AUTOSTREAM: AUTOMATIC CONTEXT EXPANSION FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Online video understanding requires models to perceive ongoing events while retaining relevant information from the past. Existing training-free methods, however, struggle to balance these two demands: the accumulated memory maintained by history-centric approaches may interfere with real-time perception, whereas recent-window approaches preserve real-time responsiveness at the cost of discarding long-term evidence. We present AutoStream, a training-free framework that realizes a *real-time perception first, history on demand* principle through *automatic context expansion*. During stream ingestion, Query-Ready Multi-Context Cache (QRMC) prepares hierarchical caches that span real-time perception and comprehensive historical context. Novelty-Weighted Reservoir Sampling (NWRS) retains temporally dispersed and informative historical evidence, while time-adaptive capacity reallocation progressively shifts the context budget toward long-range coverage as the stream grows. At query time, Confidence-Guided Context Expansion (CGCE) progressively exposes broader history based on model confidence, while its History Shortcut directly activates the comprehensive history for queries identified as history-dependent. By avoiding extra retrieval and auxiliary models, AutoStream maintains low query-time latency. With Qwen2.5-VL-7B as the backbone, AutoStream improves performance from 73.3% to 81.0% on StreamingBench and from 44.7% to 58.2% on the Backward subset of OVO-Bench. With Qwen3-VL-8B, it achieves 67.8% on OVO-Bench and 82.3% on StreamingBench.

1 INTRODUCTION

Multimodal large language models (MLLMs) have made substantial progress in video understanding (Bai et al., 2025b;a; Hurst et al., 2024; Singh et al., 2026; An et al., 2026), yet online settings remain challenging: streams arrive continuously and queries may appear at any time, so MLLMs must encode frames incrementally while maintaining a reliable internal state.

One line of work constructs streaming video-understanding datasets and post-trains models to adapt them to streaming scenarios. VideoLLM-online (Chen et al., 2024) and related studies (Qian et al., 2024; Zhang et al., 2024) represent some of the earliest efforts in this direction. To enhance historical memory, several methods (Zeng et al., 2025; Yao et al., 2025; Zhang et al., 2025) introduce dedicated modules that maintain visual memory representations throughout the stream. Other approaches develop streaming-specific training strategies (Chen et al., 2025; Xu et al., 2026; Guan et al., 2026; Wu et al., 2026) to reduce the gap between training and inference. Additional methods (Qian et al., 2025; Wang et al., 2025; Yang et al., 2026b) incorporate auxiliary modules to enable proactive responses during streaming. Concurrent work (Xia et al., 2026; Li et al., 2026; Yao et al., 2026; Lu et al., 2026) also explores native online video understanding with proactive response capabilities.

In parallel, a complementary line of work explores online video understanding without additional training. To address the mismatch between unbounded video stream and limited model capacity, many training-free methods focus on constructing comprehensive memory (Di et al., 2025; Xiong et al., 2025). External-memory approaches (Xiong et al., 2025; Yang et al., 2026a) store visual information in an auxiliary database and retrieve relevant evidence when a query arrives. In contrast, internal-memory approaches (Di et al., 2025; Yang et al., 2025; Zhang et al., 2026b; Xie et al., 2026b; Kim et al., 2025; Song et al., 2026; Chen et al., 2026) continuously update visual KV states

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.

to retain historical information within an acceptable context budget. Despite their different memory implementations, both paradigms are predominantly history-centric: accumulated historical information can occupy a substantial portion of the context and interfere with real-time perception.

Recently, several recent-window methods have revisited the importance of real-time perception. SimpleStream (Shen et al., 2026) shows that using only 2–8 recent frames can already attain comparable performance on mainstream online video understanding benchmarks (Li et al., 2025; Lin et al., 2024), especially on real-time tasks. Its performance on historical-memory tasks, however, remains unsatisfactory, and this gap widens as the video stream grows longer. DSCache (Pang et al., 2026) likewise finds that a historical-memory cache can undermine the effectiveness of real-time information and therefore decouples the two caches; at query time, however, it still concatenates both sources, leaving the model in a conflict between the two capabilities.

These approaches struggle to balance real-time perception with access to historical information because they apply a fixed context to every query. Yet different queries require evidence from different temporal scopes. We therefore advocate a *real-time perception first, history on demand* principle: preserve recent information by default and expose broader history only when required by the query. Several recent studies have made preliminary explorations along this direction (Liang et al., 2026; Zhang et al., 2026a; Ge et al., 2026). However, they rely on structured retrieval pipelines to selectively incorporate historical information. Without additional training (Zhang et al., 2026a; Ge et al., 2026) or auxiliary models (Liang et al., 2026; Zhang et al., 2026a), these pipelines are susceptible to retrieval errors; moreover, query-time retrieval introduces additional response latency. Based on the above analysis, we propose **AutoStream**, a training-free framework that realizes the *real-time perception first, history on demand* principle through *automatic expansion over multi-level contexts*. During stream ingestion, **Query-Ready Multi-Context Cache (QRMC)** prepares three cache components: a high-fidelity recent cache encoded independently of history, a history-conditioned recent cache that bridges recent and historical evidence, and a comprehensive historical cache maintained by **Novelty-Weighted Reservoir Sampling (NWRS)** (Vitter, 1985; Efraimidis & Spirakis, 2006). As the stream grows, time-adaptive capacity reallocation progressively shifts the context budget from recent spatial detail toward long-range historical coverage. All components are constructed before a query arrives. At query time, **Confidence-Guided Context Expansion (CGCE)** adapts different levels of cached context to each query. Inspired by the video auto reasoning paradigm (Liu et al., 2026), it starts from high-fidelity recent context and expands to broader historical context only when model confidence is insufficient. As a complementary mechanism within CGCE, History Shortcut sends queries that explicitly require historical evidence directly to the full historical context. By avoiding additional video re-encoding, auxiliary embedding models, and query-time retrieval, AutoStream maintains high computational efficiency.

In summary, our contributions are fourfold: (1) we propose AutoStream, a training-free framework that realizes the *real-time perception first, history on demand* principle; (2) we introduce QRMC, which incorporates NWRS to construct query-ready caches spanning different historical scopes during stream ingestion; (3) CGCE progressively expands cached context according to model confidence and incorporates History Shortcut as a complementary mechanism for explicitly history-oriented queries; and (4) extensive experiments demonstrate substantial improvements in online video understanding across multiple backbones and benchmarks.

2 Related Work

2.1 Training-based Streaming Video Understanding

Recent MLLMs have made substantial progress on offline video understanding, where a complete video is available at once (Bai et al., 2025b;a). Streaming video understanding instead requires models to ingest an unbounded stream, answer queries that arrive at arbitrary times, and in some settings respond proactively (Chen et al., 2024; Lin et al., 2024; Li et al., 2025). One line of work constructs streaming video-understanding data and post-trains the model to adapt to streaming scenarios (Chen et al., 2024; 2025; Qian et al., 2024; Zhang et al., 2024). To further improve memory, some methods introduce dedicated memory modules over the streaming input (Zeng et al., 2025; Yao et al., 2025; Zhang et al., 2025), while others design streaming training recipes that narrow the train–inference gap (Chen et al., 2025; Xu et al., 2026; Guan et al., 2026). In addition, some work adds auxiliary modules for proactive responses (Qian et al., 2025; Wang et al., 2025; Yang et al., 2026b). Recent

methods further fold this online interaction capability into the model's base abilities (Xia et al., 2026; Li et al., 2026; Yao et al., 2026; Lu et al., 2026). These trained systems improve online interaction, but they require model-specific fine-tuning, which is costly and does not transfer readily to frozen backbones. This motivates training-free inference-time designs that elicit streaming capability from existing video MLLMs.

