DUALSTREAM: EPISODIC FEEDBACK SHAPES VISUAL MEMORY FOR LONG-HORIZON STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models commonly construct video context offline from a complete video. Streaming video understanding instead requires context to be built under resource constraints as frames arrive, often before the user question is known. Many streaming methods update this context based on visual redundancy or local changes, but visual novelty alone does not determine what is worth retaining. The importance of incoming details also depends on earlier events and can change as the episode unfolds. To address this gap, we present DualStream, a training-free framework that uses evolving textual episodic memory to guide visual retention. The visual stream maintains a compact token pool, which the textual stream periodically reads to update a running summary of the observed video. Each update also produces a future focus statement that guides the ranking of visual tokens from subsequent frames. Newly retained evidence then informs the next episodic update, forming a closed loop that adapts visual selection as the video develops. On the long subset of Video-MME, DualStream reaches 60.0% accuracy, exceeding the strongest online baseline by 4.7 percentage points while dropping 86.1% of visual tokens. Beyond these gains, at the same visual token budget, DualStream outperforms both visual memory alone and combined visual and textual memory without feedback.

1 INTRODUCTION

Vision-language models (VLMs) are commonly used with offline access to a complete video (Bai et al., 2025; Chen et al., 2025c; GLM-V Team, 2026; Li et al., 2025), as illustrated in Figure 1(a). In real-world scenarios with continuous visual input, however, models interpret video as it unfolds, a task known as streaming video understanding. In this setting, context is built incrementally, and users may ask questions before the video ends. Retaining every visual token would cause the context to grow rapidly and eventually exceed the available budget. The model must therefore decide which incoming details to preserve, often before knowing what a future question will require.

Existing streaming methods construct compact visual context through caching, memory organization, and token selection (Chen et al., 2024; Zhang et al., 2026a; Yao et al., 2025; Yang et al., 2025b), as illustrated in Figure 1(b). These mechanisms preserve visual evidence for later use as new observations arrive. Other approaches maintain summaries or reasoning traces of observed events to support retrieval, ongoing reasoning, and memory compression (Liang et al., 2026; Guan et al., 2026; Liu et al., 2026). Together, these approaches make past observations available in visual or textual form. However, the understanding recorded in a textual state is not explicitly used to guide which visual evidence is retained from subsequent frames.

This limitation matters because the relevance of an incoming detail depends partly on what has happened so far. Even before a user question is known, the developing episode can indicate which objects, actions, or interactions warrant continued attention. As events unfold, those priorities may change: new evidence can alter the interpretation of earlier observations and shift what is useful to retain next. Visual selection therefore calls for an evolving guide that reflects the observed episode and is itself revised by newly retained evidence.

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.

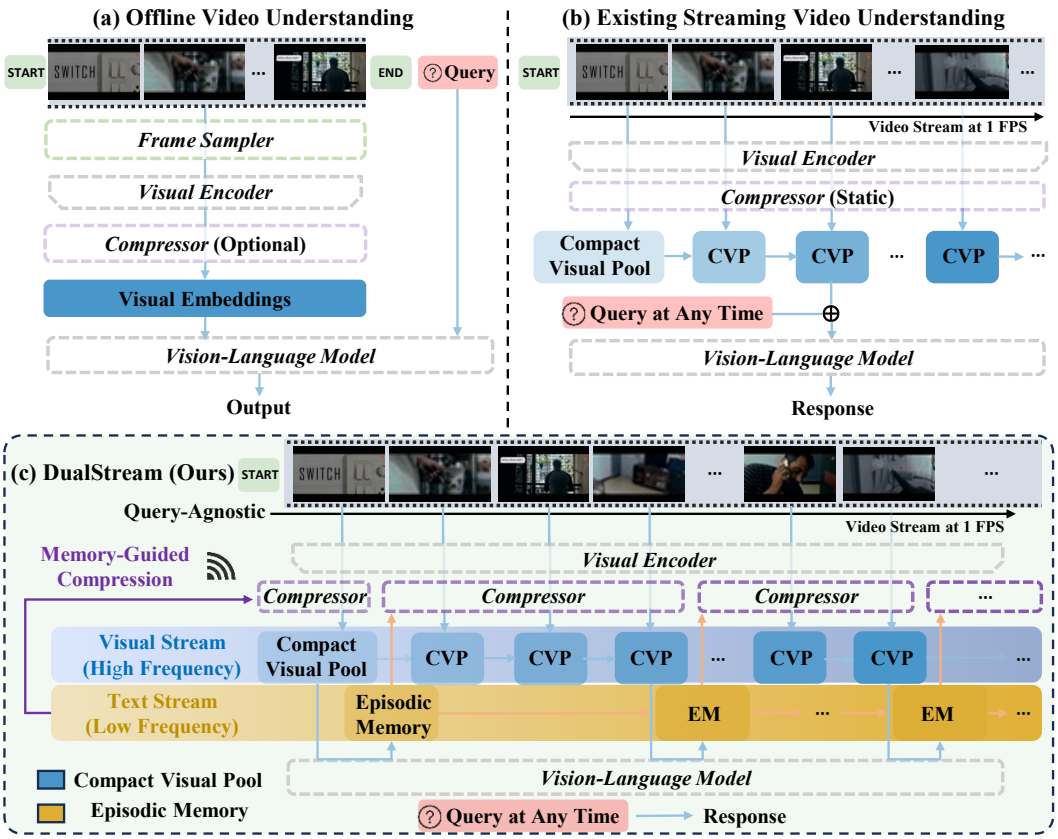

Figure 1: **Overview of DualStream and context construction in offline and streaming video understanding**. DualStream maintains a **C**ompact **V**isual **P**ool (CVP) alongside evolving **E**pisodic **M**emory (EM). EM guides later visual selection, and updated visual evidence refreshes EM.

We introduce DualStream, a training-free framework that uses **episodic feedback** to guide visual memory construction, as illustrated in Figure 1(c). Episodic feedback forms a loop between visual selection and an evolving textual account of the video: the account guides which incoming evidence to retain, and the retained evidence updates the account. Using a frozen VLM backbone, DualStream selects visual tokens as frames arrive and updates textual episodic memory less frequently. Each update combines retained evidence with the previous episodic state to revise a running summary and generate a future focus. The summary captures the observed episode, while the focus guides visual selection until the next update. When a question arrives, the system answers from its maintained memory state.

In end-of-stream question answering on four benchmarks, DualStream achieves the highest accuracy across all settings in our comparison. On the 30–60 minute Video-MME (Fu et al., 2025) long subset, it reaches 60.0% accuracy, exceeding the strongest online baseline by 4.7 percentage points while dropping 86.1% of visual tokens. Ablations with matched visual-token budgets show that feedback improves accuracy over visual memory alone and combined visual and textual memory without interaction. We also assess compression sensitivity and runtime, and evaluate question answering at intermediate timestamps on two additional benchmarks. Our main contributions are as follows:

- We propose **DualStream**, a training-free framework that turns evolving episodic understanding into guidance for visual memory construction before a user question is known.
- We demonstrate the benefit of episodic feedback through ablations with matched budgets and provide a detailed analysis of retained evidence.
- We evaluate DualStream under both end-of-stream and timestamped query settings, showing improved accuracy on long video streams with acceptable runtime cost.

2 Related Work

2.1 Long Video Understanding

Understanding long videos requires preserving evidence spread across time without excessive growth in visual context. Methods address this by compressing visual tokens and adapting their budgets (Song et al., 2024; Qi et al., 2026), or by selecting content through text-conditioned pruning and frame sampling (Cai et al., 2025; Yu et al., 2026; Tang et al., 2025; Zhang et al., 2025b; Tang et al., 2026; Tan et al., 2026). Memory-based and agentic approaches (Jin et al., 2025; Du et al., 2026; Wang et al., 2026b; Li et al., 2026) bring together observations across video segments to assemble evidence for reasoning. These approaches offer useful strategies for visual compression and memory, but those requiring known questions or complete videos are difficult to apply directly in a question-agnostic streaming setting.

2.2 Streaming Video Understanding

Streaming video understanding requires models to build context incrementally as frames arrive (Qian et al., 2025; Xia et al., 2026). As observations accumulate, streaming methods must decide how to represent and preserve visual history. Existing approaches use KV caching and retrieval (Chen et al., 2024; Di et al., 2025; Chen et al., 2025b) or hierarchical memory (Zhang et al., 2026a). To slow context growth, other approaches retain salient evidence (Huang et al., 2025; Zhang et al., 2025a; Zeng et al., 2025) or prune redundant tokens (Wang et al., 2026a). StreamAgent (Yang et al., 2025a) and QueryStream (Zhang et al., 2026b) use the actual user question to guide perception. In query-agnostic settings, TimeChat-Online (Yao et al., 2025) introduces Differential Token Drop (DTD) to remove temporally redundant tokens from incoming frames. StreamMem (Yang et al., 2025b) compresses video KV using attention from chat-template tokens as a generic query proxy. rLiVS (Dorovatas et al., 2025) selects tokens from each clip using attention scores obtained while captioning the same clip. These selection signals do not explicitly incorporate the evolving understanding of the episode, even though this understanding can shape how subsequent observations are interpreted.

