BEFORE THE QUESTION ARRIVES: CAUSAL EVIDENCE BUDGETING FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding is crucial for AI systems to interpret dynamic environments, yet providing accurate, real-time responses remains challenging. Models must process a growing visual history within limited context and memory budgets. Existing methods reduce this pressure through visual-token compression or key–value (KV) cache management, but they often fix what to retain at frame arrival, before considering earlier and later evidence together. Such early decisions can discard evidence needed by future questions, and later retrieval cannot recover it. To address this mismatch, we introduce StreamingRETAKE, a training-free framework for causal evidence budgeting that allocates visual-token and KV capacity over time. Visual-token capacity is directed toward informative frames to build a compact evidence pool, from which shared evidence is selected jointly after more frames arrive. Then retrieval combines evidence importance with question relevance to build the answer context. The framework bounds visual-history storage and answer contexts as the stream grows. Extensive experiments show that our proposed method achieves state-of-the-art accuracy of 75.52% and 81.80% on OVO-Bench Real-Time and StreamingBench Real-Time, respectively.

1 INTRODUCTION

Streaming video understanding (Lin et al., 2024; Li et al., 2025) requires continuously processing incoming frames and providing timely, accurate answers to questions that may arrive at any time, accounting for both ongoing events and earlier observations. This capability supports applications such as interactive assistants and continuous monitoring. Multimodal large language models (MLLMs) (Li et al., 2024a; Bai et al., 2025b;a) are a natural fit for this setting given their open-ended question answering, but they must store and process a growing visual history, which requires increasing memory and computation. To reduce these costs, existing training-free methods compress visual tokens before language-model prefill or manage key–value (KV) caches through compression and retrieval (Chen et al., 2026a; Zhang et al., 2026; Di et al., 2025). However, they leave a key challenge unresolved: how to use limited capacity to preserve information needed by future questions.

In a streaming setting, deciding what to retain as each frame arrives is difficult: the model lacks both later context and knowledge of future questions. Existing training-free methods rely on different selection signals, such as visual redundancy for token compression, attention-based importance for KV retention, or question relevance for KV retrieval (Chen et al., 2026a; Zhang et al., 2026; Di et al., 2025). However, under a bounded memory budget, they all consume capacity as frames arrive, and the question can only take effect after the retained history is fixed. Such decisions are irreversible, since even accurate retrieval cannot recover evidence that was already discarded. The central challenge is therefore how to allocate limited capacity across arriving frames and accumulating evidence to preserve a history that supports diverse future questions.

Our analysis of evidence selection in streaming videos reveals three limitations of existing strategies. First, fixed per-frame token budgets ignore how unevenly visual content changes: just 20% of frames account for about 65% of total frame-to-frame change. Second, later frames can make earlier evidence visually redundant, so long-term retention should not be fixed at frame arrival. Third, questions are unknown during compression, so it should preserve evidence shared across questions

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.

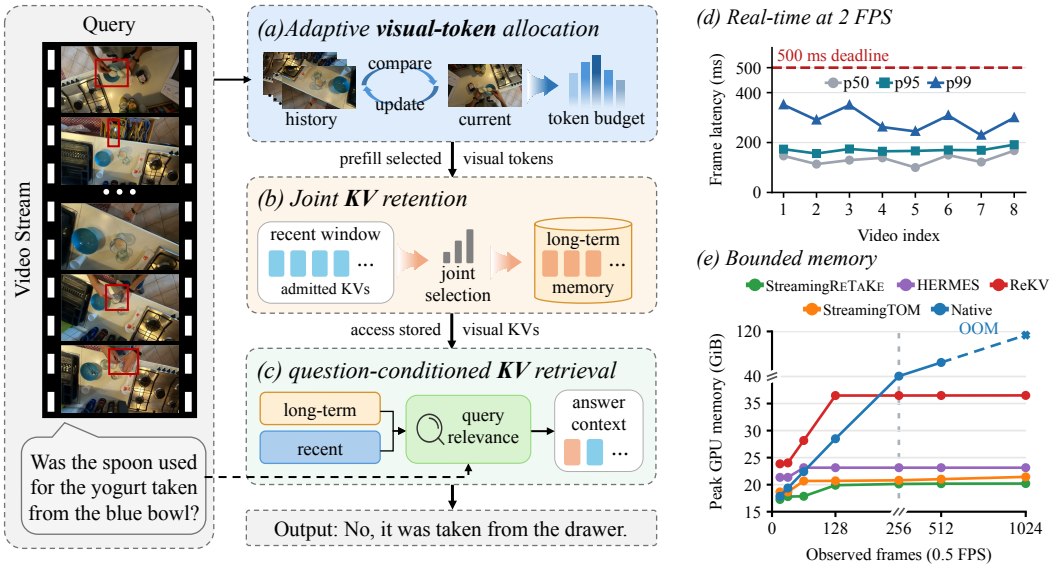


Figure 1: (a)–(c) Pipeline of StreamingReTaKe. (d) StreamingReTaKe enables real-time video processing at 2 FPS on Qwen3-VL-8B. (e) StreamingReTaKe achieves the lowest peak GPU memory among the compared streaming methods on LLaVA-OV-7B.

for later retrieval. These limitations have lasting effects because capacity wasted at input or evidence lost at retention cannot be recovered at answer time.

To address these limitations, we introduce **StreamingReTaKe**, a training-free framework for *causal evidence budgeting*. It jointly budgets visual tokens and KV memory through adaptive allocation and delayed KV selection to preserve shared evidence for future questions. (Figure 1(a)–(c)). Before prefill, *adaptive visual-token allocation* transfers capacity from redundant to more informative frames. During streaming prefill, *joint KV retention* keeps admitted KVs temporarily available and delays selection for long-term memory, allowing earlier and later evidence to compete for shared capacity. When a question arrives, *question-conditioned KV retrieval* combines the importance scores used for retention with question relevance to construct compact answer contexts without replaying or modifying the stored history. StreamingReTaKe bounds visual-history KVs in both persistent storage and answer contexts regardless of stream duration.

Extensive experiments demonstrate the effectiveness of StreamingReTaKe. It achieves state-of-the-art real-time accuracy of 75.52% on OVO-Bench and 81.80% on StreamingBench. On one H100, Qwen3-VL-8B meets 99.83% of frame deadlines at 2 FPS. With LLaVA-OneVision-7B at 0.5 FPS, it achieves 148 ms median TTFT with a peak GPU memory footprint of 20.2 GB over 1,024 frames.

Overall, our contributions can be summarized as follows:

- We introduce StreamingReTaKe, a training-free framework for streaming video understanding that preserves reusable history for future questions and constructs compact answer contexts with bounded visual-history KV storage.
- Our adaptive visual-token allocation transfers capacity from redundant to more informative frames, joint KV retention lets accumulated evidence compete for shared capacity, and question-conditioned KV retrieval combines retention scores with question relevance.
- Across four models, StreamingReTaKe achieves state-of-the-art real-time accuracy, generalizes to offline video understanding, and sustains real-time processing with faster responses.

2 Related Work

2.1 Streaming Video Understanding

Streaming video understanding requires continuous perception, timely interaction, and memory of past events. VideoLLM-online, MMDuet, ViSpeak, and Streamo develop proactive responses and

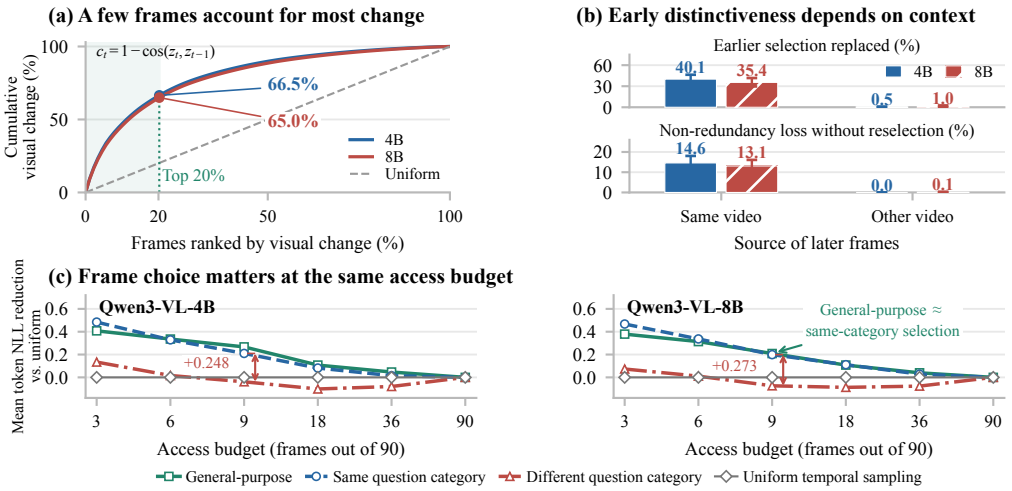


Figure 2: Visual change and evidence selection in Qwen3-VL-4B/8B. (a) The 20% of frames that differ most from their predecessors account for about 65% of total visual change. (b) Later context changes which earlier frames remain distinctive. Reselection yields higher visual non-redundancy than the original selection. (c) Shared frame selection supports full-history response prefixes across question categories, but category mismatch weakens this support.

continuous interaction (Chen et al., 2024; Wang et al., 2024b; Fu et al., 2025; Xia et al., 2025). StreamChat, StreamForest, OASIS, and StreamAgent organize history through structured memory and retrieval (Xiong et al., 2025; Zeng et al., 2025; Liang et al., 2026; Yang et al., 2025). For resource efficiency, FluxMem compresses visual tokens (Xie et al., 2026), ReKV and LiveVLM retrieve stored video KVs (Di et al., 2025; Ning et al., 2026), HERMES maintains hierarchical KV memory (Zhang et al., 2026), and Memento combines dynamic memory with question-related selection (Jiang et al., 2026). StreamingTOM uses a fixed per-frame token budget and 4-bit quantization to reduce processing and storage costs (Chen et al., 2026a). StreamKV organizes video into semantic segments and applies layer-adaptive KV compression and retrieval (Chen et al., 2026b). MuKV combines patch-, frame-, and segment-level KV compression with semi-hierarchical retrieval (Xiao et al., 2026). StreamingReTaKe coordinates adaptive visual-token allocation, joint KV retention, and question-conditioned KV retrieval under explicit input and memory budgets, preserving a common causal memory before future questions are known.

2.2 Long Video Understanding

Long-video compression reduces the visual sequence and decoder state needed to represent extended footage. LongVU and FastVID exploit spatiotemporal redundancy to reduce visual tokens (Shen et al., 2024; 2025). DyCoke combines temporal token merging with dynamic KV reduction, while HoliTom integrates token merging before and within the LLM (Tao et al., 2025; Shao et al., 2025). ReTaKe links visual keyframe selection with pivot-guided KV compression to reduce temporal and knowledge redundancy (Wang et al., 2024a). AdaReTaKe further adapts compression ratios across time and model layers (Wang et al., 2025). These methods expand the video content accommodated within a finite context. Streaming adds a further challenge: evidence must be selected before later observations and questions reveal its value. StreamingReTaKe addresses this uncertainty through adaptive visual-token allocation, joint KV retention, and question-conditioned KV retrieval.

