shows that a simple recent sliding window can remain competitive with substantially more complex memory systems, suggesting that the key factors behind effective streaming memory remain insufficiently understood. Motivated by this observation, we disentangle streaming memory into three questions—how history should be accessed, what information should be exposed, and when additional memory should be invoked—and derive our design from these analyses.

Historical Evidence Representation and Access. Long-video understanding has explored various ways to represent extended visual history under limited context budgets. MovieChat (Song et al., 2024) compresses short-term visual tokens into sparse long-term memory, while MA-LMM (He et al., 2024) incrementally maintains and merges a visual memory bank. LongVU (Shen et al., 2024) performs spatiotemporal adaptive compression, and Video Panels (Doorenbos et al., 2026) trades spatial detail for broader temporal coverage by composing multiple frames into compact panels, highlighting the inherent tension between visual fidelity and temporal coverage.

Other approaches rely on semantic compression or selective access. LLoVi (Zhang et al., 2024) represents long videos using clip-level descriptions and hierarchical summarization; VideoTree (Wang et al., 2025b) constructs query-adaptive hierarchical representations for coarse-to-fine evidence selection; and VideoAgent (Wang et al., 2024) iteratively searches for informative frames instead of loading the full video. TimeChat (Ren et al., 2024) further incorporates timestamp-aware representations, demonstrating the importance of temporal localization. These methods address different aspects of the coverage–detail trade-off, but often treat memory representation and reasoning evidence jointly. We instead separate the representation used to *locate* history from the evidence exposed for *reasoning*: compact captions and embeddings support semantic indexing, while retrieved visual observations and temporal information preserve fine-grained reasoning evidence. This distinction motivates TRACE to combine bounded sparse context and global semantic summary with selectively retrieved visual evidence, rather than relying solely on dense visual history or irreversible textual compression.

Adaptive Retrieval and Memory Invocation. Our work is also related to adaptive computation and retrieval, where additional resources are introduced only when current evidence is insufficient. In language modeling, CALM (Schuster et al., 2022) dynamically allocates computation according to model confidence. For retrieval-augmented generation, FLARE (Jiang et al., 2023) triggers retrieval when generation confidence is low, Self-RAG (Asai et al., 2023) learns reflection tokens to decide when retrieval and critique are needed, and Adaptive-RAG (Jeong et al., 2024) routes queries among retrieval strategies according to estimated complexity. These methods establish the value of conditional retrieval, but primarily operate over textual knowledge rather than continuously growing, temporally structured visual history.

Adaptive memory access has recently emerged in streaming video as well. Most closely related, OASIS (Liang et al., 2026) first reasons from short context and retrieves from hierarchical event memory when the current prediction is uncertain. SAVEMem (Wu et al., 2026) adaptively selects visual memory across temporal scales, while SelectStream (Ge et al., 2026) studies selective memory writing and retrieval under query-conditioned evidence budgets. These works reinforce the importance of conditional memory access. Our contribution is therefore not simply to retrieve mem-

Table 9: Evidence presentation accuracy (%). †Golden denotes evaluation with oracle evidence.

Modality		Temporal structure		Accuracy (%)		
Frames	Captions	Implicit time	Explicit time	Golden† 100	Matched 100	Full 445
–	–	–	–	42.00	44.00	42.47
–	–	–	✓	36.00	40.00	40.22
✓	–	✓	✓	72.00	48.00	53.93
✓	–	✓	–	70.00	44.00	53.48
✓	–	–	✓	72.00	52.00	54.38
✓	–	–	–	68.00	42.00	52.58
–	✓	–	✓	66.00	42.00	44.49
–	✓	–	–	66.00	40.00	44.04
✓	✓	✓	✓	70.00	50.00	51.91
✓	✓	✓	–	70.00	44.00	52.13
✓	✓	–	✓	76.00	50.00	52.13
✓	✓	–	–	66.00	48.00	51.24

ory on demand, but to systematically study *access*, *information*, and *invocation* as separate design dimensions and instantiate the resulting principles in a training-free progressive hierarchy. TRACE starts from Recent Context (RC), expands to bounded Sparse Memory (SM), and invokes Retrieved Memory (RM) only when predictive uncertainty remains high. Because these contexts are nested, TRACE can reuse previously computed KV states and append only newly introduced evidence, turning memory invocation from a binary retrieval decision into progressive evidence expansion while preserving visual detail and explicit temporal grounding.