2.2 Training-free Streaming Video Understanding

Given the cost of model fine-tuning and the commonality of video-understanding tasks, a growing line of work elicits streaming-understanding capability without additional training. By managing an external database (Xiong et al., 2025; Yang et al., 2026a) or internal KV cache (Di et al., 2025; Yang et al., 2025; Zhang et al., 2026b; Xie et al., 2026b; Kim et al., 2025; Song et al., 2026; Chen et al., 2026), the model can maintain a historical state while continuously ingesting new inputs. Accumulated history, however, occupies most of the context and interferes with real-time perception. SimpleStream (Shen et al., 2026) therefore retains only recent visual features: this improves results on many online benchmarks, but drops earlier events and is limited on memory-oriented tasks. These two lines of work thus trade historical memory against real-time perception. DSCache (Pang et al., 2026) decouples how the real-time and historical caches are constructed, yet still concatenates them when answering, so the interference remains at inference. These approaches struggle to balance real-time perception with access to historical information because they apply a fixed context policy to every query. A more direct solution is to preserve recent context for real-time perception and access history only on demand. Several works have begun to explore this direction (Liang et al., 2026; Zhang et al., 2026a; Ge et al., 2026); however, their structured retrieval pipelines require additional training or auxiliary modules. AutoStream instead prepares query-ready contexts with different historical scopes during stream ingestion and automatically broader history according to model confidence, enabling history on demand without query-time retrieval.

3 Method

We first present the overview of AutoStream in Section 3.1. We then describe Query-Ready Multi-Context Cache (QRMC) and Novelty-Weighted Reservoir Sampling (NWRS) during stream ingestion in Sections 3.2 and 3.3, respectively. Finally, Section 3.4 introduces Confidence-Guided Context Expansion (CGCE) at query time.

3.1 Overview: Real-Time Perception First, History on Demand

At time t, an online video understanding system has observed the video prefix $\mathcal{V}_{\leq t}$ and may receive a text query q. It must answer the query without access to future frames. AutoStream is a training-free, inference-time cache-management framework built on the principle of *real-time perception first, history on demand*. During stream ingestion, it incrementally constructs cache components spanning different temporal scopes. At query time, AutoStream automatically selects an appropriate context, defaulting to high-fidelity recent context and introducing broader history only when necessary.

As illustrated in Figure 1, AutoStream constructs three query-ready cache components during stream ingestion: an independently encoded high-fidelity *recent cache* $\mathcal{C}^{\mathrm{R}}$, a *history-conditioned recent cache* $\mathcal{C}^{\mathrm{R|H}}$, and a bounded *historical cache* $\mathcal{C}^{\mathrm{H}}$ providing broad temporal coverage. As the stream grows, Time-Adaptive Capacity Reallocation progressively shifts capacity under a fixed context budget from recent spatial detail toward long-range historical coverage.

At query time, these components are assembled into a three-level context ladder L_1–L_3, ranging from recent-only to explicit-history context. CGCE advances along this ladder according to model confidence: when an inference attempt yields low confidence, it expands to the next-level context and reruns inference using the expanded context. In addition, History Shortcut routes queries that explicitly require historical evidence directly to L_3.

3.2 Query-Ready Multi-Context Cache

Let t index the current streaming window containing the latest W frames, and let $\mathcal{S}$ denote the fixed system-prompt prefix. Each window is encoded by the ViT (Dosovitskiy et al., 2021) at two resolu-

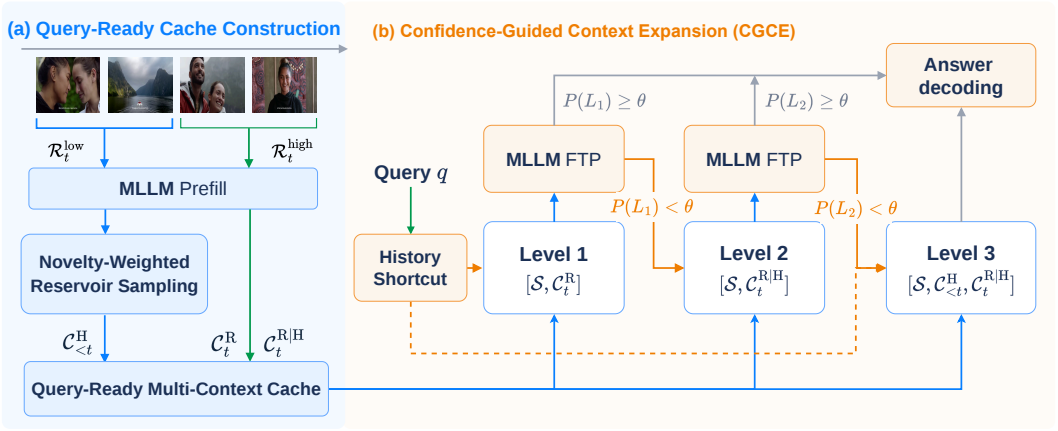


Figure 1: **Overview of AutoStream. (a)** A frozen MLLM constructs Query-Ready Multi-Context Cache (QRMC) covering recent perception to comprehensive historical coverage with Novelty-Weighted Reservoir Sampling (NWRS). **(b)** Confidence-Guided Context Expansion (CGCE) progressively exposes history when first token prediction (FTP) confidence is insufficient, while History Shortcut sends history-oriented queries directly to Level 3.

tions: the high-resolution embeddings $\mathcal{R}_t^{\text{high}}$ support recent perception, whereas the low-resolution embeddings $\mathcal{R}_t^{\text{low}}$ serve as compact historical candidates once the window becomes part of the past. We use $\text{Prefill}(\mathcal{X}; \mathcal{K})$ to denote the new KV states produced by the frozen MLLM when processing visual embeddings $\mathcal{X}$ conditioned on an existing KV prefix $\mathcal{K}$.

Before processing the current window t, AutoStream first commits the low-resolution representation of the previous window $t-1$ to history. It generates compact KV candidates conditioned on the earlier cache $\mathcal{C}_{<t-1}^{\text{H}}$ and applies NWRS to obtain the historical cache $\mathcal{C}_{<t}^{\text{H}}$ used by the current window:

$$\begin{aligned} \Delta\mathcal{C}_{t-1}^{\text{H}} &\leftarrow \text{Prefill}(\mathcal{R}_{t-1}^{\text{low}}; [\mathcal{S}, \mathcal{C}_{<t-1}^{\text{H}}]), \\ \mathcal{C}_{<t}^{\text{H}} &\leftarrow \text{NWRS}(\mathcal{C}_{<t-1}^{\text{H}}, \Delta\mathcal{C}_{t-1}^{\text{H}}). \end{aligned} \tag{1}$$

AutoStream then uses the high-resolution embeddings of the current window to construct two recent caches: one is prefilled independently of history, while the other is conditioned on the updated historical cache $\mathcal{C}_{<t}^{\text{H}}$. With $[\cdot, \cdot]$ denoting KV-cache concatenation,

$$\mathcal{C}_t^{\text{R}} \leftarrow \text{Prefill}(\mathcal{R}_t^{\text{high}}; \mathcal{S}), \qquad \mathcal{C}_t^{\text{R|H}} \leftarrow \text{Prefill}(\mathcal{R}_t^{\text{high}}; [\mathcal{S}, \mathcal{C}_{<t}^{\text{H}}]). \tag{2}$$

The recent cache $\mathcal{C}_t^{\text{R}}$ preserves a clean, history-independent representation of the current window, whereas the history-conditioned recent cache $\mathcal{C}_t^{\text{R|H}}$ serves as a bridge between recent and historical context, implicitly incorporating retained historical information without duplicating the historical cache itself.

Time-Adaptive Capacity Reallocation. A fixed per-frame resolution budget is unsuitable for an unbounded video stream, so AutoStream reallocates cache capacity as the stream grows. Let $s(t) \in \{0, \ldots, S-1\}$ denote the current stage, where r_s is the per-frame visual-token cap of the high-resolution recent embeddings $\mathcal{R}_t^{\text{high}}$ and B_s is the total budget of the historical cache $\mathcal{C}^{\text{H}}$. Under a fixed total context-length constraint, we set $r_{s+1} \leq r_s$ and $B_{s+1} \geq B_s$, progressively reallocating capacity from high-resolution recent observations to the historical cache. As the stream grows longer, this reallocation adaptively provides broader coverage. The concrete stages and budgets are given in the experiments. Because increasing B_s alone does not determine which historical content should be retained, we next introduce NWRS.