3 Preliminary Analysis

To inform evidence allocation across arriving frames, accumulated history, and eventual questions, we analyze how visual change, later context, and question relevance affect selection decisions. We examine 32 unlabeled videos (16 each from StreamingBench and Video-MME) processed at 1 FPS for 90 seconds using frozen Qwen3-VL-4B/8B (Bai et al., 2025a; Lin et al., 2024; Fu et al., 2024).

3.1 A Few Frames Account for Most Changes in Visual Content

Changes in visual content are uneven across frames: the 20% of frames that differ most from their predecessors account for about 65% of the total change. To quantify this change, let z_t be the normalized mean projected visual representation of frame t. We measure each frame's visual change relative to its predecessor as $c_t = 1 - \langle z_t, z_{t-1} \rangle$. For each video, we rank frames by c_t and measure the share of total change contributed by the top 20%. As shown in Figure 2(a), this share is 66.5% for 4B and 65.0% for 8B, versus 20% if change were uniform.

Equal token budgets overlook differences in visual change. A fixed per-frame allocation assigns equal capacity regardless of how much frames differ from their predecessors. This motivates allocating more capacity to frames with greater visual change under a bounded long-run budget.

3.2 Later Frames Reveal Redundancy in Earlier Evidence

We ask whether evidence should be selected for long-term memory at arrival or after observing more frames. For the same earlier candidate frames, we recompute visual non-redundancy after adding later frames to the scoring context. Non-redundancy is measured as one minus the maximum cosine similarity to another context frame, excluding the candidate itself. For the initial and updated n-frame sets S_0 and S_1, we measure

$$\text{Replaced} = 1 - \frac{|S_0 \cap S_1|}{n}, \qquad \text{ScoreLost} = 1 - \frac{\sum_{t \in S_0} \nu_t}{\sum_{t \in S_1} \nu_t}. \tag{1}$$

Here, ν_t is frame t's non-redundancy score against the expanded context for both sets.

As shown in Figure 2(b), later context changes 40.1% of the 4B selection and 35.4% of the 8B selection. Keeping the original selection loses 14.6% and 13.1% of the reselected set's total non-redundancy score, respectively. Adding the same number of frames from another video leaves the selection largely unchanged, with negligible score loss. Thus, later evidence can influence which earlier observations are selected for long-term memory.

Long-term selection should remain open as evidence accumulates. Keeping admitted evidence temporarily available allows earlier and later observations to compete for shared capacity.

3.3 Evidence Is Shared Across Questions but Varies in Relevance

Under a fixed frame-access budget, it is unclear whether answer-supporting evidence is shared across questions or is question-specific. We compare four selections from the same history at equal capacity. *General-purpose* selection chooses frames with the highest average importance for general questions; *same-category* and *different-category* selection use the target category or another category; *uniform sampling* selects evenly spaced frames. Selection and evaluation use disjoint prompt paraphrases; the general-purpose set is shared across categories within each video.

For question q and selected frames S, $L(q, S)$ denotes the mean token negative log-likelihood (NLL) of a response prefix generated with the full history. Figure 2(c) reports

$$\Delta_{\text{NLL}}(q, S) = L(q, U_n) - L(q, S), \qquad |S| = |U_n| = n, \tag{2}$$

where U_n is the uniformly sampled n-frame set. A positive value means that the selected context better supports the same response prefix.

Both general-purpose and same-category selection yield positive Δ_{NLL} across all tested sub-full budgets for both models. At the nine-frame budget, general-purpose selection uses one frame set shared across categories yet achieves lower NLL than same-category selection for 4B and similar NLL for 8B. Question relevance still matters. Different-category selection increases NLL by 0.248 for 4B and 0.273 for 8B relative to same-category selection, showing that evidence selected for one category may provide weaker support for another. These findings motivate preserving shared evidence before questions arrive and using question relevance to guide later retrieval.

From observations to memory design. These observations call for allocating capacity to informative frames while keeping admitted evidence temporarily available. Delaying long-term KV selection allows earlier and later evidence to compete for shared capacity. Memory should preserve shared evidence for future questions and use question relevance to build compact answer contexts.

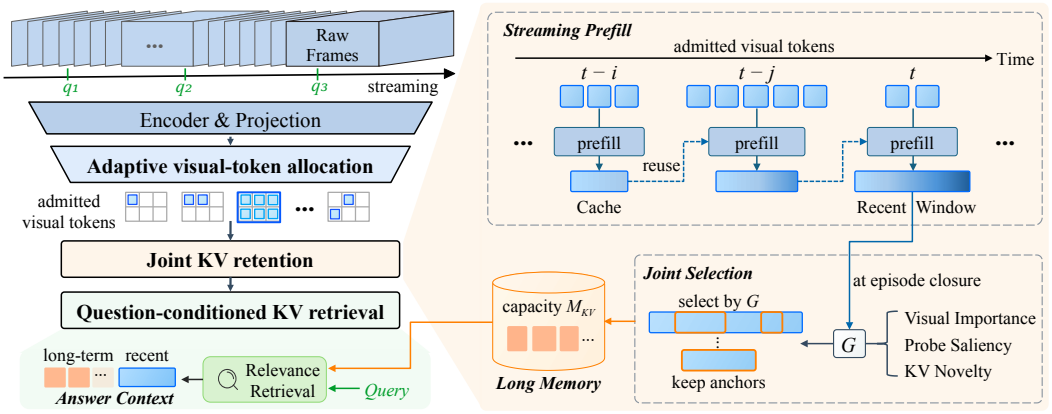


Figure 3: Detailed pipeline of StreamingReTaKe. Left: adaptive visual-token allocation, joint KV retention, and question-conditioned KV retrieval. Right: streaming prefill and joint selection at episode closure; selected KVs enter long-term memory when their frames leave the recent window.

4 StreamingReTaKe

As shown in Figure 3, StreamingReTaKe uses recent memory to keep evidence admitted by visual-token allocation available for joint retention. Covering the latest W frames, it allows earlier and later evidence to compete before long-term selection is finalized. Long-term memory preserves shared evidence for future questions within a capacity of M_{KV} visual and control KVs. Question-conditioned retrieval combines persistent importance with question relevance to select from this shared history, then adds recent memory to form the answer context.

4.1 Adaptive Visual-Token Allocation

StreamingReTaKe transfers visual-token capacity from redundant to informative frames under an average budget of B visual tokens per frame. For the projected visual tokens $X_t = \{x_{t,i}\}_{i=1}^{N}$ of frame t, the novelty score s_t measures visual change relative to the preceding frame and evolving visual prototypes updated as new frames are written. Its percentile u_t uses past observations, and r_t indicates a novelty record over up to W preceding frames.

Algorithm 1 Adaptive visual-token allocation

Require: Frame X_t, prior state Z, credit C
$(s_t, u_t) \leftarrow \textsc{Novelty}(X_t, \mathsf{Z})$
$r_t \leftarrow \textsc{NewRecord}(s_t, \mathsf{Z})$
$\alpha_t \leftarrow \mathbf{1}[t = 1 \vee (r_t \wedge C = N - B)]$
if $\alpha_t = 1$ **then** ▷ Full anchor
  $b_t \leftarrow N$
else
  $b_t \leftarrow f + \lfloor 2(B - f)u_t \rfloor$
$b_t \leftarrow \min(N, b_t, B + C)$
$C \leftarrow \min(N - B, C + B - b_t)$
$I \leftarrow \textsc{VisualImportance}(X_t, \mathsf{Z})$
$\mathcal{S}_t \leftarrow \textsc{TopK}(I, b_t)$
$\mathcal{S}_t \leftarrow \textsc{SourceOrder}(\mathcal{S}_t)$
return $\mathcal{S}_t, I|_{\mathcal{S}_t}, \alpha_t, C$

A budget bank carries unused capacity forward through credit $C \in [0, N - B]$, initialized at $N - B$. The first frame and novelty records with a full bank receive all N tokens as complete *anchors* ($\alpha_t = 1$). Other frames receive a budget that grows with u_t from a minimum of $f = \max(1, \lfloor B/2 \rfloor)$, up to $\min(N, B + C)$. Algorithm 1 specifies the allocation and credit update, which keep the long-run average at most B tokens per frame.

Given the token budget b_t, we select tokens using prototype distance δ^{ref}, local change δ^{mot}, and projected feature magnitude $\|x_{t,i}\|_2$. We use percentile rank $R_{\mathcal{A}} \in [0, 1]$ over a set $\mathcal{A}$ to align the three signal scales, with higher ranks taking priority. Visual importance is defined as

$$I_{t,i} = R_{\mathcal{U}_t}(\delta^{\mathrm{ref}}_{t,i}) + R_{\mathcal{U}_t}(\delta^{\mathrm{mot}}_{t,i}) + R_{\mathcal{U}_t}(\|x_{t,i}\|_2), \tag{3}$$

where $\mathcal{U}_t = \{1, \ldots, N\}$ indexes frame t's visual tokens and $I = (I_{t,1}, \ldots, I_{t,N})$ collects their scores. We select the top b_t token indices $\mathcal{S}_t$ by $I_{t,i}$ and restore source order before prefill.

At each refresh, the prototypes are initialized from the selected tokens after current-frame scoring. Between refreshes, redundant dropped-token features update the prototypes, allowing later scoring to incorporate intervening observations. The first frame initializes the prototypes; novelty records and the W-frame episode limit trigger subsequent refreshes. Appendix B.1 provides details.

4.2 JOINT KV RETENTION

To preserve shared evidence for future questions, StreamingReTaKe keeps long-term KV selection open as evidence accumulates. As shown in the right panel of Figure 3, streaming prefill appends admitted KVs to the W-frame recent window. Long-term selection is organized into episodes of at most W frames. At episode closure, earlier and later candidates compete under a shared retention quota. We use K for the target average number of retained visual and control KVs per frame.

A fixed query-agnostic probe provides attention-based saliency $A^{\ell}_{t,i}$ to identify evidence useful across questions. Persistent importance $G_{t,i}$ combines this signal with visual importance $I_{t,i}$ and reference-relative KV novelty $D^{\ell}_{t,i}$. The KV reference holds the KVs of the tokens used to initialize the prototypes until refresh. Non-anchor residuals compare pre-RoPE keys and values with the KV reference; anchors receive unit residuals. Across L decoder layers,

$$G_{t,i} = \frac{1}{L}\sum_{\ell=1}^{L} R_{\mathcal{S}_t}(A^{\ell}_{t,i}) + \frac{1}{L}\sum_{\ell=1}^{L} R_{\mathcal{S}_t}(D^{\ell}_{t,i}) + R_{\mathcal{S}_t}(I_{t,i}). \tag{4}$$

Each component ranks over that frame's written tokens, yielding a relative retention priority. All layers share the retained indices.

At episode closure, anchors and frame controls are accounted for first. Non-anchor tokens with the highest G share the remaining visual quota. Complete anchors are kept even if they exceed the episode's K-based target. Later candidates influence which earlier tokens earn capacity.

When a frame leaves the recent window, only its KVs selected at episode closure enter long memory. If long memory exceeds M_{KV}, we evict non-anchor tokens with the lowest retention priority across long memory until the limit is met. Anchors are considered only if further eviction is needed.