B Additional Experiment Results

B.1 Evidence Presentation Results

To further examine how different forms of historical evidence affect downstream reasoning, we evaluate combinations of two modality factors, *Frames* and *Captions*, and two temporal factors, *Implicit time* and *Explicit time*. Implicit time preserves the chronological ordering of the evidence, whereas explicit time additionally provides explicit temporal cues associated with the presented evidence. We report results under three evaluation settings: *Golden*, *Matched*, and *Full*. Golden and Matched each contain 50 samples, while Full contains all 445 samples used in this analysis.

Effect of evidence modality. Table 9 shows that visual frames are the most consistently useful form of historical evidence. On the Full set, introducing frames improves accuracy from 42.47% to 52.58% even without additional temporal cues. The same trend is more pronounced under Golden evidence, where frame-based configurations reach substantially higher accuracy than the corresponding evidence-free setting. In contrast, captions alone provide only limited improvement: their Full-set accuracy increases from 42.47% to 44.04%. Moreover, combining captions with frames does not consistently outperform using frames alone. For example, with both temporal cues enabled, Frames achieves 53.93% on Full, whereas Frames+Captions obtains 51.91%. This suggests that semantic compression through captions may discard fine-grained visual details or introduce information that interferes with evidence already preserved by visual frames.

This observation is also reflected in the factor-wise analysis reported in the main paper: Frames exhibit the strongest positive average effect ($+1.12$), whereas Captions have a negative average

Table 10: Estimated online memory construction cost. Latency is normalized by total video duration and excludes downstream QA inference.

Method	Build Latency (s / min video) ↓	Peak Device Memory (GiB) ↓	Storage (MiB / min) ↓
OASIS	116.8	≤ 20.0	0.075
WeaveTime	14.1	≤ 22.6	3104
TRACE	27.8	≤ 20	0.017

effect (-0.64). These results support using captions primarily as compact semantic indices for memory access, while retaining the associated visual observations as the evidence exposed to the QA model.

Effect of temporal structure. Temporal information provides an additional but context-dependent benefit. Implicit ordering and explicit temporal cues are complementary: when one form of temporal information is absent, introducing the other can improve reasoning, whereas their gains become smaller when both are already available. Across all configurations, their average effects are positive, with Explicit time ($+0.49$) contributing more than Implicit time ($+0.11$).

The detailed results illustrate this interaction. With frames but without captions, adding explicit temporal cues improves Full accuracy from 52.58% to 54.38%, while adding implicit ordering gives 53.48%. When both are present, the result is 53.93%, indicating that their benefits are not simply additive. A similar pattern can be observed when captions are combined with frames. These results suggest that temporal structure is useful for grounding historical evidence, but part of the information conveyed by implicit ordering and explicit timestamps is overlapping. Explicit cues generally provide the stronger signal because they directly expose the temporal location of the evidence rather than requiring the model to infer it from presentation order.

Evidence quality and presentation. The advantage of appropriate evidence presentation is strongly conditioned on whether the accessed history actually contains the answer-relevant information. Frame-based configurations perform substantially better under Golden evidence than under Matched or Full evaluation. For instance, Frames with both temporal cues reaches 72.00% on Golden but only 48.00% on Matched and 53.93% on Full. Likewise, the best Golden configuration reaches 76.00%, whereas corresponding results with retrieved evidence remain considerably lower.

This gap indicates that evidence presentation and evidence retrieval address two distinct bottlenecks. Preserving visual detail and temporal grounding can make relevant evidence easier for the VLM to exploit, but presentation alone cannot recover information that was missed during retrieval. Therefore, effective streaming memory requires both accurate historical access and an evidence representation that preserves answer-critical visual details and explicit temporal grounding.

B.2 Memory Construction Cost

Beyond answer quality, a streaming memory system must continuously maintain its historical representation as new video observations arrive. We therefore report the online memory construction cost of TRACE and compare it with OASIS and WeaveTime in Tab. 10. Rather than maintaining a complex hierarchical or repeatedly reorganized memory structure, TRACE adopts a lightweight construction pipeline based on periodic chunk-level representations and compact global summaries. This simpler design substantially reduces recurring memory-maintenance latency, making historical access more compatible with real-time streaming inference.