Complementary to visual memory, textual memory provides a semantic account of the stream for retrieval and continued reasoning. M3-Agent (Long et al., 2025) continually builds episodic and semantic memories, and OASIS (Liang et al., 2026) links event summaries to visual evidence in a hierarchy for on-demand retrieval. VST (Guan et al., 2026) and ThinkStream (Liu et al., 2026) integrate ongoing reasoning into streaming video understanding and retain the resulting traces as textual memory. Unlike DualStream, these methods do not feed their evolving semantic state back into visual selection for subsequent frames.

3 Method

DualStream bridges visual and textual memory with an episodic feedback loop. It maintains a compact visual memory of selected tokens and a less frequently updated textual episodic memory. The episodic memory summarizes the observed video and guides subsequent visual selection, while retained visual evidence informs the next episodic update. The framework is training-free and constructs both memories before a user question is known.

3.1 Task Formulation and Overview

Question-agnostic memory construction. Let a video stream sampled every Δ seconds be $F = (f_1, f_2, \ldots)$, where f_t is the frame received at time step t. The 1 fps setting corresponds to $\Delta = 1$ second. The model processes the stream sequentially, updating its state at time step t after receiving f_t. A user question q may arrive at any time, and the model must answer from the current state without accessing future frames.

Each frame contributes visual tokens to the model context. Retaining all historical tokens would cause the visual state to grow rapidly and soon become impractical. DualStream limits this growth by retaining useful evidence and discarding redundant visual content. Its state after step t is $S_t = (V_t, E_t)$, where V_t is the compact visual pool (CVP) containing retained visual tokens with their original spatiotemporal positions, and E_t is the episodic memory (EM). It is a lightweight textual

state $E_t = (m_t, g_t)$, combining a running summary m_t of the observed video with a future focus g_t that guides later visual selection. Both memories are constructed without access to the user question.

Episodic feedback. The visual stream updates its memory at each frame, while the episodic stream updates at lower-frequency observation anchors:

$$V_t = \mathrm{Select}(V_{t-1}, f_t, E_{t-1}),$$
$$E_t = \mathrm{Observe}(E_{t-1}, V_t).$$

If step t is not an observation anchor, $E_t = E_{t-1}$. The previous episodic state E_{t-1} guides which tokens from f_t enter V_t. At an observation anchor, the resulting V_t in turn informs the update of E_t. Figure 2 illustrates how the updated focus guides visual selection in the following stream segment, carrying forward the model's revised understanding of the episode. As the video develops, selection can therefore shift toward different objects, actions, or interactions. This feedback is important in long, redundant videos, where memory cannot retain every visual token and the relevance of evidence often depends on the evolving episode rather than on local frame changes alone.

Question answering. For a question q at time t, DualStream produces an answer from its current memory state:

$$\hat{y} = \mathrm{Answer}(q, V_t, E_t).$$

Because memory construction is question-agnostic, the same stored context can support multiple questions about the observed stream.

3.2 Feedback-Guided Visual Retention

DualStream separates visual retention into two stages. We first filter temporally redundant tokens using Differential Token Drop (DTD), introduced by TimeChat-Online (Yao et al., 2025), and then use the current future focus to prioritize the remaining candidates according to the evolving episode.

Visual tokens and temporal filtering. We represent the accumulated visual memory and the tokens encoded from the current frame f_t as

$$V_t = \{(z_i, p_i)\}_{i=1}^{N_t},$$
$$Z_t = \{(z_{t,i}, p_{t,i})\}_{i=1}^{M_t}.$$

Here N_t and M_t are the numbers of tokens in memory and in the current frame, respectively. The embeddings z_i and $z_{t,i}$ are projected visual features from the VLM backbone. Their positions record the original temporal, height, and width coordinates, as in $p_i = (\tau_i, h_i, w_i)$, preserving token identity and ordering during selection. Appendix D describes their use with each backbone's position encoding. At step t, visual selection chooses a subset of Z_t to append to the existing memory V_{t-1}, producing V_t.

During long video streams, tokens at the same spatial locations often remain similar across adjacent frames. DTD identifies such temporal redundancy by comparing spatially aligned tokens in adjacent frames. A token is treated as temporally redundant with respect to the preceding frame when its cosine similarity exceeds τ_d:

$$r_{t,i} = \cos(z_{t,i}, z_{t-1,i}),$$
$$R_t = \{i : r_{t,i} > \tau_d\}.$$

The remaining indices form the candidate set $C_t = \{i : i \notin R_t\}$. DTD filters local temporal redundancy, but the remaining candidates can still differ in their relevance to the developing episode. We therefore use the future focus to prioritize these candidates.

Focus-guided retention. The latest future focus g_{t-1} expresses what to attend to as the episode develops. We use it as the selection query q_t^{sel} for ranking the DTD candidates. Before the first episodic update, the generic query "*What is happening in this clip?*" is used instead. Figure 2 illustrates how the focus generated at an observation anchor guides token ranking in the following stream segment.

To translate this textual guidance into priorities over visual tokens, we derive a relevance score from the frozen VLM's text-to-visual attention. Let $Q(q_t^{\mathrm{sel}})$ denote the query's text tokens and

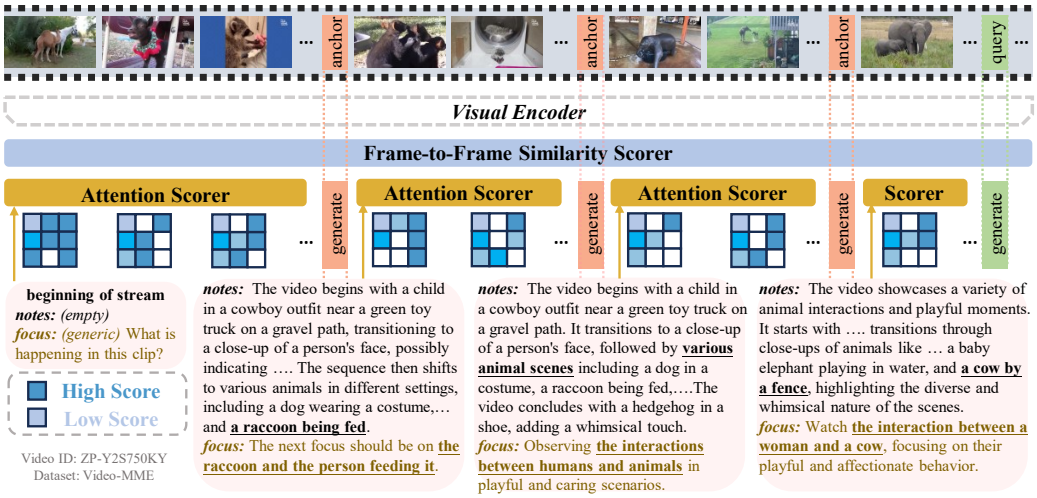

Figure 2: **Example of memory-guided visual selection**. At the i-th observation anchor, retained visual evidence updates the running summary m_i and future focus g_i. The focus g_i then serves as the text query for attention-based ranking of later visual tokens. Darker cells denote higher attention scores.

$\mathrm{Attn}_\theta^{(\ell,h)}(u,i)$ the attention weight from text token u to visual token i in head h of decoder layer ℓ. Averaging these weights over query tokens and heads gives each candidate $i \in C_t$ a score

$$a_{t,i} = \mathrm{Mean}_{u \in Q(q_t^{\mathrm{sel}}),h}\, \mathrm{Attn}_\theta^{(\ell,h)}(u,i).$$

The comparison between textual focus and visual candidates draws on the backbone's vision-language alignment. This alignment tends to be stronger in middle decoder layers than in very shallow or final layers, motivating the choice of a middle layer ℓ for attention extraction (Jiang et al., 2024; Kaduri et al., 2024; He et al., 2025; Chen et al., 2025a).

The resulting scores determine the candidates' retention priorities, while ρ controls the fraction of C_t that enters memory. The indices of the highest-scoring candidates form K_t, and their embeddings and positions are appended to V_{t-1}:

$$\begin{aligned} K_t &= \mathrm{Top}_{\rho|C_t|}\{a_{t,i} : i \in C_t\}, \\ V_t &= \mathrm{Select}(V_{t-1}, f_t, E_{t-1}) \\ &= V_{t-1} \oplus \{(z_{t,i}, p_{t,i}) : i \in K_t\}. \end{aligned}$$

Here $\oplus$ denotes appending the selected tokens in their original order, after any required capacity maintenance detailed in Appendix A.2. Keeping the projected embeddings in V_t makes the retained evidence reusable at answer time, avoiding a new visual encoding pass over the observed history.