3.3 NOVELTY-WEIGHTED RESERVOIR SAMPLING

Unlike FIFO strategy, which systematically discards early KV blocks as the stream progresses, we employ online algorithm reservoir sampling (Vitter, 1985) to avoid recency bias. After filling the cache with the first K candidates, it admits the i-th candidate with probability K/i and randomly replaces a stored block, ensuring that each of the T observed blocks has the same marginal retention probability K/T. However, this approach is content-agnostic: it assigns the same retention probability to every candidate KV block.

To bias the bounded historical cache toward novel content, Novelty-Weighted Reservoir Sampling assigns each candidate block a semantic novelty weight computed from its associated visual embedding. Let $\mathbf{z}_i$ denote the normalized frame-level visual embedding associated with the i-th candidate KV block. Its local and global novelty scores are defined as

$$n_i^{\mathrm{loc}} = 1 - \max_{\substack{\mathbf{z}_j \in \mathcal{N}_i \\ j<i}} \mathbf{z}_i^\top \mathbf{z}_j, \qquad n_i^{\mathrm{glob}} = 1 - \max_{\substack{\mathbf{z}_k \in \mathcal{H}_{i-1} \\ k<i}} \mathbf{z}_i^\top \mathbf{z}_k. \tag{3}$$

Here, $\mathcal{N}_i$ denotes a sliding reference buffer of frame-level visual embeddings, whereas $\mathcal{H}_{i-1}$ contains the visual embeddings associated with the KV blocks currently retained in $\mathcal{C}^{\mathrm{H}}_{<t}$. Each embedding in $\mathcal{H}_{i-1}$ is paired with its corresponding historical KV block and removed when that block is evicted. We combine the two novelty scores as $s_i = 1-(1-n_i^{\mathrm{loc}})(1-n_i^{\mathrm{glob}})$ and define the novelty weight as $\omega_i = \beta^{s_i}$, where $\beta = 4$ controls the strength of the novelty bias. Candidate KV blocks associated with locally distinctive or historically underrepresented visual content therefore receive larger weights.

We incorporate novelty through priority-based weighted reservoir sampling (Efraimidis & Spirakis, 2006). For every candidate KV block, we draw $U_i \sim \mathrm{Uniform}(0,1)$ once at insertion, set $u_i = -\ln U_i$, and construct an immutable exponential priority key

$$p_i = \frac{u_i}{\omega_i}. \tag{4}$$

When all ω_i are equal, this procedure is distributionally equivalent to unweighted reservoir sampling. The two factors in the priority key play complementary roles: u_i provides stochastic selection and broad coverage of the stream, whereas ω_i biases retention toward KV blocks associated with novel visual content. Specifically, a larger ω_i lowers the priority $p_i = u_i/\omega_i = -\ln U_i/\omega_i$ and therefore increases the probability that candidate block i is retained. A detailed derivation is provided in Appendix D. The retained KV blocks are finally restored to chronological order and concatenated to form the historical cache $\mathcal{C}^{\mathrm{H}}$.

3.4 CONFIDENCE-GUIDED CONTEXT EXPANSION

Inspired by video auto reasoning (Liu et al., 2026), we treat historical context as an on-demand resource analogous to chain-of-thought reasoning. We observe that, when the model answers using only the recent window, answer accuracy is positively correlated with first-token probability (Appendix B). We therefore use first-token probability as a confidence signal: Confidence-Guided Context Expansion (CGCE) starts each query from the real-time perception context and progressively introduces on-demand history only for low-confidence queries. This provides additional evidence for uncertain queries while preventing historical information from interfering with those already answerable from the current context.

At query time, the three cache components are assembled into a three-level context ladder L_1–L_3. Formally, for the current window t, Levels 1–3 context are composed as follows:

$$L_1 = [\mathcal{S}, \mathcal{C}_t^{\mathrm{R}}], \qquad L_2 = [\mathcal{S}, \mathcal{C}_t^{\mathrm{R|H}}], \qquad L_3 = [\mathcal{S}, \mathcal{C}^{\mathrm{H}}_{<t}, \mathcal{C}_t^{\mathrm{R|H}}]. \tag{5}$$

Let $P(L_\ell)$ denote the probability assigned to the predicted first output token at Level ℓ. When this confidence falls below the threshold θ, CGCE advances to the next level:

$$L_1 \xrightarrow{P(L_1)<\theta} L_2 \xrightarrow{P(L_2)<\theta} L_3. \tag{6}$$

The model returns the answer from the first level whose confidence reaches the threshold; if neither of the first two levels is sufficiently confident, Level 3 produces the final answer.

History Shortcut. Model confidence may be inflated by visual cues in the recent context, causing the model to overlook the historical requirements of some queries. To address this issue, we introduce a complementary History Shortcut. We provide the model with the user's query together with an instruction prompt, asking it to determine whether historical evidence is required to answer the query. Queries identified as history-dependent are routed directly to Level 3.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation Details. We implement AutoStream with Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-4B, and Qwen3-VL-8B (Bai et al., 2025a), using the recent-window method SimpleStream (Shen et al., 2026) as our baseline. For QRMC, we fix the recent-window size to $W = 4$ frames and divide time-adaptive capacity reallocation into three stages: 0–120 seconds, 120–1000 seconds, and beyond 1000 seconds. The overall context quota remains fixed at 12K tokens across all stages, while capacity is progressively reallocated from recent representations to historical memory as the video unfolds. Specifically, the per-frame visual-token cap r_s for the high-resolution recent embeddings $\mathcal{R}_t^{\text{high}}$ is set to 768, 512, and 256 in the three stages, respectively, while the corresponding total budget B_s for the historical cache $\mathcal{C}_{<t}^{\text{H}}$ is set to 6K, 8K, and 10K tokens. For the low-resolution embeddings $\mathcal{R}_t^{\text{low}}$ used to construct historical candidates, the per-frame visual-token cap is set to 64 on Qwen2.5-VL-7B and 128 on both Qwen3-VL models. For CGCE, we set the confidence threshold to $\theta = 0.7$. Unless otherwise specified, all ablation studies use Qwen2.5-VL-7B as the backbone.

Benchmarks. We evaluate AutoStream on two representative online video understanding benchmarks: OVO-Bench (Li et al., 2025) and StreamingBench (Lin et al., 2024). Following prior work, we report results on the Real-Time Visual Perception and Backward Tracing categories of OVO-Bench, as well as the Real-Time Visual Understanding category of StreamingBench. To further assess general video understanding and long-horizon memory in offline settings, we additionally evaluate on Video-MME (Fu et al., 2025), MVBench (Li et al., 2024), and MLVU (Zhou et al., 2025).

Compared Methods. We primarily compare training-free methods implemented on the same MLLM, including *history-memory methods* such as HERMES (Zhang et al., 2026b) and FluxMem (Xie et al., 2026b), *recent-window methods* such as SimpleStream (Shen et al., 2026) and DSCache (Pang et al., 2026), and *history-on-demand methods* such as OASIS (Liang et al., 2026) and StreamScout (Zhang et al., 2026a). We additionally report selected proprietary models (Team et al., 2024; Hurst et al., 2024; OpenAI, 2025) and open-source training-based online MLLMs (Zeng et al., 2025; Xia et al., 2026; Yao et al., 2026; Yang et al., 2026b; Lu et al., 2026) as references.