C Analysis Setup

This section specifies the experimental protocols used for the analyses of *access*, *information*, and *invocation*. Unless otherwise stated, these analyses are diagnostic experiments in which each context condition is evaluated independently without uncertainty-based routing. We distinguish these fixed-context analyses from the adaptive inference procedure used by TRACE.

C.1 Common Evaluation Protocol

Data and backbone. Our streaming analyses use the official Real-Time Visual Perception and Backward Tracing subsets of OVO-Bench, containing 837 and 631 questions, respectively, for a total of 1,468 questions. The backward subset consists of EPM (297), ASI (148), and HLD (186). Experiments reported only on EPM and ASI therefore contain 445 questions and should not be confused with the complete backward split. Unless otherwise specified, Qwen3.5-9B is used as the QA backbone with BF16 inference, SDPA attention, disabled thinking, and random seed 0. Dense retrieval uses Qwen3-Embedding-0.6B.

Causal visual context. For a query arriving at time t, the accessible video endpoint is the minimum of t and the timestamp of the final physical frame available in the video. Recent Context (RC) contains up to eight frames sampled at 2 FPS. The preceding sparse window contains up to eight frames sampled at 1 FPS and excludes source frames already included in RC. Requested timestamps are always mapped to causal source frames, and early queries use only actually available frames rather than padding the stream with synthetic observations. Visual preprocessing uses a configured upper bound of 256 visual tokens per frame; this is a preprocessing limit rather than an equal-token-budget assumption across experimental conditions.

Memory construction. Historical video is sampled at 1 FPS and partitioned into non-overlapping complete chunks of 16 frames. All frames in a chunk are provided to the memory VLM to generate a question-independent event caption with at most 128 output tokens. After every four completed chunks, their captions are merged with the previous cumulative summary, with a maximum generation length of 512 tokens. Both caption and summary generation use greedy decoding and have no access to the question, answer options, or target answer.

The original summary protocol updates the cumulative summary after each complete group of four chunks. Before the first full group, a single completed chunk directly uses its caption, while two or three completed chunks are summarized jointly. For several later analyses, we additionally use *Summary*, which incorporates the final group of one to three completed chunks into the preceding summary when such a summary exists. Frames belonging to an unfinished chunk are never included. Experiments using frozen original summaries and experiments using Summary retain their respective protocols and are not retroactively substituted for one another.

Dense retrieval. For retrieval, the question and all candidate options are jointly encoded as the query. Caption embeddings and query embeddings are ℓ_2-normalized, and candidate chunks are ranked by cosine similarity, implemented as their dot product. The top two available chunks are selected and subsequently exposed to the QA model in chronological order. Eight frames are uniformly sampled from each retrieved chunk. When fewer than two completed chunks are available, all available chunks are used; queries with no completed historical chunk remain part of the evaluation. In frozen-evidence analyses, retrieval results, captions, frames, and summaries are reused across all compared presentation conditions.

Answer scoring. For OVO-Bench, prediction is determined by the valid answer option whose corresponding token has the largest first-token logit. The independent-image implementation realizes the same rule through one-token greedy generation constrained to the valid option tokens. Ground-truth labels are used only for evaluation, oracle analysis, and probe supervision. Accuracy is computed over individual questions, and Overall accuracy is micro-averaged rather than averaging task-level percentages. All reported factor effects are single-seed descriptive differences and are not intended as statistical significance tests.

C.2 Access Analysis

Fixed access conditions. We use two complementary sets of fixed-context comparisons. For component ablations, the sparse-window condition consists of RC together with the preceding sparse visual window; the summary condition consists of RC and the cumulative summary; and their joint condition contains RC, the sparse window, and the same summary. Thus, the summary component in this analysis is evaluated together with recent visual evidence rather than as text alone.

We additionally evaluate pure memory-access paradigms. In this comparison, *summary-only* provides the cumulative textual summary without RC or sparse frames, whereas *retrieval-only* provides retrieved visual evidence and its temporal metadata without RC, sparse visual history, or the global summary. Captions used as retrieval keys are not necessarily exposed to the QA model.

These distinctions are important when interpreting the abbreviated access conditions in the main paper. Adding or removing a memory component changes the amount and modality of the input, and a common per-frame preprocessing limit does not imply an equal total context budget. Moreover, some historical retrieval experiments provide frames as independent images, whereas later experiments use video blocks. Accordingly, these experiments characterize the specified access configurations rather than an isolated causal effect of a single component.