3.3 Evolving Episodic Memory

Running summary and future focus. The retained visual evidence provides the basis for an evolving account of the video. To relate new observations to what has already happened, the VLM backbone reads the updated visual memory together with the previous episodic state and produces a running summary m_t and a future focus g_t:

$$E_t = (m_t, g_t) = \mathrm{Observe}(E_{t-1}, V_t).$$

The running summary integrates new observations into a concise account of the video so far. Each revision replaces the previous summary while preserving relevant history in compressed form.

While the summary records what has been observed, the future focus turns the current episodic understanding into guidance for subsequent visual selection. It specifies what to attend to next in a short, visually actionable sentence. Together, these textual components make the system's interpretation of the observed video explicit before a question arrives. Appendix E.1 lists the observation prompt.

Table 1: **End-of-stream long-video QA accuracy**. We implement SimpleStream using Qwen2.5-VL-7B. Results for other baselines are reported under the original protocol of each method. The best result is bold and the second best is underlined.

Method	Video-MME$_{\text{w/o sub.}}$		LVBench	MLVU	LongVideoBench
	Long	Overall		MC	val
Video Length	30–60 min	11 s–60 min	0.5–2 h	3 min–2 h	8 s–1 h
Offline Backbone					
Qwen2.5-VL-7B (Bai et al., 2025)	53.9	<u>65.1</u>	<u>45.3</u>	<u>70.2</u>	56.0
Recent-Window Control					
SimpleStream (4 frames) (Shen et al., 2026)	38.9	42.2	26.4	49.2	44.4
Open-source Online Models					
VideoLLM-online-8B (Chen et al., 2024)	–	26.9	24.0	35.2	–
Flash-VStream-7B (Zhang et al., 2025a)	50.3	61.2	42.0	66.3	–
Dispider-7B (Qian et al., 2025)	–	57.2	–	61.7	–
VideoChat-Online-7B (Huang et al., 2025)	47.1	54.4	–	60.8	54.1
TimeChat-Online-7B (Yao et al., 2025)	49.2	62.5	–	65.4	57.7
StreamForest-7B (Zeng et al., 2025)	–	61.4	–	70.0	–
HERMES$_{\text{on Qwen2.5-VL-7B}}$ (Zhang et al., 2026a)	54.4	62.0	–	–	–
VST-7B (Guan et al., 2026)	<u>55.3</u>	64.9	–	–	<u>58.0</u>
WAT-7B (Han et al., 2026)	50.8	62.4	–	63.9	–
DualStream$_{\text{on Qwen2.5-VL-7B}}$ (ours)	**60.0**	**67.3**	**48.6**	**70.4**	**59.6**

Sparse updates and causal timing. This semantic account is revised at sparse observation anchors, combining evidence accumulated through frequent visual updates. The anchors occur roughly every few minutes. Their boundaries are adjusted using recently observed visual changes to align semantic updates with content transitions when possible. Appendix A.1 details the anchor construction.

At each anchor, the current step's retained tokens first enter visual memory so that the episodic update can incorporate the latest evidence. Any visual selection that occurs while this update is being generated continues to use the most recently completed focus statement. Once the new summary and focus are available, they replace the previous episodic state, and the revised focus guides subsequent selection. The visual stream thus supplies evidence for each semantic revision, which in turn shapes what the visual stream retains next.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate DualStream on various video understanding benchmarks, processing videos in temporal order and building memories without access to the question. On Video-MME (Fu et al., 2025), LVBench (Wang et al., 2025), MLVU (Zhou et al., 2025), and LongVideoBench (Wu et al., 2024), we follow recent streaming methods (Yao et al., 2025; Zeng et al., 2025; Guan et al., 2026) and introduce each question after the final observation. This setting tests the model's ability to retain relevant information until the end of the video. We also evaluate on OVO-Bench (Niu et al., 2025) and StreamingBench (Lin et al., 2024), where questions arrive at intermediate timestamps. We report multiple-choice accuracy under each benchmark's standard scoring protocol.

Baselines. We compare DualStream with the offline backbone Qwen2.5-VL-7B and recent online video understanding models in Tables 1 and 5. Recent work (Shen et al., 2026) shows that a VLM using only recent frames can match or outperform published streaming models on OVO-Bench and StreamingBench. We therefore include SimpleStream, which answers from the latest four observed frames, as a control baseline to assess the value of retaining earlier visual information. We reproduce this baseline with the same frozen Qwen2.5-VL-7B for the end-of-stream comparison in Table 1 and

use its published results for timestamped questions in Table 5. All other baseline values come from their respective papers.

Implementation Details. Our main experiments use a frozen Qwen2.5-VL-7B (Bai et al., 2025) as the VLM backbone to facilitate comparison with prior work. We also evaluate DualStream on InternVL3-8B (Zhu et al., 2025) in Section 4.4, with adaptation and calibration details in Appendix D. We extract text-to-visual attention scores from the 0-indexed decoder layer L14 of Qwen2.5-VL, which has 28 decoder layers in total. Appendix A.3 describes the label-free probe used to select this layer. Our implementation follows the public TimeChat-Online (Yao et al., 2025) codebase for DTD and streaming inference, with modifications for episodic updates and focus-guided selection. DualStream uses an initial sampling rate of 1 fps, following common practice in recent online video understanding systems (Yao et al., 2025; Zeng et al., 2025). For long streams, the sampling rate is progressively adjusted to keep the number of resident frames bounded, as detailed in Appendix A.2. Following the original paper of DTD algorithm (Yao et al., 2025), we set the DTD threshold to $\tau_d = 0.25$ and choose the attention keep fraction $\rho = 0.55$ to target an overall token drop rate of approximately 85%. Appendix A.3 provides further details on hyperparameter selection, and Section 4.4 presents a sensitivity analysis of these hyperparameters. Episodic memory is updated at sparse observation anchors based on elapsed stream time and nearby scene changes, as detailed in Appendix A.1. All evaluations use deterministic decoding.

4.2 Results on End-of-Stream Long-Video QA

Table 1 reports the results for end-of-stream long-video QA. DualStream achieves the highest accuracy across all five settings with an average gain of 3.1 percentage points over the strongest online baseline in each column. On the long subset of Video-MME, DualStream reaches 60.0%, exceeding the offline backbone by 6.1 points and the strongest online baseline by 4.7 points while dropping 86.1% of visual tokens. These results show that the compact memory retains useful information over long video streams.

In comparison, the control baseline SimpleStream achieves 42.2% on Video-MME overall and drops to 38.9% on Video-MME long, where it trails DualStream by 21.1 points. This gap indicates that recent frames provide limited evidence for questions that require information accumulated over a long video.

4.3 Mechanism Analysis

Table 2: **Ablations on Video-MME long**. T, V, and Inter. denote text memory, visual memory, and closed-loop interaction.

Variant	T	V	Inter.	Accuracy	Token Drop
Text-only	✓			46.9%	–
Visual-only		✓		57.8%	86.1%
DualStream$_{\text{w/o inter.}}$	✓	✓		57.1%	86.1%
DualStream (ours)	✓	✓	✓	**60.0%**	86.1%

Effect of closed-loop feedback. Table 2 reports ablations of the two memories and their interaction on Video-MME long. A text-only reference based on the Socratic memory protocol (Long et al., 2025) reaches 46.9%, while our visual-only variant based on DTD (Yao et al., 2025) reaches 57.8%. This variant uses the same visual-memory pipeline and token budget as DualStream, but removes episodic memory and focus-guided selection. DualStream (w/o Inter.) combines visual and episodic memory but uses the fixed query "*What is happening in this clip?*" for visual selection throughout the video. It reaches 57.1%, showing that adding episodic text without using it to guide visual selection does not improve accuracy over visual memory alone.

Full DualStream reaches 60.0% accuracy at the same visual-token drop rate of 86.1%, significantly outperforming both the visual-only and no-interaction variants. This improvement shows the benefit of using the evolving future focus to guide subsequent visual selection. Appendix B.1 reports the statistical significance tests.