4.2 MAIN RESULTS

Table 1 compares AutoStream with other methods on StreamingBench and OVO-Bench. Following prior work (Shen et al., 2026; Wu et al., 2026), we additionally report OVO-Bench results excluding HLD. Across all three backbones, AutoStream consistently improves upon the recent-window baseline SimpleStream on StreamingBench and backward-looking reasoning while retaining competitive real-time perception. On Qwen2.5-VL-7B, AutoStream raises StreamingBench from 78.5% to 81.0% and OVO-Bench Backward from 51.9% to 58.2%. The consistent gains show that exposing history on demand strengthens temporal reasoning while preserving strong responsiveness to recent visual content. AutoStream also outperforms the history-on-demand methods OASIS and StreamScout across all reported metrics when instantiated with Qwen2.5-VL-7B. With Qwen3-VL-8B, it achieves 82.3% on StreamingBench and 71.2% on OVO-Bench Avg. without HLD, the best results among training-free methods using the same backbone. Table 2 reports complementary results on offline video-understanding benchmarks. Using the same configuration as in online video understanding, AutoStream outperforms the recent-window baseline and OASIS on long-video benchmarks such as Video-MME, while largely preserving the backbone's general video-understanding capability. These results demonstrate that AutoStream can adaptively incorporate historical informa-

Table 1: **Performance comparison (%) on StreamingBench and OVO-Bench**. StreamingBench reports Real-Time Visual Understanding. In Backward and Avg., we report *with/without* HLD (left/right). Best and second-best among *training-free* methods in each backbone group are in **bold** and underlined. "–": unreported. †: adaptive per-question frames.

Model	#Frames	StreamingBench Real-Time	OVO-Bench Real-Time	OVO-Bench Backward	Avg.
Proprietary MLLMs					
Gemini 1.5 Pro	1 fps	75.7	69.3	62.5/67.5	65.9/68.4
GPT-4o	64	73.3	64.5	60.8/66.8	62.6/65.7
GPT-5.1	64	83.5	77.2	58.1/67.8	67.6/72.5
Open-source Online MLLMs					
StreamForest-7B	1 fps	77.3	61.2	52.0/61.9	56.6/61.6
Streamo-7B	1 fps	–	66.0	46.1/53.3	56.1/59.7
JoyAI-VL-Interaction-8B	1 fps	82.7	70.9	48.3/60.9	59.6/65.9
Mage-VL-4B	1 fps	82.1	79.8	48.2/55.8	64.0/67.8
AURA-8B	1 fps	83.2	79.8	60.4/61.3	70.1/70.6
Training-free Offline-to-Online Methods					
Qwen2.5-VL-7B	1 fps	73.3	59.9	44.7/55.2	52.3/57.6
+ HERMES	1 fps	79.4	69.0	49.4/55.3	59.2/62.2
+ FluxMem	1 fps	–	66.3	46.4/56.3	56.4/61.3
+ DSCache	1 fps	**82.3**	71.7	44.8/53.3	58.3/62.5
+ OASIS	48 / 80†	70.6	67.3	52.6/55.3	59.9/61.3
+ StreamScout	4 / 12 / 24†	77.7	75.3	50.1/56.8	62.7/66.1
+ SimpleStream	4	78.5	**78.4**	51.9/57.7	65.2/**68.1**
+ **AutoStream**	**1 FPS**	81.0	75.9	**58.2/60.2**	**67.1/68.1**
Qwen3-VL-8B	1 fps	75.7	68.2	46.4/55.7	57.3/62.0
+ HERMES	2 fps	81.3	73.3	49.3/61.6	61.3/67.5
+ OASIS	48 / 80†	78.2	77.3	58.5/**62.9**	67.9/70.1
+ StreamScout	4 / 12 / 24†	80.3	77.7	**59.1**/58.4	**68.4**/68.1
+ SimpleStream	4	80.6	**81.4**	54.0/55.0	67.7/68.2
+ **AutoStream**	**1 FPS**	**82.3**	80.6	54.9/61.9	67.8/**71.2**
Qwen3-VL-4B	1 fps	74.4	63.4	47.5/57.5	54.1/60.5
+ HERMES	2 fps	78.2	72.3	55.0/59.7	63.7/66.0
+ SimpleStream	4	80.3	79.3	52.4/–	65.81/–
+ **AutoStream**	**1 FPS**	**81.5**	**80.4**	**55.7/61.2**	**68.1/70.8**

Table 2: **Performance comparison (%) on offline video-understanding benchmarks.** Video-MME is without subtitles. MLVU reports M-Avg.

Model	MVBench Avg.	MLVU M-Avg.	Video-MME Long	Video-MME Avg.
Qwen2.5-VL-7B	65.0	67.9	53.8	63.3
+ SimpleStream	56.3	47.6	38.3	41.9
+ DSCache	58.7	53.1	42.6	50.7
+ OASIS	58.5	49.2	42.9	51.5
+ **AutoStream**	64.1	66.2	53.6	62.0

tion when needed, overcoming the inability of recent-window methods to access long-term evidence while providing more effective history access than retrieval-based approaches such as OASIS.

4.3 Qualitative Result

Figure 2 presents a qualitative example of AutoStream at inference time. Given the contexts constructed by the Query-Ready Multi-Context Cache, the model first attempts to answer using the recent-window contexts at Levels 1 and 2, but exhibits low confidence at both levels. It then expands to the Level 3 context, producing the correct answer with high confidence. The question

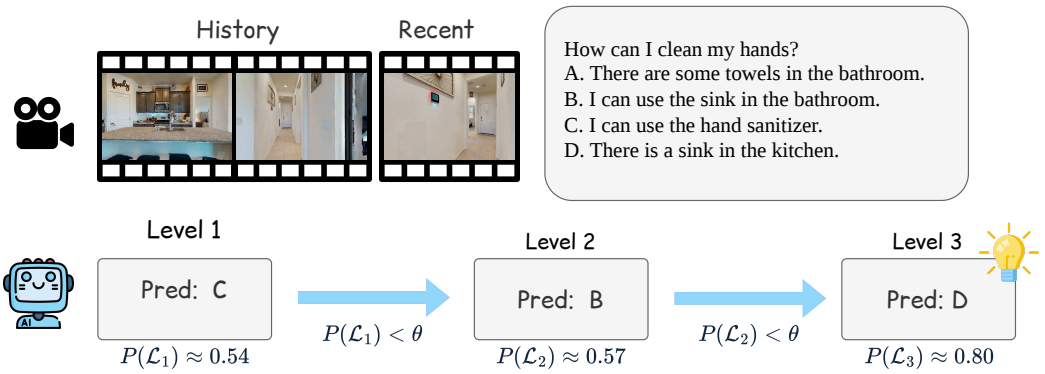


Figure 2: **Qualitative result of AutoStream.** AutoStream progressively expands from recent context to explicit historical context, producing the correct answer once sufficient historical evidence is available. The confidence threshold is set to $\theta = 0.7$.

Table 3: **Efficiency analysis.** "TTFT" / "TPOT": Time to First Token / Time Per Output Token.

Metric	Expansion		
	Level 1	1→2	1→2→3
GPU Mem. / GB ↓	18.4	18.4	18.6
TTFT / ms ↓	35.0	51.9	74.0
TPOT / ms ↓	14.9	15.0	14.9

Table 4: **Ablation on different context levels.** Best results are shown in bold.

Model	OVO-Bench Avg	StreamingBench Avg
w/o Level 1	61.7	79.7
w/o Level 2	62.3	80.4
w/o Level 3	**67.1**	79.5
Ours	**67.1**	**81.0**

indeed requires historical evidence beyond the recent window, making the model's expansion behavior consistent with its temporal requirements.

4.4 Efficiency Analysis

Following Zhang et al. (2026b), we measure query-time efficiency after the visual cache has already been built, starting from the arrival of the question. All measurements are taken on a single NVIDIA H200 GPU. Because AutoStream does not modify the KV cache, the setup is fully compatible with FlashAttention-2 (Dao, 2023). Table 3 reports efficiency under successive context expansion at 512 frames, including the time required by History Shortcut, with results across 64–1024 frames provided in Appendix Table 10. As the context expands from Level 1 to Level 3, peak GPU memory and TPOT remain stable. TTFT increases because each expansion step requires an additional first-token prediction to estimate model confidence, but remains low in absolute terms.