Oracle complementarity. To quantify complementarity among fixed access strategies, for a prediction pool $\mathcal{P}$ we compute

$$\mathrm{Acc}_{\mathrm{oracle}}(\mathcal{P}) = \frac{1}{N}\sum_{i=1}^{N} \max_{j\in\mathcal{P}} \mathbf{1}\left[\hat{y}_{ij} = y_i\right], \tag{4}$$

where $\hat{y}_{ij}$ denotes the prediction of configuration j for question i and y_i is the ground-truth answer. All configurations are aligned by question identity. This statistic measures the fraction of questions correctly answered by at least one configuration in the specified pool. It uses ground-truth correctness and therefore represents an offline complementarity analysis rather than an executable routing policy or a general upper bound for a jointly conditioned model.

C.3 Information Analysis

Evidence sources. We distinguish *Golden*, *Matched*, and *Full* evidence. Golden evidence directly uses annotated answer-relevant intervals and is therefore an offline diagnostic of evidence utilization. Because some annotated intervals extend beyond the official query timestamp, Golden results should not be interpreted as causal online performance. Matched retrieval evaluates retrieved evidence on the same question IDs as an annotated subset, whereas Full evaluates retrieval over the complete stated question set. Matching question IDs does not otherwise equalize evidence availability, frame counts, caption sources, or presentation protocols.

The current Golden100 experiment contains 100 independently evaluated questions, consisting of 50 EPM and 50 ASI examples. In the RC-plus-sparse-memory setting, all conditions share the same frozen RC, eight sparse frames, and Summary.

Independent-image evidence protocol. Our earlier evidence-only analysis is conducted on the 445 EPM and ASI questions without RC, a sparse window, or a global summary. Retrieved chunks, captions, and selected images are frozen across conditions.

Frames are provided as independent images. When captions and frames are jointly enabled, all captions precede the visual evidence. Explicit temporal information includes chunk intervals, source identifiers, and absolute frame timestamps. Removing explicit time removes these evidence annotations while retaining the query-time text and system instruction. For the frame-order intervention, frames are globally permuted across selected chunks while each frame remains paired with its timestamp and provenance; captions retain their original order.

Fixed-context video-block protocol. The later presentation analysis evaluates the complete 631-question backward split, while factor effects on EPM and ASI are additionally reported over the 445-question subset. RC, the sparse window, and Summary are fixed across all conditions. Retrieved events are presented as separate video blocks.

We independently vary four factors: retrieved visual frames (F), retrieved captions (C), explicit temporal intervals (T), and frame ordering. The explicit temporal factor specifies the start and end timestamps of each retrieved segment, rather than an absolute timestamp for every individual frame. The shuffle intervention deterministically permutes frames within each retrieved segment while preserving segment order and moving frame provenance together with the pixels. The video processor's own local temporal encoding remains unchanged. Consequently, removing explicit intervals does not eliminate every temporal signal available to the model.

Factor effects. The factorial experiment evaluates the 15 nonempty combinations of $\{F, C, T, \text{shuffle}\}$, with the shared-context input serving as the empty condition. When frames are absent, frame shuffling leaves the input unchanged. For factor f, its average effect is computed over applicable pairs of conditions differing only in that factor:

$$\beta_f = \frac{100}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \left[\text{Acc}(c, f \text{ retained}) - \text{Acc}(c, f \text{ removed}) \right]. \tag{5}$$

Accuracy inside the brackets is expressed as a fraction, so β_f is reported in percentage points. The unique-context comparison contains eight frame pairs, six caption pairs, four frame-order pairs, and six explicit-time pairs. Frame order is defined only when frames are present.

C.4 Invocation Analysis

Fixed evidence stages. The sufficiency analysis evaluates all 1,468 queries at three fixed evidence levels:

$$\text{RC}, \qquad \text{RC} + \text{SM}, \qquad \text{RC} + \text{SM} + \text{RM}. \tag{6}$$

Here, the second stage augments RC with the sparse visual window and the original cumulative summary, while the third additionally includes the original top-two retrieved historical chunks with their captions, intervals, and frames. These correspond to the recorded A0, A1, and A2 diagnostic contexts. Every query is independently evaluated at every stage; no routing is applied. The A2 context belongs to the historical diagnostic experiment and exposes retrieval captions, unlike the caption-free retrieved evidence used by the final TRACE configuration.