Evidence retention. To examine the role of future-focus guidance in evidence retention, we conduct a replay experiment. Before the question is available, we select equal numbers of tokens using future-focus guidance and visual-only DTD selection. Once both selections are fixed, we introduce the actual question and compare the retained attention mass, using retrospective question-to-visual attention as a proxy for token relevance. The attention mass is calculated within proxy-evidence bins, defined as

temporal bins that yield correct answers when evaluated in isolation. Across 382 StreamingBench source videos, future-focus guidance retains 19.86% of the question-attention mass within these bins, compared with 13.01% for DTD, a statistically significant relative increase of 52.7%. Appendix B.2 details the replay protocol and statistical analysis.

Case Study. Figure 2 illustrates how evolving episodic memory guides visual selection. As the video unfolds, the focus shifts from a raccoon being fed to a woman interacting with a cow. Each updated focus guides the ranking of subsequent visual tokens, adapting evidence retention to the developing episode. Appendix B.4 presents a contrasting case in which the focus emphasizes game strategy while the later question concerns the character's class. This mismatched emphasis may lower the priority of visual cues needed to identify the class. However, the focus provides a soft ranking signal, while ρ determines the fraction of DTD candidates retained. Tokens are selected by relative rank, so weak relevance to the focus does not automatically exclude them. Each focus guides selection until the next episodic update, when new observations can redirect it toward content more relevant to the evolving episode.

Focus statement audit. We audit generated future focus statements on Video-MME long using an LLM-as-a-Judge approach. Across 3,846 valid judge responses, visual actionability and forward-lookingness receive mean scores of 3.67 and 3.68 on a 1–5 scale, respectively. These results suggest that the self-generated focus statements are actionable rather than vague, providing guidance on which visual evidence to retain from subsequent frames. Appendix B.3 provides the audit procedure and full results.

4.4 Robustness and Efficiency

Table 3: **Runtime summary on Video-MME long**, with visual updates and CPU scene detection measured at 1 fps. At this rate, the components in the first two rows share a 1,000 ms per-frame processing budget. Appendix C.3 provides the profiling setup, runtime breakdown, timing boundaries, and further analysis.

Component	Mean (ms)	p95 (ms)
Scene detection and decoding	50.14	69.55
Complete visual-state update	625.95	885.45
Episodic memory generation	4503.31	6621.70
Query-to-answer completion	1544.44	1812.36

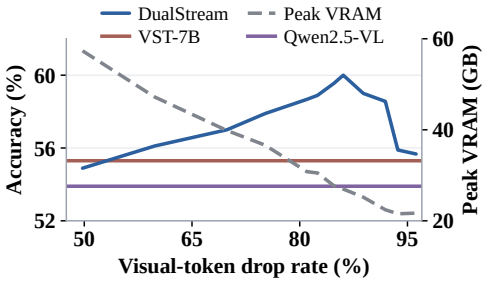

Figure 3: **Sensitivity of Video-MME long accuracy** to the compression hyperparameters τ_d and ρ. Appendix C.2 lists the tested hyperparameter combinations and corresponding token drop rates.

Compression sensitivity. DualStream's compression of the observed video stream is primarily controlled by the DTD similarity threshold τ_d and the attention keep fraction ρ. To assess sensitivity to these hyperparameters, we evaluate multiple combinations that yield overall visual-token drop rates ranging from approximately 50% to 96%. Figure 3 shows that Video-MME long accuracy stays between 58.6% and 60.0% across a broad range of compression settings as peak VRAM decreases, outperforming the strongest online and offline baselines. The default configuration ($\tau_d = 0.25, \rho = 0.55$) reaches 60.0% accuracy at 86.1% token drop. At extremely high compression rates, accuracy falls as more visual cues are removed. This inverted U-shaped trend reflects a trade-off between compression level and accuracy, while the broad accuracy plateau indicates that DualStream maintains robust performance across multiple hyperparameter settings.

Latency analysis. To characterize DualStream's streaming latency, we profile its main components on a single NVIDIA A100. Table 3 reports the results, and Appendix C.3 provides the profiling details. On Video-MME long, a complete visual update takes 0.626 s on average, with p95 and p99 latencies of 0.885 s and 0.970 s. These updates include input preprocessing, visual encoding and selection, position indexing, and memory updates. Thus, over 99% of the measured visual updates finish within the one-second interval of a 1-fps stream. The episodic stream updates less frequently. Observation anchors are 178.5 s apart on average, while generating an episodic update takes 4.503 s on average and 6.622 s at p95. Generation accounts for 2.46% of the covered stream time. If a

focus-guided selection step occurs during episodic generation, it uses the most recently committed focus statement instead of waiting for the new update.

Cross-backbone transfer. The core of DualStream is a feedback loop in which episodic memory guides visual retention and retained evidence informs subsequent episodic updates. This feedback mechanism itself is training-free and can be transferred to another VLM without training model parameters. Transfer requires adapting the visual interfaces to the target backbone, selecting a decoder layer for extracting text-to-visual attention, and tuning the hyperparameters τ_d and ρ to control the compression rate. Appendix D details this process using InternVL3-8B as an example. On the Video-MME long subset, DualStream improves accuracy over visual-only memory by 2.22 percentage points on Qwen2.5-VL-7B and 2.00 points on InternVL3-8B, as shown in Table 4, supporting the transferability of episodic guidance.

Table 4: **Cross-backbone results on Video-MME long**, with measured visual-token drop rates.

Backbone	Method	Accuracy	Token Drop
Qwen2.5-VL-7B	Visual-only	57.78%	86.06%
Qwen2.5-VL-7B	DualStream	60.00%	86.06%
InternVL3-8B	Visual-only	57.56%	86.47%
InternVL3-8B	DualStream	59.56%	87.90%

4.5 Timestamped Streaming Evaluation

We evaluate DualStream in a timestamped streaming setting on OVO-Bench and StreamingBench as a complementary experiment and report the results in Table 5. Baseline results are under each paper's native protocol. The final row reports DualStream's absolute gain over Qwen2.5-VL-7B. The recent method SimpleStream (Shen et al., 2026) achieves strong performance on both benchmarks, as shown in Table 5, using a simple strategy that answers questions from only a few recent video frames. This indicates that a few recent frames suffice for many timestamped questions, while DualStream is designed to retain information over long-horizon video streams. DualStream achieves 70.0% on OVO Real-Time, 52.4% on OVO Backward, and 77.3% on StreamingBench Real-Time, with an average gain of 7.3 percentage points over the Qwen2.5-VL backbone. These competitive results show that DualStream's accumulated memory supports questions focusing not only on earlier events but also on current content.

Table 5: **Timestamped streaming question-answering accuracy**. Best and second-best entries are bold and underlined.

Method	OVO-Bench		StreamingBench
	Real-Time	Backward	Real-Time
Offline Backbone			
Qwen2.5-VL-7B	59.9	44.7	73.3
Recent-Window Control			
SimpleStream (4 frames)	**78.4**	51.9	78.5
Open-source Online Models			
VideoLLM-online	20.8	17.7	36.0
Flash-VStream	29.9	25.4	23.2
Dispider	54.6	36.1	67.6
TimeChat-Online	58.6	42.0	73.6
StreamForest	61.2	52.0	77.3
Streamo	66.0	46.1	–
HERMES	68.4	48.1	<u>78.7</u>
VST	67.2	**56.7**	**79.5**
WAT	64.7	45.2	77.7
DualStream (ours)	<u>70.0</u>	<u>52.4</u>	77.3
Gain over backbone	**+10.1**	**+7.7**	**+4.0**

5 Conclusion

We presented DualStream, a training-free framework for streaming video understanding in which evolving episodic memory guides the construction of compact visual memory. Visual observations inform episodic updates, and each update produces a future focus for subsequent visual selection. Experiments under both end-of-stream and timestamped query settings show strong question-answering performance. On Video-MME long, DualStream reaches 60.0% accuracy while dropping 86.1% of visual tokens. Ablations show the benefit of feedback over visual memory alone and a combination of the two memories without interaction. Future work could explore how to balance focus-guided retention with broader evidence coverage for unforeseen questions.

Ethical Considerations

We evaluate DualStream only on public video-understanding benchmarks and do not collect user data or train on private videos. Streaming video understanding systems could be applied to privacy-sensitive content and may inherit biases, hallucinations, or demographic disparities from the underlying VLM. The generated episodic memory and future focus may omit, distort, or overemphasize observed events. They should not be treated as authoritative records or used for high-stakes decisions without human review. Deployments should minimize unnecessary retention of raw video and generated memory and follow applicable requirements for consent, privacy, data retention, and model or data licensing.

AI Use Statement

We used generative AI tools to support literature search and summarization, provide feedback on experimental design, discuss the analysis and presentation of experimental results, and assist with drafting and editing parts of the manuscript. AI-assisted numerical and literature claims were checked against raw experimental outputs and primary sources. We did not use generative AI to generate or relabel benchmark data. Generative models used as components of DualStream or for the disclosed LLM-based evaluation are described separately in the paper. The authors take responsibility for the final content, claims, experiments, and reported results.