4.5 Ablation Studies

Impact of Different Context Levels. Table 4 removes one context level at a time to examine the complementary roles of the three query-ready caches constructed by QRMC. Removing Level 1 or Level 2 reduces OVO-Bench from 67.1 to 61.7 and 62.3, respectively, while lowering StreamingBench from 81.0 to 79.7 and 80.4. This shows that the clean high-fidelity recent cache and the history-conditioned recent cache are not interchangeable: the former preserves current visual details, whereas the latter bridges recent observations with retained history. In contrast, removing Level 3, which serves as an explicit-history fallback, reduces StreamingBench from 81.0 to 79.5, demonstrating its value when evidence spans a broader temporal range. Together, the three caches provide complementary temporal scopes that enable on-demand context expansion without imposing equal historical context on every query.

Table 5: **Ablation on confidence threshold θ.** Best results are shown in bold.

Setting	OVO-Bench Avg	StreamingBench Avg
$\theta = 0.9$	66.2	79.4
$\theta = 0.8$	66.8	80.1
$\theta = 0.6$	**67.2**	80.5
$\theta = 0.5$	67.0	80.6
$\theta = 0.7$	67.1	**81.0**

Table 6: **Ablation on History Shortcut.** Best results are shown in bold.

Benchmark	w/o History Shortcut	**Ours**
OVO-Bench	65.0	**67.1**
StreamingBench	80.4	**81.0**
MVBench	61.0	**64.1**
MLVU	59.8	**66.2**
Video-MME	53.7	**62.0**

Table 7: **Ablation on Novelty-Weighted Reservoir Sampling.** All policies maintain the historical cache $\mathcal{C}^{\mathrm{H}}_{<t}$ under the same context budget. Best results are shown in bold.

History policy	OVO-Bench		StreamingBench	Video-MME	
	Backward	Avg	Avg	Long	Avg.
FIFO / recency	57.8	66.8	80.5	48.4	58.2
Random-only	58.1	**67.2**	80.5	52.4	61.2
Novelty-only	57.2	66.4	80.8	**53.6**	61.4
Ours (NWRS)	**58.2**	67.1	**81.0**	**53.6**	**62.0**

Impact of Confidence-Guided Context Expansion. Table 5 studies the confidence threshold governing the main expansion process. Raising θ to 0.9 expands more high-confidence queries, introducing unnecessary context and reducing OVO-Bench and StreamingBench by 0.9 and 1.6 points, respectively. This supports CGCE's intended role of supplementing low-confidence queries rather than altering already-confident responses. Conversely, lowering θ to 0.6 yields only a 0.1-point gain on OVO-Bench while reducing StreamingBench by 0.5 points. We therefore use θ=0.7 as a balanced default. Table 6 evaluates the impact of History Shortcut. While disabling it causes only a 0.6-point drop on StreamingBench, the degradation grows to 2.1 points on OVO-Bench, 3.1 on MVBench, 6.4 on MLVU, and 8.3 on Video-MME, showing its stronger benefit when questions require broader historical evidence. By routing queries with explicit historical intent directly to Level 3, History Shortcut prevents premature answers based only on insufficient recent context.

Impact of Novelty-Weighted Reservoir Sampling. Table 7 compares policies for maintaining the historical cache $\mathcal{C}^{\mathrm{H}}_{<t}$ under the same context budget. Compared with recency-only FIFO, NWRS improves OVO-Bench Avg from 66.8 to 67.1, StreamingBench from 80.5 to 81.0, and the Video-MME average from 58.2 to 62.0. Another variants ablate the two factors of the priority key ($p_i = u_i/\omega_i$) in NWRS. Random-only equalizes the novelty weights ω_i in the denominator, leaving the ranking determined solely by the stochastic term u_i; it achieves only 80.5 and 61.2 on StreamingBench and Video-MME. Novelty-only instead equalizes the stochastic terms u_i in the numerator,it reduces OVO-Bench Avg to 66.4. By combining randomized temporal coverage with novelty-aware weighting, NWRS achieves the best or near-best overall performance across all three benchmarks.

5 Conclusion

This paper presents AutoStream, a training-free framework for online video understanding built on a *real-time perception first, history on demand* principle. As the video stream unfolds, AutoStream constructs QRMC, dynamically allocating capacity across context levels according to the observed video length. Under a bounded memory budget, NWRS maintains a historical cache that balances semantic novelty with broad temporal coverage. Upon receiving a query, AutoStream automatically constructs an appropriate context for each query. CGCE progressively evaluates model confidence and expands the context when necessary. History Shortcut provides a complementary path that directly routes queries with strong historical relevance. Experimental results demonstrate that AutoStream substantially improves backward tracing and overall performance while preserving competitive real-time perception, stable GPU memory usage, and low latency.

AI USE STATEMENT

In this work, we used generative AI tools to support qualitative and thematic data analysis and to assist with translation. Additionally, we used generative AI tools to edit the manuscript for readability, identify relevant literature, and create or edit software code. We have reviewed all AI-assisted work. All AI-generated contents were manually checked by the authors for correctness before inclusion. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. Llava-onevision-2: Towards next-generation perceptual intelligence, 2026. URL https://arxiv.org/abs/2605.25979.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 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, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 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, 2024. URL https://arxiv.org/abs/2406.11816.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Livecc: Learning video llm with streaming speech transcription at scale, 2025. URL https://arxiv.org/abs/2504.16030.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding, 2026. URL https://arxiv.org/abs/2510.18269.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Pavlos S. Efraimidis and Paul G. Spirakis. Weighted random sampling with a reservoir. *Inf. Process. Lett.*, 97(5):181–185, March 2006. ISSN 0020-0190.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis, 2025. URL https://arxiv.org/abs/2405.21075.

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

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: Videollms can watch and think simultaneously, 2026. URL https://arxiv.org/abs/2603.12262.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding, 2025. URL https://arxiv.org/abs/2506.15745.

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, 2024. URL https://arxiv.org/abs/2311.17005.

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, Yuhao Dong, Haoning Wu, Zhiqiu Zhang, Yuandong Yang, Changlian Ma, Qingyu Zhang, Yansong Shi, Xinyu Chen, Haoran Chen, Zizheng Huang, Jun Zhang, Kun Ouyang, Lin Sui, Ziang Yan, Yicheng Xu, Chenting Wang, Yinan He, Hongjie Zhang, Yi Wang, Yu Qiao, Yali Wang, Ziwei Liu, Kai Chen, and Limin Wang. Videochat3: Fully open video mllm for efficient and generalist video understanding, 2026. URL https://arxiv.org/abs/2607.14935.

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?, 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, 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, 2024. URL https://arxiv.org/abs/2411.03628.

Shuming Liu, Mingchen Zhuge, Changsheng Zhao, Jun Chen, Lemeng Wu, Zechun Liu, Chenchen Zhu, Zhipeng Cai, Chong Zhou, Haozhe Liu, Ernie Chang, Saksham Suri, Hongyu Xu, Qi Qian, Wei Wen, Balakrishnan Varadarajan, Zhuang Liu, Hu Xu, Florian Bordes, Raghuraman Krishnamoorthi, Bernard Ghanem, Vikas Chandra, and Yunyang Xiong. Videoauto-r1: Video auto reasoning via thinking once, answering twice, 2026. URL https://arxiv.org/abs/2601.05175.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. Aura: Always-on understanding and real-time assistance via video streams, 2026. URL https://arxiv.org/abs/2604.04184.

OpenAI. Gpt-5.1: A smarter, more conversational chatgpt. https://openai.com/index/gpt-5-1, nov 2025.

Zhanzhong Pang, Dibyadip Chatterjee, Fadime Sener, and Angela Yao. Decouple and cache: Kv cache construction for streaming video understanding, 2026. URL https://arxiv.org/abs/2605.01858.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models, 2024. URL https://arxiv.org/abs/2405.16009.

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, 2025. URL https://arxiv.org/abs/2501.03218.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding, 2026. URL https://arxiv.org/abs/2604.02317.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card, 2026. URL https://arxiv.org/abs/2601.03267.