Operational sufficiency labels. For question i at access level ℓ, we define

$$s_i^{(\ell)} = \mathbf{1}\left[\hat{y}_i^{(\ell)} = y_i\right]. \tag{7}$$

That is, a context is operationally labeled sufficient when the QA model answers correctly under the same forward pass. This is intentionally a model-dependent proxy rather than a human annotation that the supplied evidence logically contains everything required for the answer. A correct prediction may arise from guessing, while an incorrect prediction may occur despite adequate evidence. The experiment therefore evaluates how well stage-wise QA correctness can be predicted, rather than directly supervising whether escalation would improve the answer.

Explicit self-assessment. At each stage, we additionally ask the same VLM to explicitly judge whether the supplied evidence is sufficient. We evaluate both a judge-only prompt and a judge-then-answer prompt. Both use greedy decoding with disabled thinking and a maximum generation length of 96 tokens.

Malformed outputs are retained and rerun using an explicit output contract and constrained JSON decoding. Reported repaired results combine the originally valid outputs with these fallback runs rather than applying the fallback prompt uniformly to every sample.

Latent probe. As an alternative to explicit verbal judgment, we extract the hidden state of the final prompt token during the original QA prefill. Features are taken after decoder block 24 of 32, producing a 4,096-dimensional representation. For each access stage, we train an independent MLP:

$$4096 \to 512 \to 128 \to 2, \tag{8}$$

with ReLU activations between hidden layers. Each probe uses features generated from its corresponding evidence level.

We construct five shared cross-validation folds by grouping the 512 source videos, sorting their IDs, shuffling them using NumPy seed 0, and cyclically assigning videos to folds. All questions from the same source video remain within the same fold. Feature normalization and inverse-frequency class weighting are fitted only on the four training folds.

The probe is trained using weighted cross-entropy and AdamW with learning rate 10^{-3}, zero weight decay, batch size 128, gradient-norm clipping at 5, and a maximum of 500 epochs. Early stopping uses training statistics only: after at least 20 epochs, training stops after five consecutive qualifying epochs with perfect training accuracy and loss no greater than 0.01. Held-out folds are never used for checkpoint selection. Final metrics are computed from pooled out-of-fold predictions.

Metrics. We report the predicted Sufficiency Rate, classification accuracy against the correctness-based label, Sufficiency Balanced Accuracy, and, where relevant, AUROC. Balanced accuracy is the mean recall over the sufficient and insufficient classes. An always-sufficient classifier therefore obtains a Sufficiency Rate of 100%, a balanced accuracy of 50%, and classification accuracy equal to the positive-label prevalence. The MLP probes are diagnostic supervised models and are not components of the inference-time TRACE router.

C.5 Complementary Retrieval Analysis

We complement the streaming analysis with NExT-GQA to study retrieval quality more directly. We use 3,777 test questions from 928 videos whose single annotated evidence interval has duration between 2 and 10 seconds, inclusive. Each full video is partitioned into five-second candidate segments, with a shorter final segment retained when necessary.

Each retrieval method selects two candidate segments. Eight frames from each selected segment, together with their actual source timestamps, are provided to Qwen3.5-9B. RC, cumulative summaries, and retrieval captions are excluded from the QA input. This experiment is an offline localization-and-QA evaluation rather than the causal streaming protocol used for OVO-Bench. Answers are generated deterministically as short text and subsequently parsed into multiple-choice options.

The evaluated retrieval methods include dense caption retrieval, VLM-generated evidence-need queries followed by dense retrieval, ColBERT, InternVideo2, and BM25. The separate Qwen3-TokenMaxSim experiment is reported independently and should not be conflated with ColBERT. Hit@2 indicates whether at least one selected segment has positive-duration overlap with the annotated evidence interval. Coverage measures the fraction of the annotated interval covered by the union of retrieved segments and is averaged over questions. Retrieval accuracy, downstream QA accuracy, storage cost, and query latency are reported as distinct measurements.

D Prompts

This section lists the textual templates used by the memory construction, retrieval, QA, and evidence-sufficiency experiments. The templates are shown before model-specific chat formatting is applied. Braced expressions denote substituted fields. `<VIDEO: ...>` and `<IMAGE: ...>` denote multimodal content parts rather than literal strings. Unless otherwise noted, we use `add_generation_prompt=True` and `enable_thinking=False`. The allowed answer letters are generated from the actual options and are not assumed to be fixed to four in the implementation.