Reproducibility Statement

The main text specifies DualStream's state, update rules, selection algorithm, evaluation protocol, and default configuration. The appendix details the construction of observation anchors, attention-layer selection, backbone adaptation and calibration, and prompts. Reported evaluations use public benchmarks, released frozen backbones, and deterministic decoding.

References

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, 2025. URL https://arxiv.org/abs/2502.13923.

Kaitong Cai, Jusheng Zhang, Jing Yang, Yijia Fan, Pengtao Xie, Jian Wang, and Keze Wang. FlashVLM: Text-guided visual token selection for large multimodal models, 2025. URL https://arxiv.org/abs/2512.20561.

Brandon Castellano. PySceneDetect. Software, 2014. URL https://www.scenedetect.com.

Haoran Chen, Junyan Lin, Xinghao Chen, Yue Fan, Jianfeng Dong, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. Multimodal language models see better when they look shallower. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6677–6695. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.emnlp-main.339. URL https://aclanthology.org/2025.emnlp-main.339/.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_VideoLLM-online_Online_Video_Large_Language_Model_for_Streaming_Video_CVPR_2024_paper.html.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, Ming Lu, and Shanghang Zhang. StreamKV: Streaming video question-answering with segment-based KV cache retrieval and compression, 2025b. URL https://arxiv.org/abs/2511.07278.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025c. URL https://arxiv.org/abs/2412.05271.

DeepSeek-AI. DeepSeek V4 Preview Release, 2026. URL https://api-docs.deepseek.com/news/news260424.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8g9fs6mdEG.

Evangelos Dorovatas, Soroush Seifi, Gunshi Gupta, and Rahaf Aljundi. Recurrent attention-based token selection for efficient streaming video-LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4825. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d4857f724cc1af4c8f1e18032426aa2e-Abstract-Conference.html.

Henghui Du, Chunjie Zhang, Xi Chen, Chang Zhou, and Di Hu. Video detective: Seek critical clues recurrently to answer question from long videos, 2026. URL https://arxiv.org/abs/2512.17229.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

GLM-V Team. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026. URL https://arxiv.org/abs/2507.01006.

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.

Zifan Han, Hongbo Sun, Jinglin Xu, Canhui Tang, Yulong Lei, Xuchong Zhang, Hongbin Sun, Zhongjiang He, and Hao Sun. WAT: Online video understanding needs watching before thinking, 2026. URL https://arxiv.org/abs/2603.13412.

Zoe Wanying He, Sean Trott, and Meenakshi Khosla. Seeing through words, speaking through pixels: Deep representational alignment between vision and language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35657–35672. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1806. URL https://aclanthology.org/2025.emnlp-main.1806/.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: OVBench and videochat-online. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Huang_Online_Video_Understanding_OVBench_and_VideoChat-Online_CVPR_2025_paper.html.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens, 2024. URL https://arxiv.org/abs/2411.16724.

Hongbo Jin, Qingyuan Wang, Wenhao Zhang, Yang Liu, and Sijie Cheng. Videomem: Enhancing ultra-long video understanding via adaptive memory management, 2025. URL https://arxiv.org/abs/2512.04540.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models, 2024. URL https://arxiv.org/abs/2411.17491.

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. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=zKv8qULV6n.

Keliang Li, Yansong Li, Hongze Shen, Mengdi Liu, Hong Chang, and Shiguang Shan. Lenswalk: Agentic video understanding by planning how you see in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19518–19528, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_LensWalk_Agentic_Video_Understanding_by_Planning_How_You_See_in_CVPR_2026_paper.html.

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.

Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video, 2026. URL https://arxiv.org/abs/2603.12938.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory, 2025. URL https://arxiv.org/abs/2508.09736.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How far is your video-LLMs from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Niu_OVO-Bench_How_Far_is_Your_Video-LLMs_from_Real-World_Online_Video_CVPR_2025_paper.html.

Haozhe Qi, Kevin Qu, Mahdi Rad, Rui Wang, Alexander Mathis, and Marc Pollefeys. Adapttoken: Entropy-based adaptive token selection for MLLM long video understanding, 2026. URL https://arxiv.org/abs/2603.28696.

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.

Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. MovieChat+: Question-aware sparse memory for long video question answering, 2024. URL https://arxiv.org/abs/2404.17176.

Wenhui Tan, Xiaoyi Yu, Jiaze Li, Yijing Chen, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. MSJoE: Jointly evolving MLLM and sampler for efficient long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19487–19496, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Tan_MSJoE_Jointly_Evolving_MLLM_and_Sampler_for_Efficient_Long-Form_Video_CVPR_2026_paper.html.

Canhui Tang, Zifan Han, Hongbo Sun, Sanping Zhou, Xuchong Zhang, Xin Wei, Ye Yuan, Huayu Zhang, Jinglin Xu, and Hao Sun. TSPO: Temporal sampling policy optimization for long-form video language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (11):9368–9376, 2026. doi: 10.1609/aaai.v40i11.37896. URL https://ojs.aaai.org/index.php/AAAI/article/view/37896.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Tang_Adaptive_Keyframe_Sampling_for_Long_Video_Understanding_CVPR_2025_paper.html.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_LVBench_An_Extreme_Long_Video_Understanding_Benchmark_ICCV_2025_paper.html.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression, 2026a. URL https://arxiv.org/abs/2512.00891.

Ziyang Wang, Honglu Zhou, Shijie Wang, Junnan Li, Caiming Xiong, Silvio Savarese, Mohit Bansal, Michael S. Ryoo, and Juan Carlos Niebles. Active video perception: Iterative evidence seeking for agentic long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9088–9099, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026F/html/Wang_Active_Video_Perception_Iterative_Evidence_Seeking_for_Agentic_Long_Video_CVPRF_2026_paper.html.

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*, volume 37, pp. 28828–28857, 2024. doi: 10.52202/079017-0907. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31219–31229, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xia_Streaming_Video_Instruction_Tuning_CVPR_2026_paper.html.

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

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, 2025b. URL https://arxiv.org/abs/2508.15717.

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.

An Yu, Ting Yu Tsai, Zhenfei Zhang, Weiheng Lu, Felix X.-F. Ye, and Ming-Ching Chang. ReDiPrune: Relevance-diversity pre-projection token pruning for efficient multimodal LLMs, 2026. URL https://arxiv.org/abs/2603.24680.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. Streamforest: Efficient online video understanding with persistent event memory. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/

085713-2547. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6dd91fec726dbed8915a1fbadd91d1d2-Abstract-Conference.html.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_Flash-VStream_Efficient_Real-Time_Understanding_for_Long_Video_Streams_ICCV_2025_paper.html.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. HERMES: KV cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.381. URL https://aclanthology.org/2026.acl-long.381/.

Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. QueryStream: Advancing streaming video understanding with query-aware pruning and proactive response. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 6193–6208, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0b17d256cf1fe1cc084922a8c6b565b7-Paper-Conference.pdf.

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_Q-Frame_Query-aware_Frame_Selection_and_Multi-Resolution_Adaptation_for_Video-LLMs_ICCV_2025_paper.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhou_MLVU_Benchmarking_Multi-task_Long_Video_Understanding_CVPR_2025_paper.html.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

A Implementation and Evaluation Details

A.1 Observation Anchor Construction

DualStream updates its textual episodic memory at sparse observation anchors. Each non-final boundary is determined adaptively from elapsed stream time and recently observed scene changes, without access to future frames. Starting from the current time u, we set a target interval

$$L(u) = \min(\max(0.5u, 45), 300),$$

where times are measured in seconds. Because $L(u)$ depends only on the current stream time u, the target interval starts at a 45-second lower bound and gradually increases as the stream progresses, up to a maximum of 300 seconds. This schedule provides frequent observations early in the stream and spaces them farther apart as the observed history grows. This helps avoid too few observation anchors in short streams and too many in long streams. These bounds apply to the target interval. The actual anchor spacing also depends on scene cuts and the boundary rules below.

The target boundary is $u + L(u)$. We identify nearby content transitions using PySceneDetect's ContentDetector (Castellano, 2014), which detects scene changes from frame-to-frame content differences. We search the window immediately before the target boundary, with width $\min(0.5L(u), 30)$ seconds, and choose the first detected scene cut in that window. If no scene cut is found, we place a forced boundary at the target time. For ease of evaluation, a final interval shorter than 10 seconds is merged into the preceding interval.