Baiyang Song, Yuli Lin, Qiong Wu, Tao Chen, Jun Peng, Xiao Chen, Yiyi Zhou, and Rongrong Ji. Towards a dynamic and fixed-budget memory bank for efficient streaming video understanding, 2026. URL https://arxiv.org/abs/2606.25658.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, March 1985. ISSN 0098-3500. doi: 10.1145/3147.3165. URL https://doi.org/10.1145/3147.3165.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant, 2025. URL https://arxiv.org/abs/2505.05467.

Keming Wu, Baoyi Wang, Kaichen Zhang, Xiang An, Zuhao Yang, Sudong Wang, Haowei Zhu, Tingxuan Huang, Hongcheng Gao, and Bin Wang. Streamopd: A post-training recipe with spatio-temporal cue gating for streaming video understanding, 2026. URL https://arxiv.org/abs/2608.16320.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning, 2026. URL https://arxiv.org/abs/2512.21334.

Ming Xie, Zizheng Huang, Xudong Tan, Chao Wang, Xiangyu Zeng, Wenxiao Wu, Tao Chen, Limin Wang, and Yanwei Fu. Streamov: Streaming omni-video understanding via evidence-guided memory and response triggering, 2026a. URL https://arxiv.org/abs/2605.25621.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. Fluxmem: Adaptive hierarchical memory for streaming video understanding, 2026b. 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, 2025. URL https://arxiv.org/abs/2501.13468.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams, 2026. URL https://arxiv.org/abs/2510.09608.

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, 2026a. URL https://arxiv.org/abs/2508.01875.

Senqiao Yang, Kaichen Zhang, Zhaoyang Jia, Jinghao Guo, Yifei Shen, Xinjie Zhang, Xiaoyi Zhang, Haoqing Wang, Xiao Li, Peng Zhang, Xiang An, Yin Xie, Zhening Liu, Xun Guo, Jiahao Li, Shicheng Zheng, Jinglu Wang, Zongyu Guo, Wenxuan Xie, Zihan Zheng, Yuxuan Luo, Bin Li, and Yan Lu. Mage-vl: An efficient codec-native streaming multimodal foundation model, 2026b. URL https://arxiv.org/abs/2607.24904.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. Streammem: Query-agnostic kv cache memory for streaming video understanding, 2025. URL https://arxiv.org/abs/2508.15717.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Xiangyu Zeng, Yifei Li, Haowen Hou, Zheming Liang, Congcong Wang, Kaiwen Tuo, Jun Zhang, Yuhan Zhu, Yuhang Cao, Shenglong Ye, Shuai Xie, Shuhuan Gu, Haoyang Huang, Qingyi Si, Nan Duan, and Jiaqi Wang. Joyai-vl-interaction: Real-time vision-language interaction intelligence, 2026. URL https://arxiv.org/abs/2606.14777.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. TimeChat-Online: 80% Visual Tokens are Naturally Redundant in Streaming Videos, 2025. URL https://arxiv.org/abs/2504.17343.

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

Ce Zhang, Jing Bi, Jinxi He, Jianshu Zhang, Jingyang Lin, Yunzhong Xiao, Minghao Fu, Yaqi Xie, Zhentao Xie, Weicong Chen, Katia Sycara, and Ming Zhou. Streamscout: Learning when to look deeper for streaming video understanding, 2026a. URL https://arxiv.org/abs/2609.00291.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams, 2024. URL https://arxiv.org/abs/2406.08085.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams, 2025. URL https://arxiv.org/abs/2506.23825.

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

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding, 2025. URL https://arxiv.org/abs/2406.04264.

APPENDIX

Table of contents:

- §A: Generalizability Analysis
- §B: Analysis of Token Probability and Answer Correctness
- §C: Qualitative Example of History Shortcut
- §D: Theoretical Analysis of NWRS
- §E: Pseudocode of QRMC and NWRS
- §F: More results for Efficiency Analysis
- §G: Ablation of Time-Adaptive Capacity Reallocation
- §H: Limitations and Future Work

A GENERALIZABILITY ANALYSIS

Table 8: **Performance comparison (%) on StreamingBench and OVO-Bench**. Best results are in **bold**.

Model	#Frames	StreamingBench	OVO-Bench		
		Real-Time	Real-Time	Backward	**Avg.**
Qwen2.5-VL-32B	1 fps	74.3	64.4	50.3	57.4
+ HERMES	1 fps	80.1	72.4	55.4	63.9
+ SimpleStream	4	79.2	**83.0**	48.6	65.8
+ **AutoStream**	1 fps	**81.2**	79.6	**57.9**	**68.8**
Qwen3-VL-30B-A3B	1fps	78.1	63.6	54.6	59.1
+ SimpleStream	2	80.0	82.7	58.6	70.6
+ **AutoStream**	1 fps	**82.7**	**84.2**	**60.2**	**72.2**

Table 9: **Performance comparison (%) on StreamingBench and OVO-Bench**. Best results are shown in bold.

Model	#Frames	StreamingBench	OVO-Bench		
		Real-Time	Real-Time	Backward	**Avg.**
Qwen2.5-VL-7B	1 fps	73.3	59.9	44.7	52.3
+ HERMES	1 fps	79.4	69.0	49.4	59.2
+ HERMES + CGCE	1 fps	**80.8**	**74.8**	**50.3**	**62.3**

In Table 8, we additionally report AutoStream on Qwen2.5-VL-32B (Bai et al., 2025b) and Qwen3-VL-30B-A3B (Bai et al., 2025a), showing that it generalizes across model scales and architectures.

We further apply Confidence-Guided Context Expansion (CGCE) to HERMES (Zhang et al., 2026b) as a **plug-and-play module**. Concretely, on top of HERMES's native memory, we maintain an additional recent window to form a two-level context structure and expand from the recent context to the full context according to model confidence. The results are shown in Table 9. HERMES is a training-free method centered on historical memory. Introducing CGCE substantially improves its performance on real-time tasks while preserving performance on memory-oriented tasks, without changing its original memory design or model architecture.

B ANALYSIS OF TOKEN PROBABILITY AND ANSWER CORRECTNESS

Probability and answer correctness. We analyze recent-window baseline (i.e., SimpleStream (Shen et al., 2026)) answers from Qwen2.5-VL-7B and Qwen3-VL-8B on OVOBench and StreamingBench. We measure confidence using the predicted first-token probability and the geometric mean of all generated-token probabilities (Liu et al., 2026). Across all four model–benchmark settings, answer accuracy increases with the probability quintile for both measures; Figure 3 shows the first-token results. Linear probability models with video-ID-clustered robust standard errors yield positive slopes in all eight tests, each remaining significant after joint Holm

correction (adjusted $p < 0.001$). This association supports using low first-token probability as a signal for context expansion in Section 3.4, but does not establish calibrated correctness probabilities or the causal benefit of expansion.

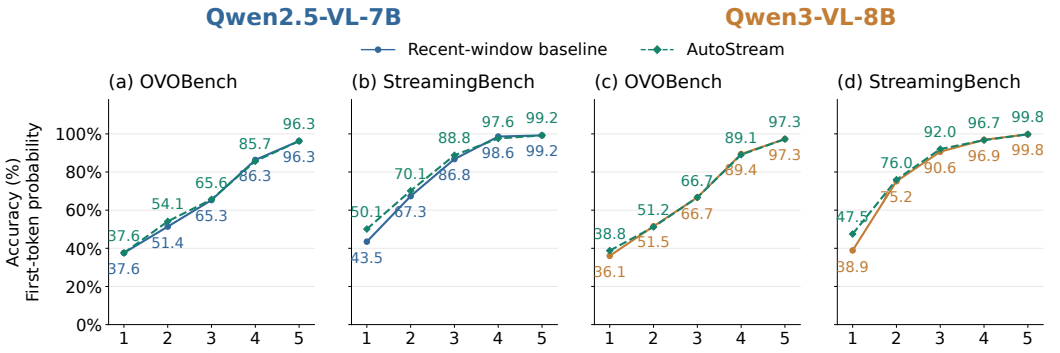


Figure 3: **Answer accuracy by first-token-probability quintile.** Solid curves show recent-window accuracy; green dashed curves show AutoStream accuracy on the same baseline-defined groups. The indices 1–5 denote probability-rank groups from low to high rather than equally spaced probability values.