4.3 QUESTION-CONDITIONED KV RETRIEVAL

When a question arrives, StreamingReTaKe combines persistent importance from retention with question relevance to retrieve long-term evidence for a compact answer context. It preserves the complete recent window and fills the remaining capacity with the retrieved evidence. The answer uses a fork of the stream state, so retrieval and generation leave persistent memory unchanged. If a question-time frame is supplied, the branch first processes it through the same visual-token allocation and window updates; recent occupancy C_R is measured afterward.

Question-conditioned retrieval uses the question as a query to score stored pre-RoPE keys, providing a lightweight ranking without replaying the video history. Let $a^{\ell}_{i}(q)$ be the normalized query–key compatibility score for token i, averaged over question tokens and heads. $\mathcal{L}_q$ denotes the scoring layers, and R ranks long-term visual tokens:

$$J_i(q) = \frac{1}{|\mathcal{L}_q|}\sum_{\ell\in\mathcal{L}_q} R(a^{\ell}_{i}(q)), \qquad S_i(q) = R(G_i) + R(J_i(q)). \tag{5}$$

Persistent importance G_i favors evidence useful across questions, while question relevance $J_i(q)$ directs retrieval toward the current question.

If recent memory occupies C_R visual and control KVs, long-term selection receives $\max(0, M_{\mathrm{KV}} - C_R)$ slots. Candidates are admitted in descending $S_i(q)$ order with their required controls, then combined with recent evidence in chronological order.

Unified resource guarantees. The budget bank links visual-token allocation to persistent storage and the answer context. These bounds count visual history KVs and frame controls. Let C_t denote credit after frame t. Starting from $C_0 = N - B$, the bank updates $C_t = \min\{N - B, C_{t-1} + B - b_t\}$ and caps admission at $b_t \le B + C_{t-1}$, ensuring $0 \le C_t \le N - B$. Let $w_t = \min(t, W)$ and $\mathcal{M}_t = \mathcal{M}^{\mathrm{long}}_t \,\|\, \mathcal{M}^{\mathrm{recent}}_t$. Summing over the recent window gives

$$|\mathcal{M}^{\mathrm{long}}_t| \le M_{\mathrm{KV}}, \qquad |\mathcal{M}^{\mathrm{recent}}_t| \le w_t(B+2) + N - B \le W(B+2) + N - B. \tag{6}$$

The two additional KVs per frame are its vision-start/end controls. An answer branch inherits the bank credit and applies the same window updates to a question-time frame. Its visual-history answer context satisfies

$$|\mathcal{C}_q| \le \max(M_{\mathrm{KV}}, C_R) \le \max(M_{\mathrm{KV}}, W(B+2) + N - B)\,. \tag{7}$$

Table 1: Accuracy (%) comparison of streaming video understanding methods on OVO-Bench and StreamingBench. Add-on methods use the preceding backbone. Avg. averages the Real-Time and Backward scores. We report the better HERMES results from its 4K- and 6K-token configurations.

Method	Size	Frames	OVO-Bench			StreamingBench
			Real-Time	Backward	Avg.	Real-Time
Proprietary Models						
Gemini 1.5 Pro	–	1 fps	69.30	62.54	65.92	75.70
GPT-4o	–	64	64.50	60.75	62.63	73.30
Open-source Offline MLLMs						
LongVU	7B	1 fps	57.40	35.01	46.21	–
LLaVA-Video	7B	64	63.50	40.40	51.95	–
InternVL2	8B	64/16	60.40	43.44	51.92	63.70
Open-source Online MLLMs (Training-Based)						
VideoLLM-online	8B	2 fps	20.80	17.73	19.27	36.00
Dispider	7B	1 fps	54.60	36.06	45.33	67.60
StreamForest	7B	1 fps	61.20	52.02	56.61	77.30
Streamo	7B	1 fps	65.98	46.10	56.04	–
Open-source Online MLLMs (Training-Free)						
LLaVA-OV	7B	64	63.50	40.40	51.95	71.34
+ ReKV	7B	0.5 fps	57.33	44.16	50.75	69.10
+ StreamingTOM	7B	0.5 fps	63.60	45.01	54.31	70.10
+ HERMES	7B	0.5 fps	66.34	**50.20**	**58.27**	73.23
+ StreamingReTaKe	7B	0.5 fps	**66.78**	48.30	57.54	**73.99**
Qwen2.5-VL	7B	1 fps	63.30	41.90	52.60	73.90
+ HERMES	7B	1 fps	68.98	49.43	59.21	**79.44**
+ FluxMem	7B	1 fps	67.20	47.24	57.22	76.40
+ StreamingReTaKe	7B	1 fps	**71.01**	**51.18**	**61.10**	77.55
Qwen3-VL	4B	1 fps	68.23	44.72	56.47	73.24
+ StreamingReTaKe	4B	1 fps	**70.45**	**52.75**	**61.60**	**77.80**
Qwen3-VL	4B	2 fps	70.67	50.05	60.36	78.32
+ HERMES	4B	2 fps	72.32	55.03	63.68	78.40
+ StreamingReTaKe	4B	2 fps	**75.52**	**55.11**	**65.32**	**81.52**
Qwen3-VL	8B	1 fps	68.89	44.76	56.83	74.12
+ StreamingReTaKe	8B	1 fps	**69.14**	**53.18**	**61.16**	**77.48**
Qwen3-VL	8B	2 fps	68.64	47.03	57.84	78.92
+ HERMES	8B	2 fps	73.29	49.29	61.29	81.32
+ StreamingReTaKe	8B	2 fps	**74.61**	**52.32**	**63.46**	**81.80**

Thus, B governs both admission and recent-memory residency, while M_{KV} caps long-term storage and sets the target answer capacity. These bounds make visual-history KV storage and answer-context occupancy independent of stream duration. Appendix B provides the algorithms and proofs.

5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate streaming video understanding on OVO-Bench (Li et al., 2025) and StreamingBench (Lin et al., 2024). We consider the Real-Time Visual Perception and Backward Tracing tasks of OVO-Bench, and the Real-Time Visual Understanding tasks of StreamingBench, covering both current-scene perception and access to historical evidence. To assess generalization to offline video understanding, we additionally evaluate on MVBench (Li et al., 2024b), Video-MME (Fu et al., 2024), EgoSchema (Mangalam et al., 2023), and the validation split of LongVideoBench (Wu et al., 2024). They cover temporal understanding, video comprehension, egocentric reasoning, and long-context reasoning.

Models. We apply StreamingReTaKe to LLaVA-OneVision-7B (Li et al., 2024a), Qwen2.5-VL-7B (Bai et al., 2025b), and Qwen3-VL-4B/8B (Bai et al., 2025a) for streaming evaluation. Offline evaluation uses LLaVA-OneVision-7B. All backbones remain frozen, with no additional training. Comparisons include proprietary models, open-source offline and online models, and training-free streaming methods, including ReKV (Di et al., 2025), HERMES (Zhang et al., 2026), and FluxMem (Xie et al., 2026).

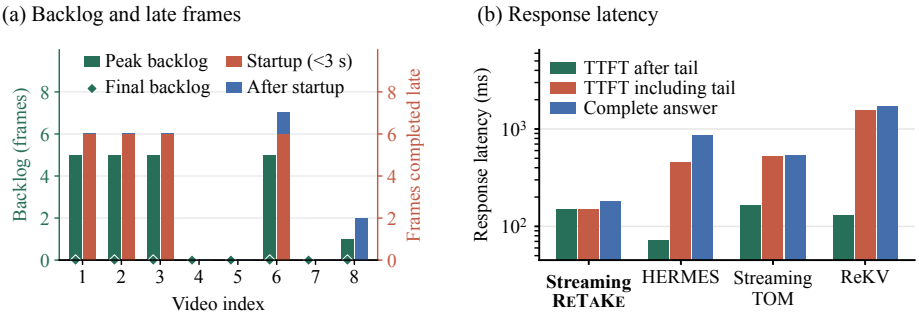


Figure 4: Streaming service and response latency. (a) Qwen3-VL-8B at 2 FPS: peak/final backlog (left axis) and late frames (right axis), split at 3s into startup and post-startup. Every video ends with zero backlog. (b) LLaVA-OneVision-7B median response latency; "tail" denotes pending frames. Complete-answer times reflect each method's output length.

Table 2: **Offline video understanding.** Accuracy (%) on MVBench, EgoSchema, Video-MME, and LongVideoBench validation. Add-on methods use LLaVA-OV-7B. HERMES: best reported results.

Model	Frames	MVBench	EgoSchema	Video-MME		LongVideoBench
				Long	Avg.	Val
LLaVA-OV-7B	64	57.02	59.93	48.00	57.67	56.50
+ ReKV	0.5 fps	56.83	60.70	46.89	57.74	55.80
+ HERMES	0.5 fps	56.92	60.29	**49.22**	**58.85**	–
+ LiveVLM	0.2/0.5 fps	–	63.00	48.80	57.30	56.10
+ StreamingReTaKe	0.5 fps	**56.95**	**63.20**	49.00	58.37	**57.00**

Implementation details. Our experiments are conducted on NVIDIA H100 GPUs. We use $(B, K, W, M_{\mathrm{KV}}) = (128, 32, 8, 4894)$ for LLaVA-OV and $(64, 16, 16, 4096)$ for Qwen2.5-VL-7B as reference settings. Qwen3-VL-4B/8B share the reference setting $(72, 24, 32, 4096)$. Reference sampling rates are 0.5 FPS for LLaVA-OV, 1 FPS for Qwen2.5-VL, and 2 FPS for Qwen3-VL.

5.2 Main Results

Streaming video understanding. Table 1 shows that StreamingReTaKe improves OVO-Bench average accuracy and StreamingBench accuracy over the backbones across four configurations, without additional training. It achieves 75.52% on OVO-Bench Real-Time with Qwen3-VL-4B and 81.80% on StreamingBench with Qwen3-VL-8B, the table's highest real-time accuracies. The 4B model reaches 81.52% on StreamingBench, exceeding HERMES by 3.12 percentage points at the same 2-FPS rate. Gains extend to historical reasoning: Qwen3-VL-4B reaches 65.32% OVO-Bench average accuracy, while Qwen3-VL-8B improves Backward Tracing over HERMES by 3.03 percentage points at the same 2-FPS rate. At matched backbone and frame settings, StreamingReTaKe exceeds HERMES in OVO-Bench average accuracy by 1.89, 1.64, and 2.17 points on Qwen2.5-VL-7B and Qwen3-VL-4B/8B, respectively. These results show causal evidence budgeting supports current-scene perception and access to earlier events across model families and sizes.

Offline video understanding. Table 2 evaluates whether the same streaming memory preserves video understanding across temporal scales. On EgoSchema, StreamingReTaKe achieves 63.20% with LLaVA-OneVision-7B, the best result among the compared training-free methods, exceeding the backbone and HERMES by 3.27 and 2.91 percentage points, respectively. On LongVideoBench validation, it achieves the highest reported accuracy in the table at 57.00%, exceeding the backbone by 0.50 points. On Video-MME, it improves over the backbone by 0.70 points overall and 1.00 point on long videos, reaching 58.37% and 49.00%. It also maintains short-video understanding on MVBench, scoring 56.95% compared with the backbone's 57.02%. These results show that compact causal memory retains useful evidence for video understanding beyond streaming benchmarks.