For the Video-MME long subset, this procedure yields 4,146 anchors over 300 unique videos. Consecutive anchor boundaries are 178.5 seconds apart on average (median 187.9, range 11.6–308.5), and 3,341/4,146 intervals (80.6%) fall between 45 and 300 seconds. The average video duration is 41.1 minutes, with 13.8 anchors per video. Among all anchors, 3,266 are scene boundaries, 587 are forced boundaries, and 293 are end boundaries. Forced boundaries account for 14.2% of all anchors. Appendix C.1 compares this adaptive schedule with fixed updates every 180 seconds.

A.2 Progressive Frame Sampling Strategy

A streaming system needs a memory budget that remains fixed as the stream grows. The central contribution of DualStream is episodic feedback for selecting incoming visual evidence. This mechanism determines what enters memory, while deciding which existing evidence to retain, merge, or evict once an overall budget is reached is a separate design dimension. DualStream's feedback mechanism can be combined with different policies for managing that budget. We use a progressive frame sampling heuristic to approximate uniform temporal sampling of the observed stream under a fixed frame cap. This strategy is not a core contribution of DualStream.

Sampling algorithm. Let B denote the maximum number of frames represented in resident visual memory, and let Δ denote the temporal spacing of the resident frame grid at the start of each replacement phase. We begin with $\Delta = 1$ second, corresponding to the default rate of 1 fps. The algorithm assumes an even B, with $B = 900$ used in our experiments. Each frame arrives at a sampling time and is filtered by DualStream. Its retained tokens enter visual memory. Once B frames are resident, a replacement phase starts from a snapshot of their chronological order. We plan to retain alternating frames on a grid with spacing 2Δ and mark the intervening frames for eviction. The eviction order is fixed from this initial snapshot and is not recomputed from the changing pool after each replacement.

Subsequent frames are admitted at the future points of the same 2Δ grid. When a new sampled frame arrives, we first release the next marked old frame, then add the new frame. This keeps at most B frames resident throughout the replacement. After $B/2$ such updates, half of the original frames remain and half of the pool consists of newly arrived frames. At the end of each replacement phase, the retained frames form a uniform temporal grid with spacing 2Δ. We set $\Delta \leftarrow 2\Delta$ and begin another replacement phase.

The schedule depends on elapsed time and the current phase. A phase spreads the removal of old frames across successive arrivals, with one eviction per admitted frame. As the stream grows, the fixed pool covers a longer history at a coarser temporal resolution. This trades sampling density for temporal coverage, while episodic feedback continues to determine which visual details are retained within the admitted frames.

Local DTD reconstruction and reselection. Removing a frame can change the predecessor against which a surviving frame is compared. Its previous DTD mask may then exclude tokens that are no longer redundant under the new comparison. We therefore revisit frames whose predecessor changes within their original observation segment. DTD is recomputed for these frames. When no aligned predecessor is available, all tokens are eligible, following the boundary rule in Appendix A.3.

After updating these masks, we perform semantic reselection over the surviving eligible candidates in the affected observation segment. The ranking reuses the cached attention scores associated with the focus used when that segment was originally processed, and retains the top ρ fraction of eligible candidates. Historical attention scores provide an approximation to its relevance after the visual context changes. Reusing them avoids another attention-scoring pass over the reconstructed historical segment. The original token positions and observation-segment membership are preserved throughout this maintenance.

Local DTD reconstruction and reselection are triggered only after the frame cap B is reached in extremely long video streams. Local memory maintenance comprises evicting historical groups and releasing their tokens and candidate caches, recomputing DTD masks for surviving groups whose predecessors have changed, reselecting tokens using cached attention scores, and rebuilding the visual pool and its position and group indices. In our measurements, the total time for these operations is 29.15 ms on average, with a p95 latency of 55.23 ms.

Progressive memory maintenance

State. Resident frames, their selected tokens and the reference grid spacing Δ.

1. Wait until the next scheduled sample arrives.
2. If the pool is full, release the next old frame in the eviction queue together with its selected tokens.
3. Recompute affected DTD masks within the original observation segments, then reselect from surviving candidates using their historical attention scores.
4. Encode the new frame, apply DualStream selection, and store its selected tokens.
5. When the pool first fills, initialize a phase on the 2Δ grid. After all $B/2$ planned evictions in a phase, set $\Delta \leftarrow 2\Delta$ and initialize the next phase from the current pool.

Budget accounting. This algorithm places an explicit upper bound on the number of resident frames. In our setting, the limit is $B = 900$ frames. Combined with DualStream's token compression rate of approximately 85%, this corresponds to a memory budget equivalent to approximately 135 frames, which is manageable for most VLMs.

A.3 Attention Layer and Compression Parameters

Memory-guided visual selection in DualStream follows a two-stage rule. The first stage uses the Differential Token Drop (DTD) algorithm (Yao et al., 2025) to remove redundant tokens by comparing adjacent frames. The frozen VLM then ranks the remaining candidates by text-to-visual attention. At the first visual time step, or when no aligned token grid is available from the preceding frame, all tokens enter the DTD candidate set.

We select the attention layer using a text-sensitivity probe. For each candidate decoder layer, we compute visual-token attention score vectors under different focus statements on the same visual inputs and report their mean pairwise correlation r. Lower r indicates greater sensitivity of visual ranking to the focus statement. For Qwen2.5-VL-7B, Table A.1 reports probe results using visual inputs from multiple segments of a source video. The visual inputs are held fixed across the four focus statements.

The probe selects the 0-indexed layer L14, which has the lowest mean cross-focus correlation and lies in the middle of Qwen2.5-VL's 28 decoder layers. This result is consistent with prior layer-wise analyses showing that middle VLM layers carry important signals for visual processing and cross-modal information flow (Jiang et al., 2024; Kaduri et al., 2024).

Layer selection is specific to the backbone. When adapting DualStream to a new VLM, the text-sensitivity probe should be repeated to select its scoring layer instead of directly reusing L14.

The scoring procedure uses the frozen backbone's cross-modal attention without additional training. The scorer encodes the selection query, runs the VLM up to the selected decoder layer, and computes attention from query tokens to visual tokens. Each visual token receives a score equal to the mean attention probability over query tokens and attention heads. In the default setting, the selection query

Table A.1: Representative layers from the layer probe. Lower correlation indicates greater sensitivity of visual-token rankings to the focus statement.

Decoder Layer	Mean Cross-Focus Correlation r
6	0.9649
11	0.8108
14	0.5937
19	0.6574
27	0.9450

is the most recent future focus. Before any future focus has been generated, the generic query is "*What is happening in this clip?*"

During visual selection, L14 attention scores rank the DTD candidates, and DualStream keeps the top ρ fraction, where ρ is the keep fraction after DTD, set to 0.55 by default. Tokens removed by DTD are excluded from this ranking.

We set the DTD threshold to $\tau_d = 0.25$, following TimeChat-Online (Yao et al., 2025). The attention keep fraction is chosen to target an overall token drop rate of approximately 85% after both filtering stages, to match the approximate token-drop rate of the TimeChat-Online baseline, yielding the default $\rho = 0.55$.

A.4 Benchmarks and Evaluation Protocols

For the benchmarks introduced in Section 4.1, we follow the official evaluation protocols, with evaluation subsets and reported metrics chosen in line with prior work. We report our main benchmark results from a single evaluation run per configuration. For Video-MME, we include overall and long-subset accuracy without subtitles. We report accuracy on the multiple-choice tasks in MLVU and the validation split of LongVideoBench. For OVO-Bench, we evaluate Real-Time Visual Perception and Backward Tracing. For StreamingBench, we evaluate the Real-Time Visual Understanding subset using the 60-second context preceding each query, following the benchmark's main evaluation setting.

B Mechanism Analysis

B.1 Statistical Comparisons for the Feedback Ablation

We compare the variants in Table 2 on the same 900 questions from 300 Video-MME long videos, pairing predictions by question. Each variant is evaluated with greedy decoding (`do_sample=False`), yielding a fixed set of predictions for the paired analysis. We estimate 95% percentile confidence intervals using 100,000 bootstrap samples of source videos, keeping all questions from each sampled video together. We also report two-sided exact sign tests over discordant question pairs. A win means that the first variant answers correctly and the second incorrectly. A loss is the reverse.

Table B.1: Paired comparisons on Video-MME long. Differences and confidence intervals are in percentage points. Confidence intervals are clustered by video.

Comparison	Difference	95% CI	Wins/losses	p
Full vs. w/o interaction	$+2.89$	$[0.67, 5.11]$	63/37	0.012
Full vs. visual-only	$+2.22$	$[0.56, 3.89]$	41/21	0.015
W/o interaction vs. visual-only	-0.67	$[-2.89, 1.56]$	49/55	0.624

Full DualStream significantly outperforms both controls. Both confidence intervals exclude zero, and both exact sign tests have $p < 0.05$. The difference between the two controls is not statistically significant.