Choice of confidence measure. We analyze the monotonic relationship between first-token and mean-token probabilities. Across the four combinations of two models and two benchmarks, their mean Spearman rank correlation is $\rho = 0.92$, indicating that the two measures rank question confidence similarly and thus yield comparable context-expansion decisions. First-token probability is available immediately after the first token is generated, whereas mean-token probability requires the complete answer. We therefore adopt first-token probability as a more timely expansion signal, avoiding full decoding of the initial answer before deciding whether to expand the context and thereby helping reduce the TTFT of the final answer.

Impact of Confidence-Guided Context Expansion. CGCE is motivated by the observation that low model confidence often indicates that the recent context is insufficient for reliable answering, calling for broader historical context to be introduced on demand. As shown in Figure 3, AutoStream generally achieves higher accuracy in the lower baseline first-token-probability quintiles, with the clearest gains on StreamingBench, while the two curves converge as baseline confidence increases. This pattern is consistent with CGCE: broader historical context is allocated to low-confidence questions that are most likely to benefit from it, thereby improving overall answer accuracy.

C Qualitative Analysis of History Shortcut

Figure 4 qualitatively illustrates how History Shortcut operates when triggered. For a counting question that requires evidence from earlier video content, History Shortcut recognizes the historical dependency and directly activates Level 3. This allows the model to incorporate the relevant historical evidence without first relying on insufficient recent context.

D Theoretical Analysis of NWRS

Uniform reservoir sampling. Classical reservoir sampling maintains a uniform sample without replacement from a stream of unknown length using one pass and fixed memory (Vitter, 1985). A reservoir of capacity K first stores the initial K candidates; when candidate $i > K$ arrives, it is admitted with probability K/i and replaces a uniformly selected resident. After any t candidates, every candidate has marginal retention probability K/t, and every size-K subset is equally likely. Thus, *uniform* denotes statistical uniformity over the observed candidates rather than deterministic

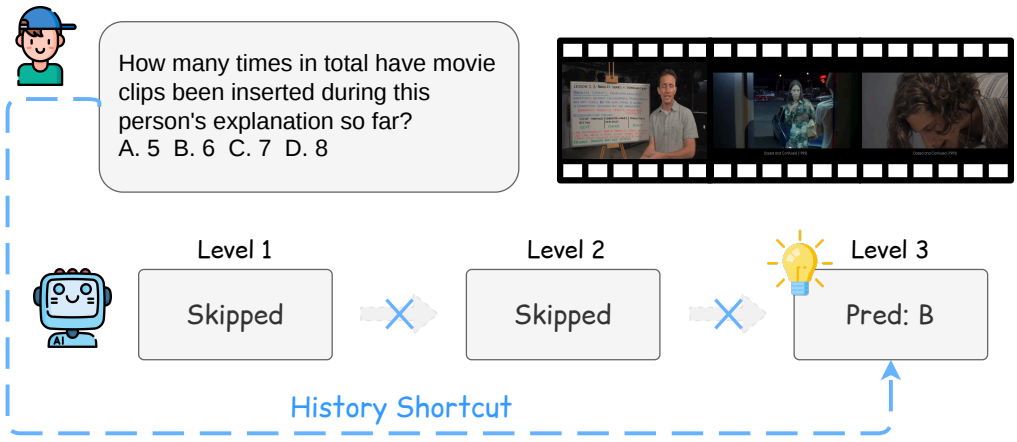


Figure 4: **Qualitative example of History Shortcut when triggered.**

sampling at uniformly spaced timestamps. The latter requires the video duration to be known in advance, which is infeasible in the online setting. Reservoir sampling provides a theoretically reliable alternative.

Priority-based implementation. Following Efraimidis & Spirakis (2006), we use a priority-based implementation that extends naturally to nonuniform weights. Each candidate draws $U_i \sim \mathrm{Uniform}(0,1)$ once at insertion, constructs $p_i = -\ln U_i/\omega_i$ as in Eq. equation 4, and the K smallest priorities are retained. When all ω_i are equal, the priorities are i.i.d. continuous random variables, so every priority ordering is equally likely. Consequently, each size-K subset is selected with probability $\binom{t}{K}^{-1}$ and every candidate again has marginal probability K/t. The priority-based and classical reservoirs are therefore distributionally equivalent, although their step-wise updates differ.

Weighted reservoir sampling. For nonuniform weights, $-\ln U_i \sim \mathrm{Exp}(1)$, so p_i is exponentially distributed with rate ω_i. A larger ω_i is therefore more likely to produce a smaller priority; in particular,

$$\Pr\left(i = \arg\min_j p_j\right) = \frac{\omega_i}{\sum_j \omega_j}. \tag{7}$$

Ranking candidates by these priorities yields weighted sampling without replacement. We refer readers to Efraimidis & Spirakis (2006) for a fuller treatment; here, we use this established result only to explain how novelty weights modify retention probabilities.

Novelty weighting. In our method, $s_i = 1 - (1 - n_i^{\mathrm{loc}})(1 - n_i^{\mathrm{glob}})$ is monotonic in both local and global novelty, so a frame receives greater importance when either novelty signal is high. The mapping $\omega_i = \beta^{s_i}$ keeps the weight positive and bounded within $[1, \beta]$, introducing a controlled novelty bias. When $\beta = 1$, the method reduces to uniform reservoir sampling; when $\beta > 1$, novel frames receive a greater chance of retention, while U_i preserves stochastic coverage instead of reducing the method to a deterministic novelty-only ranking.

E Pseudocode of QRMC and NWRS

Algorithms 1 and 2 follow the temporal semantics of Eq. (1): before processing window t, NWRS pairs the post-ViT frame embeddings in $\mathcal{R}_{t-1}^{\mathrm{low}}$ with per-frame candidate KV blocks $\Delta\mathcal{C}_{t-1,i}^{\mathrm{H}}$ and updates $\mathcal{C}_{<t-1}^{\mathrm{H}}$ to $\mathcal{C}_{<t}^{\mathrm{H}}$ under budget B_s. The persistent records $\mathcal{M}$ store each visual embedding, its KV block, timestamp, and immutable priority; $\mathcal{N}_i$ and $\mathcal{H}_{i-1}$ contain frame-level visual embeddings only. Although $\mathcal{N}_i$ is also maintained as a sliding window, it is updated independently because the

first few frames in the $\mathcal{R}^{\text{low}}_{t-1}$ require embeddings from earlier windows to compute their novelty scores.

Algorithm 1 Query-Ready Multi-Context Cache Construction (QRMC)

Require: Streaming windows represented by post-ViT embeddings $\mathcal{R}^{\text{high}}_t$ and $\mathcal{R}^{\text{low}}_t$; system prefix $\mathcal{S}$; historical-budget schedule $B_{s(t)}$; novelty bias $\beta > 1$

$\mathcal{C}^{\text{H}}_{<1} \leftarrow \emptyset$; initialize persistent record set $\mathcal{M}$ and local reference buffer $\mathcal{N}$
for each current streaming window $t = 1, 2, \ldots$ **do**
  if $t > 1$ **then**
    $\Delta\mathcal{C}^{\text{H}}_{t-1} \leftarrow \text{Prefill}(\mathcal{R}^{\text{low}}_{t-1}; [\mathcal{S}, \mathcal{C}^{\text{H}}_{<t-1}])$
    $\mathcal{C}^{\text{H}}_{<t} \leftarrow \text{NWRS}(\mathcal{C}^{\text{H}}_{<t-1}, \mathcal{R}^{\text{low}}_{t-1}, \Delta\mathcal{C}^{\text{H}}_{t-1}; B_{s(t)}, \beta)$
  end if
  $\mathcal{C}^{\text{R}}_t \leftarrow \text{Prefill}(\mathcal{R}^{\text{high}}_t; \mathcal{S})$
  $\mathcal{C}^{\text{R|H}}_t \leftarrow \text{Prefill}(\mathcal{R}^{\text{high}}_t; [\mathcal{S}, \mathcal{C}^{\text{H}}_{<t}])$
  Make $\mathcal{C}^{\text{R}}_t$, $\mathcal{C}^{\text{R|H}}_t$, and $\mathcal{C}^{\text{H}}_{<t}$ available during window t
end for