5.3 Efficiency Analysis

Sustained real-time processing. StreamingReTaKe sustains continuous video ingestion while answering interleaved questions. We feed Qwen3-VL-8B a wall-clock-paced 2-FPS stream on one

H100, covering eight long videos, 15,638 frames, and 40 questions. As shown in Figure 1(d), 99.83% of frames meet the 500 ms arrival deadline, with a 95th-percentile latency of 179 ms. All videos clear their queues and end with zero backlog (Figure 4(a)), showing that ingestion keeps pace with the stream.

Response latency and memory. Figure 4(b) compares LLaVA-OneVision-7B methods at 0.5 FPS on one H100. Across 185 questions from 37 videos, StreamingReTaKe's median time to first token (TTFT) is 148 ms, including frames pending at question arrival: $3.1\times$, $3.5\times$, and $10.5\times$ faster than HERMES, StreamingTOM, and ReKV, respectively. On a shared video extended to 1,024 frames, its physical GPU memory peak stays near 20.2 GiB, the lowest among the compared streaming methods (Figure 1(e)). Bounded memory and access enable prompt responses.

5.4 Ablation Studies

Table 3: Component ablations on StreamingBench with Qwen3-VL-8B.

Stage	Setting	Acc. (%)
Full StreamingReTaKe		**81.80**
Write	Fixed token budget	80.68
	Uniform tokens	80.76
Retain	Arrival-locked retention	80.60
	FIFO eviction	79.76
Read	Importance only	79.92
	Question relevance only	80.20
	Full cache	80.64

Table 4: Resource sensitivity on StreamingBench with Qwen3-VL-8B.

Setting	Acc. (%)	Δ (pp)
Full StreamingReTaKe	**81.80**	–
$B = 64$	81.12	-0.68
$K = 16$	81.08	-0.72
$K = 32$	81.16	-0.64
$W = 16$	79.44	-2.36
$W = 64$	69.18	-12.63
$M_{\mathrm{KV}} = 2048$	70.02	-11.78
$M_{\mathrm{KV}} = 8192$	80.12	-1.68
Query layers: deepest 10%	81.36	-0.44

Evidence selection across stages. Table 3 tests selection at each stage. Adaptive budgets improve accuracy by 1.12 percentage points over a fixed 72-token budget per frame, while importance-based token selection adds 1.04 points over uniform selection.

Long-term commitment. Arrival-locked retention selects each frame's long-term tokens at arrival, whereas Full selects jointly at episode closure. Both use the same scoring rule, writing policy, $(B, K, W) = (72, 24, 32)$, complete recent KVs, migration, global eviction, and Read rules, with 4,096 long-memory KVs. The control assigns each quota increment to the arriving frame, matching Full's total non-anchor visual quota after anchor and control accounting. Joint allocation reaches 81.80%, compared with 80.60% for arrival-locked retention and 79.76% for FIFO eviction.

Question-time selection. For reading, combining persistent importance and question relevance achieves 81.80%, compared with 79.92% using importance alone and 80.20% using question relevance alone. The compact combined selection also exceeds full-cache reading by 1.16 points.

Resource allocation. Table 4 shows that accuracy stays within 0.72 points of the default when reducing B to 64 or setting K to 16 or 32. Query scoring with the deepest 10% of layers reaches 81.36%. Reducing W to 16 gives 79.44%, whereas $W = 64$ and $M_{\mathrm{KV}} = 2048$ drop to 69.18% and 70.02%, respectively. Both reduce the answer capacity available for long-term evidence; at nominal allocation, a full recent window exceeds the answer target in either setting (Appendix D.1). Increasing M_{KV} to 8,192 also lowers accuracy by 1.68 points, favoring balanced capacity allocation.

6 Conclusion

We presented StreamingReTaKe, a training-free framework coordinating adaptive visual-token allocation, joint KV retention, and question-conditioned KV retrieval for streaming video understanding. It allocates capacity to visual change and delays long-term commitment to preserve evidence for future questions, then combines persistent importance with question relevance to select compact answer contexts. Visual-token and memory budgets bound visual-history KVs in storage and answer contexts as streams grow. Experiments across multiple video MLLMs demonstrate improved streaming accuracy and transfer to offline video understanding, alongside sustained real-time processing, prompt responses, and compact memory use.

Reproducibility statement

The main text describes the proposed method, evaluation protocols, and experimental settings. The appendix provides detailed algorithms, resource-bound derivations, preliminary-analysis protocols, ablation settings, and additional results.

AI use statement

We used generative AI tools to improve the manuscript's language, organization, and presentation of research ideas; to assist with implementing methods and evaluation code; and to analyze and interpret experimental results. The authors reviewed and revised the AI-assisted material and take responsibility for the final text, code, analyses, and claims.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2406.11816.

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*, 2026a. URL https://arxiv.org/abs/2510.18269.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. StreamKV: Streaming video question-answering with segment-based KV cache retrieval and compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3120–3128, 2026b. doi: 10.1609/aaai.v40i4.37305. URL https://doi.org/10.1609/aaai.v40i4.37305.

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*, 2025. URL https://arxiv.org/abs/2503.00540.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. URL https://arxiv.org/abs/2405.21075.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. ViSpeak: Visual instruction feedback in streaming videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2503.12769.

Hongxiang Jiang, Zengrui Ge, Guo Chen, Qixiong Wang, Jile Jiao, Xuetao Feng, Yuan Wang, and Yan Wang. Memento: Toward an all-day proactive assistant for ultra-long streaming video. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FtdbdoGbk3.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a. URL https://arxiv.org/abs/2408.03326.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. URL https://arxiv.org/abs/2311.17005.

Yifei Li, Junbo Niu, 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? *arXiv preprint arXiv:2501.05510*, 2025. URL https://arxiv.org/abs/2501.05510.

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*, 2026. URL https://arxiv.org/abs/2604.17052.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. StreamingBench: Assessing the gap for MLLMs to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*, 2024. URL https://arxiv.org/abs/2411.03628.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2308.09126.

Zhenyu Ning, Guangda Liu, Qihao Jin, Chengwei Li, Wenchao Ding, Minyi Guo, and Jieru Zhao. LiveVLM: Efficient online video understanding via streaming-oriented KV cache and retrieval. In *Design Automation Conference (DAC)*, 2026. doi: 10.1145/3770743.3804012. URL https://arxiv.org/abs/2505.15269.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models. *arXiv preprint arXiv:2505.21334*, 2025. URL https://arxiv.org/abs/2505.21334.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*, 2025. URL https://arxiv.org/abs/2503.11187.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024. URL https://arxiv.org/abs/2410.17434.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2411.15024.

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. ReTaKe: Reducing temporal and knowledge redundancy for long video understanding. *arXiv preprint arXiv:2412.20504*, 2024a. URL https://arxiv.org/abs/2412.20504.

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. AdaReTaKe: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://aclanthology.org/2025.findings-acl.283/.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. VideoLLM knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. *arXiv preprint arXiv:2411.17991*, 2024b. URL https://arxiv.org/abs/2411.17991.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.15754.

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

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. URL https://arxiv.org/abs/2605.22269.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. FluxMem: Adaptive hierarchical memory for streaming video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.02096.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv preprint arXiv:2501.13468*, 2025. URL https://arxiv.org/abs/2501.13468.

Haolin Yang, Feilong Tang, Lingxiao Zhao, Xinlin Zhuang, Yifan Lu, Xiang An, Ming Hu, Xiaofeng Zhang, Abdalla Swikir, Junjun He, Zongyuan Ge, Muhammad Haris Khan, and Imran Razzak. StreamAgent: Towards anticipatory agents for streaming video understanding. *arXiv preprint arXiv:2508.01875*, 2025. URL https://arxiv.org/abs/2508.01875.

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*, 2025. URL https://arxiv.org/abs/2509.24871.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. HERMES: KV cache as hierarchical memory for efficient streaming video understanding. *arXiv preprint arXiv:2601.14724*, 2026. URL https://arxiv.org/abs/2601.14724.

A Preliminary Analysis: Protocols and Additional Results

A.1 Data and Setup

Both models use the same 32-video sample, balanced equally between StreamingBench and Video-MME. Videos of at least 90 seconds are selected by a fixed seeded hash ordering of source and relative path. Each contributes its first 90 seconds at 1 FPS. Frame groups are encoded individually and comprise one frame and its corresponding visual KVs. The image processor uses a pixel budget of 357,376. All projected visual tokens and their causal KVs are retained. The study uses raw videos and generic prompts independently of benchmark questions and annotations.

A.2 Visual Change and Retention

For normalized mean projected visual features z_t, adjacent change is $c_t = 1 - \langle z_t, z_{t-1} \rangle$. The top-20% share sums the largest 18 changes among the 89 frames with a predecessor and divides by the total change within each video. This share is 66.5% for 4B and 65.0% for 8B.

Retention candidates are indexed $t \in \{30, \ldots, 59\}$ in the 90-frame sequence, starting from zero. For each candidate, let P_t contain its preceding 30 frames and $F_t(h)$ its next h frames. Its arrival-time and future-informed visual non-redundancy scores are

$$\nu_t^W = 1 - \max_{s \in P_t} \langle z_t, z_s \rangle, \qquad \nu_t^R(h) = 1 - \max_{s \in P_t \cup F_t(h)} \langle z_t, z_s \rangle. \tag{8}$$

A high score means that even the most similar context frame differs from the candidate. The top-n sets under the two scores are S_W and $S_R(h)$. At the primary setting, $n = 6$ and $h = 30$. Selected-frame replacement and score loss without updating are

$$T(h) = 1 - \frac{|S_W \cap S_R(h)|}{n}, \qquad D(h) = 100 \left(1 - \frac{\sum_{t \in S_W} \nu_t^R(h)}{\sum_{t \in S_R(h)} \nu_t^R(h)} \right). \tag{9}$$

Figure 2(b) shows $100T$ and D. Both selections are scored against the same expanded context; the ratio is computed within each video before averaging. For the cross-video control, future frames come from a different video in the same source at the corresponding timestamps, while the target candidates and their past contexts remain unchanged.

Figure S2 varies the future horizon over 5, 15, and 30 seconds, the temporal exclusion gap over 0, 1, 2, and 5 seconds, and the retention fraction over 10%, 20%, and 30%. An exclusion gap g removes the nearest g frames on each side of a candidate from its comparison context. The 4B control replaces 0.5% of selected frames in the main comparison and 0.0% at the corresponding sensitivity setting, which uses a separate control measurement. Score measurements are unchanged between the two figures.

A.3 Behavioral Influence and Cross-Fitted Selection

The six question categories concern people and objects; actions and interactions; text, numbers, and symbols; spatial relations; visual attributes; and changes and events. Each has three fixed paraphrases. For every prompt q, the model deterministically generates a full-history response prefix y_q^{full} of up to 16 tokens. With e_t denoting the visual KVs of frame t, frame influence is

$$I_q(t) = \text{NLL}(y_q^{\text{full}} \mid q, H_T \setminus e_t) - \text{NLL}(y_q^{\text{full}} \mid q, H_T). \tag{10}$$

For response tokens $y_1, \ldots, y_m$, the NLL is $-m^{-1} \sum_{j=1}^{m} \log p(y_j \mid y_{<j}, q, S)$, using natural logarithms and teacher forcing. It averages over response tokens and measures fidelity to this fixed response prefix.