D.1 Memory Construction Prompts

Chunk captioning. Each completed video chunk is accompanied by the following system message:

```
You are an event captioner for an online video memory.
Describe only concrete visual facts in this chronological 16-frame chunk.
Preserve chronology and include visible people, objects, actions, locations,
state changes, and immediate results.
Do not guess, and do not refer to any question or answer.
Output one self-contained paragraph only.
```

The corresponding user message is:

```
<VIDEO: 16 chronological chunk frames>
Caption this 16-frame video chunk.
```

Regular summary update. After four new chunks have been completed, the cumulative summary is updated using:

```
Update one cumulative video memory using the previous memory and four new
chronological chunk captions. Preserve concrete facts and chronology, remove
```

```
redundancy, add nothing, and output one self-contained paragraph only.

Previous memory:
{previous}

Four new chunk captions:
{captions}
```

At the first regular update, {previous} is (empty). The caption field consists of lines of the form Chunk {index}: {caption}.

Initial summary. When two or three completed chunks exist before the first regular four-chunk update, we use:

```
Summarize these {chunk_count} chronological video chunk captions into one
cumulative video memory. Preserve concrete facts and chronology, remove
redundancy, add nothing, and output one self-contained paragraph only.

Chunk captions:
{captions}
```

A single completed chunk uses its caption directly, while no completed chunk is represented as (No completed video chunk yet.).

Partial summary update. Summary incorporates a final incomplete group of completed chunks using:

```
Update one cumulative video memory using the previous memory and
{chunk_count} new chronological chunk captions.
Preserve concrete facts and chronology, remove redundancy, add nothing,
and output one self-contained paragraph only.

Previous memory:
{previous}

New chunk captions:
{captions}
```

This template is used only by Summary and is not applied retroactively to experiments using frozen original summaries.

D.2 QA AND RETRIEVAL PROMPTS

QA system message. The fixed-stage online QA experiments use:

```
You are evaluating online video multiple-choice question answering.
Use only the supplied evidence and do not assume unseen past or future
content. Evidence is ordered from the always-available recent view to
optional short-term memory, cumulative memory summary, and retrieved
historical chunks.
Answer with exactly one option letter and no explanation.
```

Recent Context.

```
Evaluation time: {query_time:.3f}s.
Available video endpoint: {effective_endpoint:.3f}s.

Always-available recent view, up to 8 frames at 2 FPS:
<VIDEO: Recent frames>

End of Recent evidence.
```

Sparse Memory extension. The next evidence level appends the sparse visual window and cumulative summary:

```
Additional non-overlapping sparse window before Recent:
<VIDEO: Sparse-window frames>

Cumulative Memory Summary:
{total_summary}

End of sparse-window and Summary evidence.
```

If no earlier sparse frame exists, the corresponding field is replaced by `(No earlier sparse-window frame exists.)`.

Original retrieved-memory extension. The historical sufficiency experiment appends:

```
Retrieved historical chunks:

Chunk {chunk_id} [{start:.1f}s-{end:.1f}s], caption: {caption}
<VIDEO: 8 uniformly sampled frames from this chunk>

End of retrieved evidence.
```

The chunk entry is repeated for all selected chunks in chronological order.

Caption-free retrieved evidence. For the controlled frames-plus-time presentation used by the final retrieval format, the retrieved-memory block becomes:

```
Retrieved historical chunks:

Segment {letter}, [{start:.1f}s-{end:.1f}s]
Segment {letter} frames:
<VIDEO: 8 uniformly sampled frames from this segment>

End of retrieved evidence.
```

The segment entry repeats in chronological order. Removing T removes its interval header; removing F removes the frame label and video content; enabling C appends `caption: {caption}` to the segment header. If no historical chunk is retrievable, the field contains `(No completed chunk is retrievable.)`. The shared-context baseline omits the retrieval block entirely.

QA suffix.

```
QUERY_START_7E4C:
Question: {question}
Options:
A. {option_a}; B. {option_b}; C. {option_c}; D. {option_d};

Respond only with the letter corresponding to your chosen option
(e.g., A, B, C).
Do not include any additional text or explanation in your response.
```

Dense retrieval query. The embedding model receives:

```
Question: {question}
Options:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}
```

This is an encoder input rather than a generative prompt. Neither the retrieval input nor the QA prompt contains the target answer. The uncertainty-based router operates directly on the QA model's predictive distribution and therefore requires no additional natural-language routing prompt.