Algorithm 2 Novelty-Weighted Reservoir Sampling (NWRS)

Require: Earlier historical cache $\mathcal{C}^{\text{H}}_{<t-1}$; post-ViT embeddings $\mathcal{R}^{\text{low}}_{t-1}$; aligned candidate KV blocks $\Delta\mathcal{C}^{\text{H}}_{t-1}$; persistent $\mathcal{M}$ and $\mathcal{N}$; budget $B_{s(t)}$; bias $\beta > 1$

Ensure: Historical cache $\mathcal{C}^{\text{H}}_{<t}$ for current window t; updated $\mathcal{M}$ and $\mathcal{N}$

for each aligned pair $(\mathcal{R}^{\text{low}}_{t-1,i}, \Delta\mathcal{C}^{\text{H}}_{t-1,i})$, in temporal order **do**
  $\mathbf{z}_i \leftarrow \text{Normalize}_{\ell_2}(\text{MeanPool}(\mathcal{R}^{\text{low}}_{t-1,i}))$
  $\mathcal{N}_i \leftarrow$ visual embeddings currently stored in the local reference buffer $\mathcal{N}$
  $\mathcal{H}_{i-1} \leftarrow \{\mathbf{z}(m) : m \in \mathcal{M}\}$ ▷ Embeddings paired with retained KV blocks
  $n^{\text{loc}}_i \leftarrow 1 - \max_{\mathbf{z}_j \in \mathcal{N}_i} \mathbf{z}_i^\top \mathbf{z}_j$
  $n^{\text{glob}}_i \leftarrow 1 - \max_{\mathbf{z}_j \in \mathcal{H}_{i-1}} \mathbf{z}_i^\top \mathbf{z}_j$
  For an empty reference set, set the corresponding novelty to 1.
  $s_i \leftarrow 1 - (1 - n^{\text{loc}}_i)(1 - n^{\text{glob}}_i)$; $\omega_i \leftarrow \beta^{s_i}$
  Draw $U_i \sim \text{Uniform}(0, 1)$ once; $p_i \leftarrow -\ln U_i / \omega_i$ ▷ Keep p_i fixed
  Insert $(i, \mathbf{z}_i, \Delta\mathcal{C}^{\text{H}}_{t-1,i}, \text{timestamp}_i, p_i)$ into $\mathcal{M}$
  while $\sum_{m \in \mathcal{M}} \text{KVSize}(m) > B_{s(t)}$ **do**
    Remove $\arg\max_{m \in \mathcal{M}} p(m)$ from $\mathcal{M}$
  end while
  Append $\mathbf{z}_i$ to $\mathcal{N}$ and discard embeddings outside the local reference window
end for
Sort $\mathcal{M}$ by timestamp; $\mathcal{C}^{\text{H}}_{<t} \leftarrow \text{Concat}(\text{KV blocks in } \mathcal{M})$
Persist $\mathcal{M}$ and $\mathcal{N}$ for the next NWRS call
return $\mathcal{C}^{\text{H}}_{<t}$

F More results for Efficiency Analysis

Table 10 reports query-time context-expansion cost at 64, 256, 512, and 1024 frames. These frame counts cover different stages of time-adaptive capacity reallocation. Efficiency remains stable across them because the total context length is fixed and does not grow without bound as the input video becomes longer. When a question arrives, neither repeated prefill nor additional retrieval is required; CGCE only advances through prefetched context levels. As Table 10 shows, this overhead is small.

G Ablation of Time-Adaptive Capacity Reallocation

Table 11 compares the complete three-stage schedule with three fixed-stage configurations. Rather than fixing only the recent-frame cap r_s, each baseline takes one complete stage-wise allocation from

Table 10: **Query-time efficiency across input lengths.** L1 denotes answering at Level 1 without expansion. "TTFT" / "TPOT": Time to First Token / Time Per Output Token. Timing starts at query arrival and includes about 20 ms for History Shortcut.

Frames	GPU / GB ↓			TTFT / ms ↓			TPOT / ms ↓		
	L1	1→2	1→2→3	L1	1→2	1→2→3	L1	1→2	1→2→3
64	18.2	18.2	18.7	34.9	51.6	75.1	14.7	14.7	14.6
256	18.4	18.4	18.6	34.5	51.1	72.6	14.6	14.7	14.6
512	18.4	18.4	18.6	35.0	51.9	74.0	14.9	15.0	14.9
1024	18.2	18.2	19.0	34.4	53.5	80.2	15.0	15.1	15.2

Table 11: **Fixed (r_s, B_s) allocation pairs versus adaptive scheduling.** Each constant baseline applies one stage's recent-frame cap r_s and historical-cache budget B_s throughout the entire stream. Best results are highlighted in bold.

Model	OVO-Bench Avg.	StreamingBench Avg.	MVBench Avg.	MLVU M-Avg.	Video-MME Long	Video-MME Avg.
+ Const-256-10K	66.3	79.4	63.1	**66.9**	53.1	**62.6**
+ Const-512-8K	66.6	**81.0**	63.6	65.2	53.4	61.9
+ Const-768-6K	66.5	80.8	64.0	63.7	51.8	60.7
+ **AutoStream**	**67.1**	**81.0**	**64.1**	66.2	**53.6**	62.0

the adaptive schedule and applies the same (r_s, B_s) pair throughout the entire stream: $(768, 6\text{K})$, $(512, 8\text{K})$, or $(256, 10\text{K})$. Thus, both the per-frame visual-token cap for recent observations and the total historical-cache budget remain constant over video time, while the overall context quota stays at 12K tokens. The fixed $(256, 10\text{K})$ allocation favors long-range memory, yielding 66.9 on MLVU and 62.6 on Video-MME Avg., but continuously compresses recent observations and reduces OVO-Bench, StreamingBench, and MVBench to 66.3, 79.4, and 63.1, respectively. Conversely, the fixed $(768, 6\text{K})$ allocation retains richer recent detail but limits historical capacity, with MLVU and Video-MME Long dropping to 63.7 and 51.8. The fixed $(512, 8\text{K})$ allocation provides an intermediate compromise, but a single allocation pair cannot accommodate the different capacity requirements of short and long streams.

In contrast, Time-Adaptive Capacity Reallocation switches the complete (r_s, B_s) pair from $(768, 6\text{K})$ to $(512, 8\text{K})$ and then $(256, 10\text{K})$ as the stream grows. It therefore preserves high-resolution recent perception early in the stream and gradually extends historical coverage at later stages. AutoStream achieves the best results on OVO-Bench, MVBench, and Video-MME Long (67.1, 64.1, and 53.6, respectively), while tying the best StreamingBench score at 81.0. Although the fixed $(256, 10\text{K})$ configuration remains slightly better on individual long-video aggregate metrics, the adaptive configuration attains the highest simple average across all six reported metrics. These results show that the gain does not arise from any single fixed allocation pair, but from jointly adapting recent detail and historical coverage over video time.

H Limitations and Future Work

AutoStream achieves state-of-the-art performance in streaming video understanding and further advances the performance of history-on-demand methods. Nevertheless, its current design has several limitations that leave room for future research. Existing history-on-demand methods, including AutoStream, primarily manage historical context along the temporal dimension, with video frames as the basic unit. Some tasks, particularly those involving spatial understanding, may instead require managing the KV states of different visual regions within a frame. Developing finer-grained history-on-demand mechanisms that jointly consider temporal and spatial information is therefore a promising direction. In addition, an increasing number of training-based methods have explored enabling models to respond proactively as a video stream unfolds (Xia et al., 2026; Yao et al., 2026; Yang et al., 2026b; Xie et al., 2026a). Extending history-on-demand strategies to such proactive-response settings represents another valuable direction for future work.