For held-out slot p, same-category selection averages the other two paraphrases in that question's category. General-purpose selection pools the 12 remaining prompts across all six categories:

$$I_{-p}^{\text{pool}}(t) = \frac{1}{12} \sum_{k=1}^{6} \sum_{r \neq p} I_{k,r}(t). \tag{11}$$

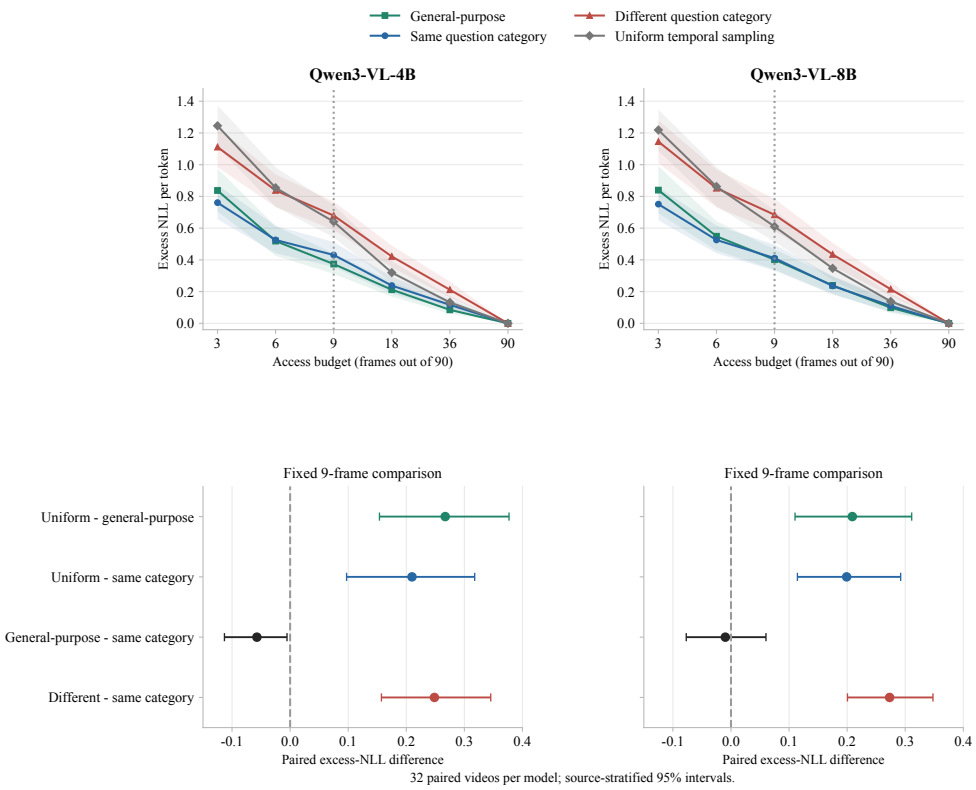


Figure S1: Evidence composition and exposed capacity both affect support. Top: excess mean token NLL at all six tested budgets; the vertical dotted line marks nine frames. Bottom: nine-frame comparisons. Budgets are equally spaced discrete categories; connecting lines guide the eye.

It selects the top-n frame set and uses that same set for all six held-out categories in the fold. Selection and evaluation use disjoint paraphrases of the same categories. Different-category evaluation averages the five nonmatching category selections for each held-out prompt. Uniform sampling selects frame $\lfloor (j + 0.5)90/n \rfloor$, for $j = 0, \ldots, n - 1$.

The intervention masks unselected visual KVs at answer time while retaining the full cache, all 180 nonvisual frame controls, and the original positions. Hidden states and cached values were computed from the full causal history. This isolates answer-time access to contextualized evidence. Each video contributes 18 held-out prompt measurements per condition and budget.

A.4 Read Comparisons and Capacity Sensitivity

The access budgets are 3, 6, 9, 18, 36, and 90 frame groups.

Let $L(q, S) = \mathrm{NLL}(y_q^{\mathrm{full}} \mid q, S)$ be mean token NLL. Excess NLL is $L(q, S) - L(q, H_T)$. Figure 2(c) plots $L(q, S_{\mathrm{uniform}}) - L(q, S)$; positive values favor selection over uniform sampling. The full-history baseline cancels in this difference and the category-mismatch gap.

Across all five sub-full budgets and both models, general-purpose and same-category selection have lower mean loss than uniform sampling, and same-category selection has lower mean loss than different-category selection. The relative ordering of general-purpose and same-category selection varies with capacity: same-category selection is better at three frames, while pooling is competitive at nine frames. Every adjacent capacity increase reduces excess NLL for every condition and both models. At 36 frames, pooled loss is 0.0850/0.0985 for 4B/8B; full exposure at 90 frames gives zero excess loss by definition. These measurements quantify the joint influence of evidence selection and capacity on behavioral fidelity.

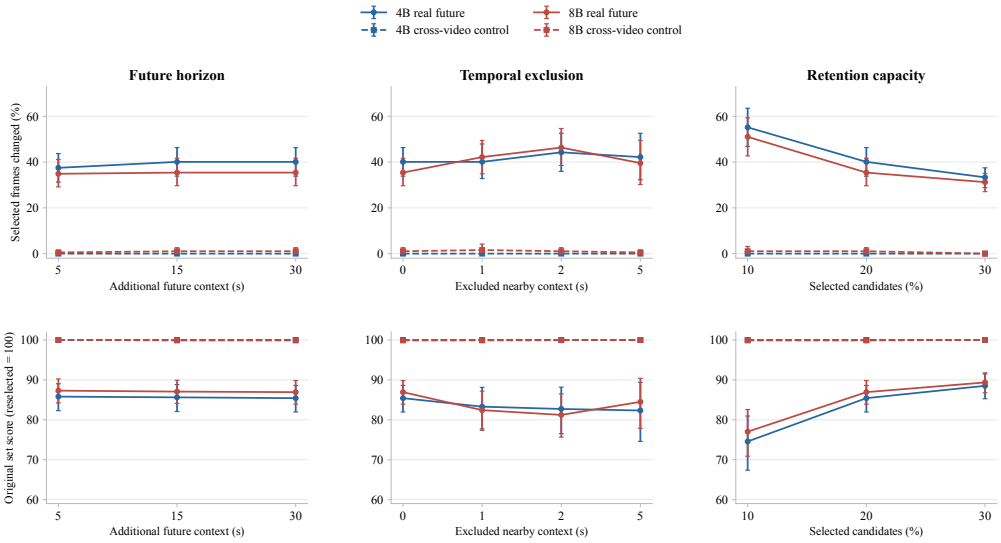


Figure S2: Earlier selections change with later context across horizons and budgets. Columns vary one factor, holding others at a 30-second horizon, zero exclusion gap, and 20% retention. Top: selected-frame replacement. Bottom: the original selection's non-redundancy score relative to reselection at 100, equal to $100 - D$. Solid curves use same-video future context; dashed curves use source-matched other-video context.

B StreamingReTaKe: Additional Details

B.1 Visual Novelty, Reference Refresh, and Budget Accounting

Let $X_t = \{x_{t,i}\}_{i=1}^{N}$ be the projected visual tokens of frame t and $\mathcal{P}_t = \{p_j\}$ the reference before frame t. For $t > 1$, we measure adjacent-frame change, reference departure, and token novelty:

$$\begin{aligned} d_t^{\text{adj}} &= 1 - \cos(\text{mean}(X_t), \text{mean}(X_{t-1}))\,, \\ d_t^{\text{ref}} &= 1 - \cos(\text{mean}(X_t), \text{mean}(\mathcal{P}_t))\,, \\ d_t^{\text{tok}} &= \text{TopMean}_i \left[1 - \max_j \cos(x_{t,i}, p_j)\right]. \end{aligned} \tag{12}$$

TopMean averages values at or above their median. Each signal is calibrated with a causal histogram CDF, and the combined score s_t is mapped to novelty quantile u_t:

$$s_t = \sum_{c\in\{\text{adj,ref,tok}\}} \widehat{F}_{c,t-1}(d_t^c), \qquad u_t = \widehat{F}_{s,t-1}(s_t). \tag{13}$$

Histograms update after frame t; allocation uses the current observation and prior statistics.

Budget bank and allocation. Let C_{t-1} be the credit available before frame t and $\widehat{b}_t$ its requested token count. The bank caps the request by the base allowance B plus available credit, then saves the unspent balance up to $C_{\max}$:

$$b_t = \min(\widehat{b}_t, B + C_{t-1}), \qquad C_t = \min(C_{\max}, C_{t-1} + B - b_t). \tag{14}$$

We initialize $C_0 = C_{\max} = N - B$ to fund the first full frame. Requests $\widehat{b}_t$ follow these rules.

For $t > 1$, define the novelty-record indicator over available preceding scores as

$$r_t = \mathbf{1}\left[s_t > \max_{\max(1,t-W)\le j<t} s_j\right]. \tag{15}$$

The first frame receives a full allocation. Subsequently, the full-anchor indicator is

$$\alpha_t = \mathbf{1}[r_t = 1 \ \wedge \ C_{t-1} = C_{\max}], \qquad t > 1, \quad \alpha_1 = 1. \tag{16}$$

With $f = \max(1, \lfloor B/2 \rfloor)$ and $v = B - f$, the requested visual allocation is

$$\widehat{b}_t = \begin{cases} N, & \alpha_t = 1, \\ \min(N, f + \lfloor 2vu_t \rfloor), & \alpha_t = 0. \end{cases} \tag{17}$$

Algorithm 1 applies this request and bank cap with mutable credit C; the full update appears in Algorithm 2. An unfunded record closes the preceding episode and refreshes the working reference with its adaptively selected tokens. After W frames, the next frame starts a new episode and refreshes the reference. Only the first frame and fully funded records receive anchor protection.

A reference refresh initializes $\mathcal{P}_{t+1}$ from the current selected visual tokens and saves the decoder KVs produced by those same tokens. Between refreshes, the more redundant half of dropped tokens update their nearest visual prototypes through a nonnegative similarity-weighted running mean; the saved reference KVs remain fixed. These updates operate in projected visual space, using the dropped tokens' pre-decoder features. The KV reference supplies the residuals in Equation 23 until the next refresh. With value-only residuals, $D_{t,i}^{\ell}$ uses only the value term.

The allocation in Equation 14 is equivalent to

$$b_t = \begin{cases} \widehat{b}_t, & \widehat{b}_t \le B, \\ B + \min(C_{t-1}, \widehat{b}_t - B), & \widehat{b}_t > B. \end{cases} \tag{18}$$

Since $b_t \le B + C_{t-1}$ and $C_0 = N - B \ge 0$, credit remains nonnegative. The bank update also gives $C_t \le C_{t-1} + B - b_t$. Summing over $t = 1, \ldots, T$ yields

$$\sum_{t=1}^{T} b_t \le TB + C_0 - C_T \le TB + C_0. \tag{19}$$

Consequently, $\limsup_{T\to\infty} T^{-1}\sum_{t=1}^{T} b_t \le B$. Initial credit covers the first full anchor.