D.3 EVIDENCE-PRESENTATION PROMPTS

The earlier independent-image analysis uses a separate QA format. Golden evidence is treated as an offline diagnostic because annotated evidence may occur after the query timestamp, whereas retrieved evidence follows the causal memory constraint.

Golden-evidence system message.

```
Answer the multiple-choice question using the supplied evidence.
This is an offline evidence QA task: evidence can occur before or after the
annotated query time.
Timestamps refer to the source video.
Output only the selected option letter.
```

Retrieved-evidence system message.

```
Answer the multiple-choice question using the supplied evidence from the
existing video memory.
All evidence is available by the original query time.
Timestamps refer to the source video.
Output only the selected option letter.
```

Evidence-only QA suffix.

```
Annotated query time: {query_time:.3f} seconds.
Question: {question}
Options:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}
Output only the selected option letter.
```

Conditional evidence fields.

```
Center chunk {chunk_id} [{start:.3f}, {end:.3f}) seconds.
Caption: {caption}

Frame from {source_id} at {timestamp:.6f} seconds:
<IMAGE: corresponding frame>
```

These fields are conditionally included according to the presentation configuration: chunk and frame-time annotations are enabled with explicit temporal evidence T, captions with C, and visual inputs with F. When captions and frames are jointly included, captions precede the image blocks.

Question-independent evidence captioning. When evidence captions are generated independently from the question, the system message is:

```
Describe the visible facts in these chronological frames: people, objects,
actions, locations, state changes, and immediate results.
Preserve chronology within this unit.
Be specific about visible details; state uncertainty where needed.
Do not guess unseen events or relate this unit to other units.
Output one concise factual paragraph only.
```

with the user input

```
<IMAGE: chronological evidence frame 1>
<IMAGE: chronological evidence frame 2>
<...remaining evidence images...>

Describe these frames in their displayed chronological order.
```

D.4 EVIDENCE-SUFFICIENCY PROMPTS

For explicit sufficiency analysis, the frozen visual and memory evidence is unchanged, while the standard QA instruction is replaced by one of the following diagnostic prompts.

Judge only.

```
You judge evidence sufficiency for online video QA.
Decide whether the supplied evidence already contains enough information to
determine the correct answer now.
Judge the evidence, not your confidence or general knowledge.
Missing, ambiguous, or unseen required evidence means insufficient.

Return exactly one JSON object and no prose:
{"label":"sufficient","confidence":0.00}

The label must be sufficient or insufficient.
Confidence is the probability that your chosen label is correct.
Do not answer the video question.
```

Judge then answer.

```
You judge evidence sufficiency and conditionally answer an online video
question.
First decide whether the supplied evidence already contains enough
information to determine the correct answer now.
Judge the evidence, not your confidence or general knowledge.
Missing, ambiguous, or unseen required evidence means insufficient.

Return exactly one JSON object and no prose.
If sufficient:
{"label":"sufficient","confidence":0.00,"answer":"one allowed option letter"}
If insufficient:
{"label":"insufficient","confidence":0.00,"answer":null}

Confidence is the probability that your chosen sufficiency label is correct.
Never guess an answer when evidence is insufficient.
```

Shared judgment suffix.

```
QUERY_START_7E4C:
Question: {question}
Options:
A. {option_a}; B. {option_b}; C. {option_c}; D. {option_d};
Allowed answers: A, B, C, D
```

Allowed answer letters are generated from the actual set of options.

Fallback output contract. Malformed outputs are rerun with an explicit contract and constrained JSON decoding. For judge-only evaluation, we append:

```
OUTPUT CONTRACT:
Judge evidence sufficiency.
The label field MUST be exactly sufficient or insufficient, never an option
letter.
Option letters above belong to the video question, not to the sufficiency
label.
Do not answer the video question.
Return only label and confidence in JSON.
```

For joint judgment and answering, the ending is replaced with:

```
Return label, confidence and answer in JSON.
```

```
Only when label is sufficient may answer contain an option letter;
otherwise answer must be null.
```

Constrained decoding permits only the two valid sufficiency labels, confidence values from 0.00 to 1.00 in increments of 0.01, and, in the joint setting, a valid option letter when sufficient or `null` when insufficient. “‘