Table B.2: LLM-as-a-Judge audit of future focus on the Video-MME long subset. The four dimensions measure whether each guidance sentence is specific, visually actionable, forward-looking, and concise on a 1–5 scale. Statistics are computed over 3,846 valid judge responses, with 95% confidence intervals for the mean scores.

Dimension	Mean	Standard Deviation	Median	95% CI
Specificity	3.27	1.10	3	[3.24, 3.31]
Actionability	3.67	1.27	4	[3.63, 3.71]
Forward-lookingness	3.68	1.32	4	[3.64, 3.72]
Conciseness	3.56	0.81	3	[3.53, 3.58]

B.2 Evidence-Retention Protocol and Statistical Analysis

The replay experiment described in Section 4.3 examines whether future-focus guidance improves the retention of visual evidence relevant to downstream questions. We replay one question-agnostic trajectory per question using its recorded future-focus sequence. Temporal bins that yield the correct answer when evaluated in isolation serve as proxy evidence. At each event, future-focus guidance and DTD-only selection retain equal numbers of tokens. After both masks are fixed, we introduce the downstream question in a retrospective forward pass and measure the fraction of reference question-to-visual attention mass covered by the selected tokens within the proxy-evidence bins.

The analysis includes 730 questions with proxy evidence after future focus becomes active, drawn from 382 StreamingBench source videos. Future-focus selection covers 19.86% of the question-attention mass while retaining 11.40% of the tokens in these bins. DTD-only covers 13.01% at 11.32% retention. Token counts are matched at each event across all DTD candidates, so the masks can retain different numbers of tokens within the proxy-evidence bins.

The absolute difference in attention mass, computed from unrounded values, is 6.86 percentage points. A percentile bootstrap clustered by source video with 50,000 samples gives a 95% confidence interval of $[6.48, 7.25]$ points. The interval excludes zero, supporting a statistically significant increase in retained question-attention mass.

B.3 Future Focus Audit

Table B.2 reports an LLM-as-a-Judge audit (Zheng et al., 2023) of generated future focus statements, using DeepSeek-V4-Flash (DeepSeek-AI, 2026). The judge receives the previous running summary and the focus statement, and scores the statement's specificity, visual actionability, forward-lookingness, and conciseness on a 1–5 scale. The audit tests suitability as a visual-selection query, not future-frame factual correctness, since future frames are unknown at generation time. The full judge prompt and scoring rubric are provided in Appendix E.2.

Across 3,846 valid judge responses, future focus scores highest on forward-lookingness (3.68) and visual actionability (3.67), both central to its role as a control signal. Conciseness averages 3.56, while specificity averages 3.27, reflecting the difficulty of generating question-agnostic guidance before knowing what the user will ask. These scores provide supporting evidence that the self-generated guidance is usable as a visual-selection query, rather than merely a free-form summary.

B.4 How Imperfect Focus Affects Visual Selection

DualStream uses self-generated focus statements to guide visual selection before future observations and user questions are available. The resulting guidance may emphasize aspects of the video that differ from those needed by a later question. Table B.3 illustrates this mismatch with a recorded card-game example from Video-MME (video `QdHBJ-NiaHM`, question `739-2`). The focus was generated at the 0-indexed anchor 6 and reused verbatim as the selection query at anchor 7.

The focus emphasizes strategic moves and changes in game state, whereas the later question concerns the character's class. Such guidance may give action-related tokens higher priority than details needed for class recognition, leaving some of those details outside the retained subset. However, the focus acts as a soft ranking signal over DTD candidates, while ρ determines the fraction retained.

Table B.3: Recorded future focus and later question in the card-game case. The focus text is reproduced verbatim from the case record.

Field	Recorded text
Future focus	*Watch for the gamer's next strategic move and the resulting changes in the game state.*
Later question	What is the occupation of the character the player controls?

Tokens are selected by relative rank, so details not explicitly mentioned in the focus can still be retained. DualStream appends the selected embeddings to its existing visual memory, preserving earlier evidence for subsequent observation updates and question answering.

Later observation updates provide an opportunity to revise these selection priorities. At each anchor, the backbone reads the accumulated visual memory, including the current anchor's new retained visual inputs, together with the previous running summary. This context informs an updated summary and a newly generated future focus. Once the update is complete, the new focus replaces the previous selection query. New observations can therefore redirect subsequent retention as the model's understanding of the episode evolves.

C Robustness and Efficiency

C.1 Effect of the Observation Schedule

We replace the adaptive observation schedule described in Appendix A.1 with updates every 180 seconds on the same 300 Video-MME long videos and 900 questions. This interval is close to the adaptive schedule's mean anchor spacing of 178.5 seconds. Both configurations use Full DualStream with the same backbone, compression parameters, and decoding setup.

Table C.1: Observation schedules for Full DualStream on the same 300 Video-MME long videos. Update counts are summed across all videos. Token drop is the weighted fraction of encoded visual tokens discarded.

Observation schedule	Observation updates	Accuracy (%)	Token drop (%)
Adaptive	4,146	60.00	86.06
Fixed, 180 s	4,243	58.33	85.92

Fixed-interval updates achieve 58.33% accuracy, compared with 60.00% for the adaptive schedule (Table C.1). The difference is -1.67 percentage points, with a 95% confidence interval of $[-3.56, 0.11]$ from 50,000 bootstrap samples clustered by video. The configurations have similar aggregate visual-token budgets. Cumulative encoded token counts differ by 0.52%, and token drop rates are 85.92% and 86.06%, respectively. These results suggest that DualStream remains effective with a simple periodic update schedule.

The fixed schedule performs slightly more observation updates overall. Update timing offers a plausible explanation for the adaptive schedule's numerical advantage. The first episodic update occurs at 41.7 seconds on average under the adaptive schedule, compared with 180 seconds under the fixed schedule. Earlier updates allow episodic guidance to influence visual retention sooner. Content-based boundaries may also provide more coherent event context for refreshing the running summary and future focus. Because each focus guides subsequent token selection, differences in these updates can propagate into later visual memory. The comparison therefore evaluates the scheduling policy as a whole, including update frequency and interval boundaries.

C.2 Compression Settings and Token Drop Rates

DualStream's compression depends on both the DTD threshold τ_d and the attention keep fraction ρ. DTD removes tokens whose cosine similarity to the spatially aligned tokens in the preceding frame exceeds τ_d. Attention-based selection then retains the top ρ fraction of the remaining candidates. The two parameters therefore jointly control how many encoded visual tokens enter memory.

Table C.2 reports τ_d, ρ, and the measured visual-token drop rate for each configuration in Figure 3. Token drop is the weighted fraction of encoded visual tokens discarded by the two filtering stages. As described in Appendix A.3, the default $\tau_d = 0.25$ follows TimeChat-Online (Yao et al., 2025), and $\rho = 0.55$ targets an overall token drop rate of approximately 85% to match the compression level of that baseline.

Table C.2: Compression settings and measured visual-token drop rates on Video-MME long for the 12 configurations in Figure 3.

τ_d	ρ	**Token drop (%)**
0.50	0.75	49.76
0.50	0.60	59.81
0.50	0.45	69.86
0.30	0.70	75.17
0.25	0.75	80.99
0.27	0.60	82.47
0.25	0.60	84.79
0.25	0.55	86.06
0.25	0.44	88.85
0.20	0.50	91.92
0.25	0.25	93.66
0.25	0.15	96.20

C.3 Runtime Profiling Details

Table C.3: Detailed runtime profile on Video-MME long, with visual updates and CPU scene detection measured at 1 fps. Percentiles are computed separately for each row. Episodic cadence is the mean interval between observation anchors.

Component	**Cadence**	**Mean (ms)**	**p95 (ms)**
Scene detection and decoding (CPU)	1 fps	**50.14**	**69.55**
Decoding and frame extraction		40.85	59.57
Scene detection		9.29	12.02
Complete visual-state update	1 fps	**625.95**	**885.45**
Input preprocessing		347.45	508.76
Visual encoding and selection		274.41	438.69
Vision encoding		96.73	107.10
DTD		0.46	0.55
Attention and selection		177.22	335.92
Position indexing		1.11	1.30
Scoring input construction		1.51	2.10
Memory append and other overhead		1.47	1.82
Local memory maintenance	On replacement	29.15	55.23
Episodic memory generation	178.5 s (mean)	4503.31	6621.70
Query-to-answer completion	On demand	1544.44	1812.36

Table C.3 reports the full runtime breakdown. The visual-state updates, episodic generation, and question answering are profiled on eight Video-MME long videos. A complete visual update takes 0.626 s on average and 0.885 s at p95. Episodic memory generation takes 4.503 s on average and 6.622 s at p95, accounting for 2.46% of the covered stream time.