Rolling-window memory guarantees. The same recurrence constrains any contiguous interval of admitted frames. Since $0 \le C_s \le C_{\max} = N - B$ and $b_s \le B + C_{s-1} - C_s$, summing over $s = a, \dots, t$ gives

$$\sum_{s=a}^{t} b_s \le (t-a+1)B + C_{a-1} - C_t \le (t-a+1)B + N - B. \tag{20}$$

For $w_t = \min(t, W)$, recent memory holds these last w_t frames at their write-stage resolution, together with two vision controls per frame. Therefore,

$$\begin{aligned} |\mathcal{M}_t^{\text{recent}}| &= \sum_{s=t-w_t+1}^{t} (b_s + 2) \\ &\le w_t(B+2) + C_{t-w_t} - C_t \\ &\le w_t(B+2) + N - B \le W(B+2) + N - B. \end{aligned} \tag{21}$$

The window can include the first full anchor: its initial credit is already covered by C_0. Combining the recent bound with long-memory capacity yields

$$|\mathcal{M}_t| \le M_{\text{KV}} + W(B+2) + N - B. \tag{22}$$

An answer branch copies the current bank credit and allocation state. Any question-time frame is admitted through that bank, so its branch history satisfies Equation 20 as well. With C_R recent KVs, control-aware retrieval adds at most $\max(0, M_{\text{KV}} - C_R)$ long-term KVs, establishing Equation 7. The same writing budget therefore bounds admission, persistent memory, and the answer context.

B.2 Token Scoring and Ranking

In Equation 3, δ^{ref} is cosine distance to the nearest reference prototype, and δ^{mot} is cosine distance to the best match in the corresponding 3×3 neighborhood of the preceding frame. Visual importance is computed over all N tokens; the top b_t are restored to source order before language-model inference.

The selected tokens are processed with the causal cache, their vision-start/end controls, and a fixed query-agnostic probe. Probe attention supplies $A_{t,i}^{\ell}$; probe tokens are discarded after scoring. For a non-anchor frame, let $\rho_t < t$ be the preceding reference-refresh frame. Its saved visual KVs, indexed by $j \in \mathcal{S}_{\rho_t}$, define

$$D_{t,i}^{\ell} = \tfrac{1}{2}\left(\min_j[1 - \cos(\widetilde{K}_{t,i}^{\ell}, \widetilde{K}_{\rho_t,j}^{\ell})] + \min_j[1 - \cos(V_{t,i}^{\ell}, V_{\rho_t,j}^{\ell})]\right). \tag{23}$$

Keys $\widetilde{K}$ are taken before rotary position encoding. Full anchors receive a unit residual.

For a nonempty visual-index set $\mathcal{A}$ of size m, let $n_<(i) = |\{j \in \mathcal{A} : v_j < v_i\}|$ and $n_=(i) = |\{j \in \mathcal{A} : v_j = v_i\}|$. The percentile operator is

$$R_{\mathcal{A}}(v_i) = \begin{cases} \dfrac{n_<(i) + \frac{1}{2}(n_=(i) - 1)}{m - 1}, & m > 1, \\ \frac{1}{2}, & m = 1. \end{cases} \tag{24}$$

Thus, tied values receive their average zero-based rank divided by $m-1$; a constant vector receives 0.5 throughout. Top-k selection breaks score ties by source order, using chronological frame order before within-frame token order when candidates span frames.

Equation 3 ranks each component over all N visual tokens, including on full-anchor frames. Equation 4 ranks each layer's attention and residual over the b_t written visual tokens, and re-ranks their stored $I_{t,i}$ values over that same set. Vision controls and probe tokens are excluded from these ranking domains. On the first frame, the absent reference and preceding frame give $\delta_{1,i}^{\text{ref}} = \delta_{1,i}^{\text{mot}} = 1$, so $I_{1,i} = 1 + R_{\mathcal{U}_1}(\|x_{1,i}\|_2)$. Full anchors also have $D_{t,i}^{\ell} = 1$, contributing 0.5 after ranking. The same tie rule applies to the resident long-term visual candidates in Equation 5.

B.3 Episode Allocation

For a closed episode E of T_E frames, let $Q_E = \lfloor KT_E \rfloor$ and A_E the full-anchor visual-plus-control cost. For n_E non-anchor frames with two controls each, the visual quota is

$$Q_E^{\text{visual}} = \max(0, Q_E - A_E - 2n_E). \tag{25}$$

The highest-G non-anchor visual tokens share this allowance across the episode, and each anchor receives its complete KVs, including when anchor cost exceeds the target Q_E. Allocation is determined at closure; the recent window continues to retain each frame's admitted visual KVs and controls until migration. The W-frame episode limit ensures that allocation is determined before its first frame leaves the recent window. Global eviction subsequently enforces the hard long-memory cap, including visual and control KVs.

At capacity pressure, non-anchor visual tokens are evicted globally in ascending order of

$$E_{t,i} = G_{t,i} + \lambda u_t, \qquad \lambda = 0.5, \tag{26}$$

before considering anchors. Long memory counts visual and control KVs. A frame's controls are removed with its last visual KV, maintaining the hard capacity M_{KV} after migration.

B.4 Indexed Retrieval

Retrieval uses the question as a query to score resident long-term visual keys. With $q^{\ell}_{r,h}$ denoting the query for question token r and head h, the average compatibility is

$$a_i^{\ell}(q) = \frac{1}{HT_q} \sum_{h=1}^{H} \sum_{r=1}^{T_q} \text{softmax}_j \left(\frac{\langle q^{\ell}_{r,h}, \widetilde{k}^{\ell}_{j,g(h)} \rangle}{\sqrt{d}} \right)_i . \tag{27}$$

Here, T_q, H, and d are prompt length, query-head count, and head dimension, and $g(h)$ maps a query head to its grouped key head. Scoring in pre-RoPE space measures compatibility without position-dependent rotary phase. The index ℓ ranges over the scoring layers $\mathcal{L}_q$. Percentile ranks over resident long-term visual candidates give $J_i(q)$ and $S_i(q)$ in Equation 5.

With C_R recent visual and control KVs, long-term selection receives $A_q = \max(0, M_{\text{KV}} - C_R)$. Candidates are traversed in descending $S_i(q)$, charging one visual KV and, for the first selected token of a frame, its two controls. Candidates that do not fit are skipped. Selected tokens are restored to chronological frame and source order, followed by the complete recent window. If the recent window exceeds the target, it supplies the answer context and no long-term tokens are added. The cache is ephemeral: retrieval preserves persistent importance and both memory components.

B.5 Chronological Position Reconstruction

Pruning produces an irregular sequence of surviving frames and spatial locations. Each retained KV stores its source-frame identifier, three-axis offset $o_{i,a}$, and frame span σ_f. For surviving frames $f_1, \ldots, f_m$ in chronological order, compact positions are

$$\beta_1 = 0, \qquad \beta_r = \sum_{j<r} \sigma_{f_j}, \qquad \pi_{i,a} = \beta_r + o_{i,a}, \quad i \in f_r,\ a \in \{t, h, w\}. \tag{28}$$

The backend stores pre-RoPE keys and reapplies rotary encoding at $\pi_{i,a}$ when rebuilding a selected cache. This compacts surviving frames chronologically while preserving their within-frame temporal and spatial offsets. The same reconstruction serves persistent-cache assembly and ephemeral question-time reads; cached values retain their causally computed representations.

B.6 Frame Updates and Question-Time Branching

Algorithms 2 and 3 specify the causal update and question-time read. A state Z contains the bank, novelty histograms and record window, episode metadata, recent and long memory, both reference representations, the preceding frame, and frame/position counters. A fork copies mutable state and shares immutable tensor payloads. Each question at a shared boundary forks the same preceding causal state; its answer state is discarded after use.

In the algorithms, a frame record F_t contains selected indices $\mathcal{S}_t$, visual and control KVs $\mathcal{K}_t$, and retention metadata. E is the open episode, and $\mathcal{R}$ is the recent-frame queue. The initial state has $C = N - B$, empty memories and statistics, no reference, and $t = 1$. Symbols $C, t, E, \mathcal{R}, \mathcal{M}^{\text{long}}$ denote mutable fields of the supplied state Z. WriteFrame performs visual-token selection and prefill, returns the frame record, and advances the position counter; probe KVs are excluded. CloseEpisode applies Appendix B.3 and leaves an empty episode unchanged; Evict enforces the long-memory cap. Admit selects KVs in source order and charges controls (Section 4.3).

Algorithm 2 Causal evidence update

Require: State Z, frame X_t, optional question q
Ensure: Updated state Z

function UpdateFrame(Z, $X_t, q = \varnothing$)
  $(s_t, u_t) \leftarrow \text{Novelty}(X_t, \mathsf{Z})$ ▷ Past statistics; Eq. 13
  $r_t \leftarrow \mathbf{1}[t > 1 \wedge s_t > \max(\mathsf{Z}.\text{records})]$
  $\alpha_t \leftarrow \mathbf{1}[t = 1 \vee (r_t = 1 \wedge C = N - B)]$
  $\eta_t \leftarrow \mathbf{1}[t = 1 \vee r_t = 1 \vee |E| = W]$ ▷ Refresh decision
  $\mathsf{Z}.\text{records} \leftarrow \text{LastW}(\mathsf{Z}.\text{records} \parallel s_t)$
  $b_t \leftarrow \min(\widehat{b}_t, B + C)$ ▷ $\widehat{b}_t$ from Eq. 17
  $C \leftarrow \min(N - B, C + B - b_t)$
  $F_t \leftarrow \text{WriteFrame}(X_t, b_t, \mathsf{Z}, q, u_t, \alpha_t)$
  $(\mathcal{S}_t, \mathcal{K}_t) \leftarrow (F_t.\text{indices}, F_t.\text{KVs})$
  if $\eta_t = 1$ **then**
    CloseEpisode(E); $E \leftarrow \varnothing$
  $E \leftarrow E \parallel F_t$; $\mathcal{R} \leftarrow \mathcal{R} \parallel F_t$
  while $|\mathcal{R}| > W$ **do**
    $F \leftarrow \text{PopOldest}(\mathcal{R})$
    $\mathcal{M}^{\text{long}} \leftarrow \mathcal{M}^{\text{long}} \parallel \text{ConsolidatedKVs}(F)$
    $\mathcal{M}^{\text{long}} \leftarrow \text{Evict}(\mathcal{M}^{\text{long}}, M_{\text{KV}})$
  if $\eta_t = 1$ **then**
    $(\mathsf{Z}.\mathcal{P}, \mathsf{Z}.\text{KVref}) \leftarrow (X_t[\mathcal{S}_t], \mathcal{K}_t^{\text{visual}})$
  else
    $\mathsf{Z}.\mathcal{P} \leftarrow \text{UpdatePrototypes}(\mathsf{Z}.\mathcal{P}, X_t, \mathcal{S}_t)$
  UpdateStatistics(Z, X_t, s_t) ▷ Novelty histograms
  $\mathsf{Z}.\text{previous} \leftarrow X_t$; $t \leftarrow t + 1$
  return Z

Algorithm 3 Question-time retrieval and stream continuation

Require: Pre-frame state Z, questions $\mathcal{Q}$, optional frame X_t
Require: a: whether X_t is a scheduled frame and the stream continues
Ensure: Answers $\mathcal{Y}$, query-agnostic state Z

$\mathcal{Y} \leftarrow \varnothing$
for all $q \in \mathcal{Q}$ **do**
  $\mathsf{Z}^q \leftarrow \textsc{Fork}(\mathsf{Z})$ ▷ Copy credit, references, and memory state
  if $X_t \neq \varnothing$ **then**
    $\mathsf{Z}^q \leftarrow \textsc{UpdateFrame}(\mathsf{Z}^q, X_t, q)$
  $\mathcal{V} \leftarrow \textsc{LongVisualKVs}(\mathsf{Z}^q)$
  $J(q) \leftarrow \textsc{IndexedAttention}(q, \mathcal{V})$ ▷ Eq. 5
  $S(q) \leftarrow R(G_{\mathcal{V}}) + R(J(q))$
  $C_R \leftarrow |\textsc{RecentKVs}(\mathsf{Z}^q)|; \quad A_q \leftarrow \max(0, M_{\mathrm{KV}} - C_R)$
  $\mathcal{C}_q \leftarrow \textsc{Admit}(\mathcal{V}, S(q), A_q) \parallel \textsc{RecentKVs}(\mathsf{Z}^q)$
  $\mathcal{C}_q \leftarrow \textsc{ReconstructPositions}(\mathcal{C}_q)$
  $\mathcal{Y}[q] \leftarrow \textsc{Generate}(q, \mathcal{C}_q); \quad \textsc{Discard}(\mathsf{Z}^q)$
if a **then**
  $\mathsf{Z} \leftarrow \textsc{UpdateFrame}(\mathsf{Z}, X_t, \varnothing)$
return $(\mathcal{Y}, \mathsf{Z})$

C TASK-LEVEL STREAMING RESULTS

We use the best reported HERMES results.

Table S1: **OVO-Bench task-level results.** Accuracy (%) for Real-Time Visual Perception and Backward Tracing. OCR: optical character recognition; ACR: action recognition; ATR: attribute recognition; STU: spatial understanding; FPD: future prediction; OJR: object recognition; EPM: episodic memory; ASI: action sequence identification; HLD: hallucination detection.

			Real-Time Visual Perception							Backward Tracing			
Method	**Size**	**Frames**	**OCR**	**ACR**	**ATR**	**STU**	**FPD**	**OJR**	**Avg.**	**EPM**	**ASI**	**HLD**	**Avg.**
Proprietary Models													
Gemini 1.5 Pro	–	1 fps	85.90	67.00	79.30	58.40	63.40	62.00	69.30	58.59	76.35	52.64	62.54
GPT-4o	–	64	69.80	64.20	71.60	51.10	70.30	59.80	64.50	57.91	75.68	48.66	60.75
Open-source Offline MLLMs													
LongVU	7B	1 fps	55.70	49.50	59.50	48.30	68.30	63.00	57.40	40.74	59.46	4.84	35.01
LLaVA-Video	7B	64	69.10	58.70	68.80	49.40	74.30	59.80	63.50	56.23	57.43	7.53	40.40
InternVL2	8B	64/16	67.10	60.60	63.80	46.10	68.30	56.50	60.40	48.15	57.43	24.73	43.44
Open-source Online MLLMs (Training-Based)													
VideoLLM-online	8B	2 fps	8.10	23.90	12.10	14.00	45.50	21.20	20.80	22.22	18.80	12.18	17.73
Dispider	7B	1 fps	57.70	49.50	62.10	44.90	61.40	51.60	54.60	48.48	55.41	4.30	36.06
StreamForest	7B	1 fps	68.50	53.20	71.60	47.80	65.40	60.90	61.20	58.92	64.86	32.26	52.02
Streamo	3B	1 fps	78.52	52.29	67.24	44.38	55.45	71.20	61.51	51.18	57.43	16.67	41.76
Streamo	7B	1 fps	79.19	57.80	75.00	49.44	64.36	70.11	65.98	54.55	52.03	31.72	46.10
Open-source Online MLLMs (Training-Free)													
LLaVA-OV	7B	64	69.10	58.70	68.80	49.40	74.30	59.80	63.50	56.23	57.43	7.53	40.40
+ ReKV	7B	0.5 fps	52.35	54.13	69.83	43.26	67.33	57.07	57.33	57.58	56.08	18.82	44.16
+ StreamingTOM	7B	0.5 fps	67.80	61.50	**75.90**	47.80	68.30	**66.30**	63.60	36.70	41.89	**56.45**	45.01
+ HERMES	7B	0.5 fps	**72.48**	**62.39**	74.14	50.56	**73.27**	65.22	66.34	**60.61**	**61.49**	28.49	**50.20**
+ StreamingReTaKe	7B	0.5 fps	71.81	**62.39**	72.41	**54.49**	**73.27**	**66.30**	**66.78**	57.90	58.10	29.00	48.30
Qwen2.5-VL	7B	1 fps	79.20	53.20	67.20	51.70	71.30	57.10	63.30	47.14	60.81	17.74	41.90
+ HERMES	7B	1 fps	**85.23**	64.22	71.55	53.37	74.26	65.22	68.98	48.48	62.16	**37.63**	49.43
+ FluxMem	7B	1 fps	81.20	59.60	70.70	53.40	**75.20**	63.00	67.20	48.48	**64.19**	29.03	47.24
+ StreamingReTaKe	7B	1 fps	80.54	**70.64**	**75.86**	**53.93**	72.28	**72.83**	**71.01**	**54.55**	**64.19**	34.80	**51.18**
Qwen3-VL	4B	1 fps	80.54	60.55	72.41	52.25	76.24	67.39	68.23	59.90	60.81	13.44	44.72
+ StreamingReTaKe	4B	1 fps	80.54	67.89	71.55	56.18	74.26	72.28	70.45	61.95	60.81	35.48	52.75
+ HERMES	4B	2 fps	81.88	**72.48**	**78.45**	59.55	**75.25**	66.30	72.32	55.89	63.51	**45.70**	55.03
+ StreamingReTaKe	4B	2 fps	**89.93**	**72.48**	75.86	**61.80**	74.26	**78.80**	**75.52**	**62.29**	**67.57**	35.48	**55.11**
Qwen3-VL	8B	1 fps	82.55	64.22	73.28	57.87	71.29	64.13	68.89	55.56	66.89	11.83	44.76
+ StreamingReTaKe	8B	1 fps	82.55	68.81	70.69	56.18	70.30	66.30	69.14	56.57	75.00	27.96	53.18
+ HERMES	8B	2 fps	**87.92**	**74.31**	**79.31**	56.18	70.30	71.74	73.29	55.56	67.57	24.73	49.29
+ StreamingReTaKe	8B	2 fps	87.25	68.81	76.72	**62.36**	**74.26**	**78.26**	**74.61**	**56.57**	**70.27**	**30.11**	**52.32**

Table S2: **StreamingBench Real-Time Visual Understanding (a).** Per-task accuracy (%); task abbreviations follow StreamingBench. The two tables share the same methods and frame settings.

Method	Size	Frames	OP	CR	CS	ATP	EU
Proprietary Models							
Gemini 1.5 Pro	–	1 fps	79.00	80.50	83.50	79.70	80.00
GPT-4o	–	64	77.10	80.50	83.90	76.50	70.20
Open-source Offline MLLMs							
LongVU	7B	1 fps	–	–	–	–	–
LLaVA-Video	7B	64	–	–	–	–	–
InternVL2	8B	64/16	68.10	60.90	69.40	77.10	67.70
Open-source Online MLLMs (Training-Based)							
VideoLLM-online	8B	2 fps	39.10	40.10	34.50	31.10	46.00
Dispider	7B	1 fps	74.90	75.50	74.10	73.10	74.40
StreamForest	7B	1 fps	83.10	82.80	82.70	84.30	77.50
Streamo	3B	1 fps	–	–	–	–	–
Streamo	7B	1 fps	–	–	–	–	–
Open-source Online MLLMs (Training-Free)							
LLaVA-OV	7B	64	78.75	78.12	80.76	81.19	71.70
+ ReKV	7B	0.5 fps	74.40	78.90	78.60	77.10	68.30
+ StreamingTOM	7B	0.5 fps	79.20	80.10	78.20	79.40	67.70
+ HERMES	7B	0.5 fps	79.02	**81.25**	**87.70**	80.20	69.18
+ StreamingReTaKe	7B	0.5 fps	**81.20**	79.69	82.02	**83.17**	**72.96**
Qwen2.5-VL	7B	1 fps	78.30	80.50	79.80	82.40	75.50
+ HERMES	7B	1 fps	**83.65**	**81.25**	**88.01**	**87.46**	76.73
+ FluxMem	7B	1 fps	80.20	81.10	81.40	85.30	78.00
+ StreamingReTaKe	7B	1 fps	83.20	74.22	87.07	83.65	**78.62**
Qwen3-VL	4B	1 fps	78.32	75.78	80.76	78.53	76.10
+ StreamingReTaKe	4B	1 fps	83.47	78.12	88.33	84.29	78.62
+ HERMES	4B	2 fps	79.56	78.12	88.33	**87.79**	79.25
+ StreamingReTaKe	4B	2 fps	**86.10**	**80.47**	**92.74**	87.13	**81.13**
Qwen3-VL	8B	1 fps	79.67	75.78	82.65	81.73	74.21
+ StreamingReTaKe	8B	1 fps	82.66	77.34	89.59	85.58	76.10
+ HERMES	8B	2 fps	84.47	78.91	90.54	**88.78**	78.62
+ StreamingReTaKe	8B	2 fps	**87.47**	**81.25**	**93.38**	86.80	**83.02**

Table S3: **StreamingBench Real-Time Visual Understanding (b).** Per-task accuracy (%); task abbreviations follow StreamingBench. The two tables share the same methods and frame settings.

Method	Size	Frames	TR	PR	SU	ACP	CT	Avg.
Proprietary Models								
Gemini 1.5 Pro	–	1 fps	84.70	77.80	64.20	72.00	48.70	75.70
GPT-4o	–	64	83.80	66.70	62.20	69.10	49.20	73.30
Open-source Offline MLLMs								
LongVU	7B	1 fps	–	–	–	–	–	–
LLaVA-Video	7B	64	–	–	–	–	–	–
InternVL2	8B	64/16	62.90	59.30	53.30	55.00	56.50	63.70
Open-source Online MLLMs (Training-Based)								
VideoLLM-online	8B	2 fps	32.40	31.50	34.20	42.50	27.90	36.00
Dispider	7B	1 fps	59.90	76.10	62.90	62.20	45.80	67.60
StreamForest	7B	1 fps	78.20	76.90	69.10	75.60	54.40	77.30
Streamo	3B	1 fps	–	–	–	–	–	–
Streamo	7B	1 fps	–	–	–	–	–	–
Open-source Online MLLMs (Training-Free)								
LLaVA-OV	7B	64	72.59	72.22	63.82	66.01	38.34	71.34
+ ReKV	7B	0.5 fps	67.90	67.60	62.60	64.30	44.60	69.10
+ StreamingTOM	7B	0.5 fps	68.90	69.40	60.00	64.60	**46.30**	70.10
+ HERMES	7B	0.5 fps	71.96	73.15	66.26	69.41	43.52	73.23
+ StreamingReTaKe	7B	0.5 fps	**74.77**	**74.07**	**67.07**	**71.39**	41.97	**73.99**
Qwen2.5-VL	7B	1 fps	80.40	74.10	62.60	67.60	51.10	73.90
+ HERMES	7B	1 fps	**86.60**	82.41	76.02	**73.94**	46.63	**79.44**
+ FluxMem	7B	1 fps	83.80	80.60	65.90	69.60	**52.10**	76.40
+ StreamingReTaKe	7B	1 fps	82.55	**83.33**	**78.05**	70.25	44.04	77.55
Qwen3-VL	4B	1 fps	75.08	80.56	63.01	67.05	55.32	73.24
+ StreamingReTaKe	4B	1 fps	81.62	81.48	71.14	70.74	50.53	77.80
+ HERMES	4B	2 fps	85.98	**85.19**	72.36	74.22	43.52	78.40
+ StreamingReTaKe	4B	2 fps	**88.47**	**85.19**	**73.98**	**75.92**	**52.85**	**81.52**
Qwen3-VL	8B	1 fps	77.88	81.48	61.79	65.34	56.91	74.12
+ StreamingReTaKe	8B	1 fps	80.69	80.56	68.70	68.18	56.38	77.48
+ HERMES	8B	2 fps	**88.79**	**87.96**	**74.39**	**79.32**	48.70	81.32
+ StreamingReTaKe	8B	2 fps	87.23	82.41	72.76	75.92	**56.48**	**81.80**

D Ablation Settings and Scoring

All rows in Tables 3 and 4 use the same 2,495-question StreamingBench Real-Time set and report question-level accuracy. The reference configuration uses Qwen3-VL-8B at 2 FPS, $B = 72$, $K = 24$, $W = 32$, and a $M_{\mathrm{KV}} = 4096$. Query scoring uses the deepest 30% of layers. Input settings are fixed at 357,376 maximum pixels for model input and a 552,960-pixel decode limit. Each variant changes only the indicated component or parameter.

Fixed per-frame budgeting replaces novelty-adaptive allocation with a 72-token allowance. Uniform token selection replaces importance-based selection within each frame while preserving the adaptive writing-budget policy.

Arrival-locked retention assigns immutable long-term token identities at arrival using the same scoring rule as Full. At episode prefix length r, let $A_E(r)$ denote the anchor visual-plus-control cost and $n_E(r)$ the number of non-anchor frames. The available non-anchor visual quota is $V_E(r) = \max(0, \lfloor Kr \rfloor - A_E(r) - 2n_E(r))$. Each non-anchor arrival receives $V_E(r) - V_E(r-1)$ slots for its highest-scoring tokens; complete anchors retain their protection. The total assigned visual quota at closure matches joint selection. Both variants preserve all admitted KVs in the same recent window and migrate them at the same window-exit time, with the same global eviction and Read rules. Different selected frame support can give different actual control-token occupancy under the common capacity bound. FIFO changes global overflow eviction to arrival order, preserving episode-level selection and preferential anchor retention.

Importance-only and question-only reading use $R(G_i)$ and $R(J_i(q))$, respectively, in place of their sum, retaining the same answer-cache target. Full-cache reading exposes all resident long and recent KVs; its answer context therefore follows resident-memory size rather than the 4,096-KV target used by indexed retrieval. Query-depth sensitivity changes the selected scoring layers from the deepest 30% to the deepest 10%.

The reference answers 2,041 questions correctly. FIFO reports 1,990 correct answers after one answer-format correction; its original output gives 1,989/2,495 (79.72%). The $W = 64$ result is 1,726/2,495 (69.18%), with seven invalid answers counted as incorrect. The $M_{\mathrm{KV}} = 2048$ accuracy is author-reported as 70.02%, with $\Delta = -11.78$ points relative to the reported 81.80% reference. Other Δ values in Table 4 use unrounded accuracies before rounding to two decimals.

D.1 Recent-Window Occupancy and Long-Term Access

The resource variants change the capacity available for historical evidence in an answer. Let $w \le W$ be the number of recent frames after any question-time frame update, and let $\bar{b}_R$ be their average allocated visual-token count. Including two control KVs per frame, recent occupancy and the remaining long-term retrieval allowance are

$$C_R = w(\bar{b}_R + 2), \qquad A_q = \max\{0, M_{\mathrm{KV}} - w(\bar{b}_R + 2)\}. \tag{29}$$

For a full window, $A_q = 0$ whenever $\bar{b}_R \ge M_{\mathrm{KV}}/W - 2$. At the nominal writing rate $\bar{b}_R = B = 72$, this gives

(W, M_{KV})	C_R	A_q	Zero-allowance threshold for $\bar{b}_R$
$(32, 4096)$	2368	1728	126
$(64, 4096)$	4736	0	62
$(32, 2048)$	2368	0	62

(30)

Thus, doubling W or halving M_{KV} lowers the threshold from 126 to 62 visual tokens per frame, below the nominal rate of 72. These configurations can leave no capacity for long-term evidence in an answer even when that evidence remains stored. Reducing M_{KV} also lowers the persistent long-memory limit, so more stored evidence may be evicted before a question arrives.

Actual $\bar{b}_R$ varies with adaptive allocation; the exact condition for exhausting long-term retrieval capacity is $C_R \ge M_{\mathrm{KV}}$. The complete recent window then supplies the answer context. Long-term KVs are omitted from that answer while remaining in persistent memory, so older evidence can become inaccessible to the current question.

E RELATIONSHIP TO RETAKE AND ADARETAKE

Name and shared motivation. The name StreamingRETAKE acknowledges the inspiration of RETAKE and AdaRETAKE (Wang et al., 2024a; 2025). In RETAKE, the name abbreviates *REducing TemporAl and Knowledge rEdundancy*. We retain this name to reflect the shared goal of reducing visual and KV redundancy, and add *Streaming* to identify the setting of continuously arriving video and questions. The shared methodological ideas are training-free compression, visual change as a selection signal, and attention-guided KV selection with protection for important visual evidence.

Prior compression designs. RETAKE combines DPSelect, which selects keyframes using inter-frame distance peaks, with PivotKV, which preserves pivot tokens and compresses other tokens during chunked prefill (Wang et al., 2024a). AdaRETAKE allocates compression ratios across video chunks and model layers, using temporal variation and prompt attention to retain informative KVs (Wang et al., 2025). Its temporal allocation distributes a context budget over the supplied video's chunks. These designs expand the amount of video that fits into the model context. Streaming additionally requires maintaining reusable history before future questions are known, with bounded storage and timely access throughout continued ingestion.

Causal allocation to incoming frames. StreamingRETAKE allocates visual tokens before prefill using only the observed stream. A budget bank transfers unused capacity from redundant frames to informative ones while bounding average token admission. This supports continuous processing without requiring the complete video length or a budget allocation over all future chunks. The allocation also selects within-frame tokens and maintains evolving visual prototypes to represent previous observations.

Joint retention under bounded storage. StreamingRETAKE keeps admitted KVs in a recent window and jointly selects long-term evidence from accumulated candidates under an episode quota. This gives earlier and later evidence a shared allocation opportunity. The selected KVs migrate into bounded long-term memory, where global eviction controls storage as the stream grows. All decoder layers share the retained token indices. These mechanisms coordinate admission, recent residency, and long-term retention, providing memory bounds independent of stream duration.

Shared history and question-conditioned access. StreamingRETAKE preserves shared evidence before future questions are known. When a question arrives, it serves as a query to retrieve a compact context using both persistent importance and question relevance. Retrieval operates on stored KVs, and answering leaves the persistent history unchanged, allowing subsequent questions to reuse that history while video ingestion continues.

F RESOURCE ACCOUNTING FOR THE LLAVA COMPARISON

We distinguish persistent video KVs, answer-time video KVs, and the physical GPU footprint, which includes weights, auxiliary state, temporary tensors, and allocator reservation. The LLaVA HERMES accuracy in Table 1 uses its reported 4K configuration; the controlled efficiency comparison in Section 5.3 uses its released 6000-token setting.

Accounting for recent memory. Our LLaVA reference uses $N = 196$, $B = 128$, $K = 32$, $W = 8$, and $M_{\mathrm{KV}} = 4894$. This backbone has no per-frame vision-boundary tokens. The budget bank bounds recent video KVs by $WB + N - B = 1092$, so long-term memory, recent memory, and the 14-token fixed prefix together occupy at most $4894 + 1092 + 14 = 6000$ decoder positions per layer. The answer exposes at most 4894 video KVs, including recent evidence. The 4894 setting therefore explicitly accounts for recent residency. The 73.99% StreamingBench result in Table 1 uses $K = 64, W = 6$: across all 2495 questions, resident/answer video KVs average 4367.89/3904.17 and reach maxima of 5730/4894.

Measured resource and response costs. Table S4 compares the same LLaVA-OneVision-7B weights in FP16 on one H100, with identical decoded frames at 0.5 FPS and 384×384 vision inputs. It covers the 37 videos and 185 question arrivals used in Section 5.3. Our $W = 8$ reference processes frames individually; HERMES uses 16-frame chunks. Each retains its prompt and software configuration. Our allocator uses `expandable_segments:True`; HERMES uses its default allocator.

Table S4: LLaVA resource and response measurements on 185 common questions. Video-KV counts exclude fixed prefixes and question/answer text. Latencies are medians; physical memory is the cohort maximum, measured by 10 ms NVML sampling and query-boundary readings.

Measurement	HERMES (6000)	StreamingRETAKE
Resident video KVs, median / maximum	6000 / 6000	4050 / 5983
Answer video KVs, median / maximum	6000 / 6000	4050 / 4894
Physical GPU peak (GiB)	24.80	22.06
Pending frames at query, median / maximum	6 / 15	0 / 0
Pending-frame processing (ms)	381.75	0.01
TTFT after pending-frame processing (ms)	72.10	148.42
TTFT from question arrival (ms)	454.83	148.43

HERMES answers from its retained history, reaching 6013 positions per layer including its 13-token prefix. Our maximum answer history is 18.4% smaller. HERMES's pending-frame cost includes vision encoding and compression; our timed answer path includes activation-frame prefill, retrieval, and first-token generation with vision features already available. Including these outstanding operations, our median TTFT is $3.06\times$ lower. A separate serialized write profile on the same videos gives 79.20 ms/frame for our one-frame processing and 32.32 ms/frame for HERMES's 16-frame batching; our source-time factor of 0.0394 leaves approximately $25\times$ headroom at 0.5 FPS.

Long-stream residency. On the shared 1024-frame video in Figure 1(e), our resident/answer video-KV counts are 5897/4894, versus HERMES's 6000/6000. Including recent memory and auxiliary state, physical peaks are 20.20 versus 23.14 GiB, a 12.7% reduction. From 512 to 1536 frames, our answer history stays at 4894 KVs and physical peak rises only from 20.18 to 20.32 GiB. These measurements connect bounded persistent storage and compact answer access to the complete implementation footprint.