The complete visual-state update is timed from event preparation through the addition of retained tokens to visual memory. Input preprocessing includes frame loading and sampling, processor operations, and transfer to the GPU. The visual encoding and selection subtotal includes the vision encoder, DTD, and attention scoring followed by selection of the top-ranked DTD candidates. Position indexing computes the temporal and spatial indices for visual tokens. Scoring input construction covers tokenizing the selection query and assembling the text and visual inputs for attention scoring.

The remaining overhead is computed by subtracting the listed components from each event's total latency. It includes prompt construction, appending retained tokens to memory, and bookkeeping. We compute each subtotal before aggregating across events, so the mean latencies add up, whereas the p95 values need not.

We separately profile CPU scene detection at 1 fps on three Video-MME long videos. We time 300 updates per video and repeat the measurement twice, yielding 1,800 timed updates. At each sampled timestamp, the decoder advances through intervening frames and extracts the latest frame available at that time. PySceneDetect's ContentDetector then processes this frame at its original 1280×720 resolution, with a threshold of 27 and a minimum scene length of one sampled frame. The total interval from decoding through scene detection takes 50.14 ms on average, with a p95 latency of 69.55 ms. Video opening is excluded from this interval.

D Backbone Adaptation and Cross-Backbone Evaluation

D.1 Training-Free Inference and Model-Specific Calibration

DualStream keeps the weights of the vision encoder, projector, and language model frozen, and the episodic feedback mechanism can be applied across VLM backbones. However, VLM backbones differ in visual embedding construction, model architecture, and sensitivity to hyperparameters. Adapting DualStream to a new backbone therefore requires both implementation changes and hyperparameter calibration, rather than simply replacing the checkpoint. The implementation must accommodate the model's visual-token layout, prompt boundaries, position encoding, and attention interfaces. Calibration selects a decoder layer and sets the redundancy threshold and attention keep fraction. These choices affect which evidence is retained without introducing learned parameters.

D.2 Transferring from Qwen2.5-VL to InternVL3-8B

We use `OpenGVLab/InternVL3-8B` to illustrate how to transfer DualStream to a new VLM backbone.

Visual encoding and token selection. With InternVL, we process each sampled frame as a single image tile, producing a spatial grid of projected visual tokens. We store each token with its frame index and spatial coordinates. DTD filters temporal redundancy by comparing tokens at corresponding locations in adjacent frames. Tokens without an aligned predecessor enter the candidate set. The attention stage ranks the candidates using the latest future focus and retains the specified fraction. We append the selected embeddings to memory in their original order, preserving frame membership and spatial correspondence.

Visual boundaries and position encoding. InternVL consumes the retained embeddings through its native conversation template. For each retained frame group, we construct a `Frame`i label followed by `<img>`, one `<IMG_CONTEXT>` placeholder per retained token, and `</img>`. Each placeholder embedding is replaced by its corresponding projected visual feature. The number of placeholders must equal the number of retained features, and their order must agree.

The memory coordinates and language-model position IDs serve different purposes. Frame and spatial coordinates support DTD, token ordering, and frame grouping. InternVL's language tower receives consecutive one-dimensional position IDs for the assembled text-and-visual sequence. Qwen2.5-VL instead uses its native multimodal 3-D position encoding. Keeping memory metadata separate from input position construction lets the same selection mechanism operate with both layouts.

Attention extraction and layer selection. We extract attention weights from the focus text to visual tokens at the selected decoder layer of InternVL. Averaging these weights over focus tokens and attention heads gives one score per visual token. Only the DTD candidates participate in the subsequent top-ρ selection.

We repeat the text-sensitivity probe in Appendix A.3 on the adapted model. Four focus prompts emphasize faces and expressions, visible text, vehicle motion, and background scenery on the same visual input. For each of the 28 decoder layers, we compute the mean pairwise correlation between the resulting visual-score vectors and select the layer with the lowest correlation. This procedure

selects the 0-indexed L14 for InternVL3-8B. The probe uses visual inputs and generic focus prompts without benchmark questions or answer labels.

Compression calibration. Different visual encoders induce different adjacent-frame similarity distributions, so a numerical DTD threshold can produce different retention rates across backbones. We calibrate the threshold using unlabeled visual inputs, accounting for the subsequent attention keep fraction. The InternVL DualStream configuration uses $\tau_d = 0.25$ and $\rho = 0.50$, targeting a compression level similar to that of the Qwen2.5-VL experiment. We measure the realized drop rate over the evaluation inputs, including tokens retained at boundaries where temporal comparison is unavailable.

D.3 Cross-Backbone Results

Table D.1 compares Visual-only and DualStream on the same 900 Video-MME long questions. Visual-only builds compact visual memory using visual redundancy scores. DualStream adds future focus to guide later visual retention. Both methods use frozen backbone weights.

Table D.1: Cross-backbone evaluation on Video-MME long. Token drop is the percentage of visual tokens removed. Bold marks the higher accuracy within each backbone. Each row reports its realized compression rate.

Backbone	Method	Correct / Total	Accuracy (%)	Token Drop (%)
Qwen2.5-VL-7B	Visual-only	520 / 900	57.78	86.06
Qwen2.5-VL-7B	DualStream	540 / 900	**60.00**	86.06
InternVL3-8B	Visual-only	518 / 900	57.56	86.47
InternVL3-8B	DualStream	536 / 900	**59.56**	87.90

DualStream improves accuracy by 2.22 percentage points on Qwen2.5-VL-7B and 2.00 points on InternVL3-8B. For InternVL, this corresponds to 18 additional correct answers. The results on both backbones show that episodic guidance can support visual retention across different VLM backbones with model-specific calibration.

E Prompts

We present the prompts used to update episodic memory and to audit the generated future focus statements. The observation prompt below is paired with retained embeddings from the compact visual memory. DualStream inserts these embeddings into the VLM input with the corresponding visual boundary tokens and backbone-specific position encoding, allowing the frozen backbone to read the retained pool as visual context. The Qwen2.5-VL implementation uses 3-D position IDs. The InternVL input layout is described in Appendix D. Throughout memory construction, the observation prompt remains question-agnostic.

E.1 Question-Free Observation Prompt

Table E.1 gives the prompt used at each observation anchor to update the running summary and generate the future focus for subsequent visual selection.

The `notes` field stores the running summary. It is replaced at each observation anchor with an updated cumulative account of the observed video. The `next_focus` field stores the future focus used as the text query for later attention-based visual selection.

E.2 Future Focus Audit Prompt

The prompt in Table E.2 is used to assess how well each generated future focus guides subsequent visual selection, using the previous running summary as context.

Table E.1: Question-free observation prompt.

Question-Free Observation Prompt

System message:
You are observing the same video over time. Earlier notes summarize what you have already seen when they are available. The current visual input reflects the video at the current point in time and may still include useful earlier context.
You do NOT yet know the final question. Track what matters so you can answer later.

Output a JSON object with exactly two keys:
1. "notes": Concise cumulative notes about everything seen so far. Cover people, objects, actions, scene changes, dialogue gist, and unusual details.
2. "next_focus": One short sentence about what to watch for next.

User message:
[Earlier notes]
{memory_notes}

Output JSON with keys: notes, next_focus.

Table E.2: Future focus audit prompt.

Future Focus Audit Prompt

System message:
You are auditing one sentence produced by a streaming video-memory system. The sentence is NEXT_FOCUS: it should guide which visual evidence to retain from future video frames. When PAST_NOTES are provided, use them only as the current episodic state. Score NEXT_FOCUS as a control signal. Do not judge unseen-video factual correctness. Return only JSON.

User message:
PAST_NOTES, the already-observed episodic state; do not score the notes: {notes_json}
NEXT_FOCUS: {focus_json}
Score NEXT_FOCUS on four dimensions from 1 to 5.

Important: NEXT_FOCUS was generated before future frames were seen. Do not require it to name exact future objects or events that are unknowable. Prospective target categories are valid when they can guide visual selection.

Use PAST_NOTES differently by dimension:
- specificity: score the visual target in NEXT_FOCUS; notes may resolve references. Do not require exact identity for future items.
- visual_actionability: score whether NEXT_FOCUS, with notes as context, can be turned into a visual token-selection query. Facial expressions, gestures, interactions, object use, scene changes, and text overlays are visual.
- forward_lookingness: compare NEXT_FOCUS against PAST_NOTES. Continuation, change, outcome, reaction, or the next instance of an observed activity is forward-looking; pure restatement is not.
- conciseness: score NEXT_FOCUS alone.

Rubric:
- specificity: 1=generic advice, 3=concrete target category, 5=clear observable people/objects/actions/scenes/text/state changes.
- visual_actionability: 1=not usable for visual retrieval, 3=partly usable, 5=directly usable as a visual token-selection query.
- forward_lookingness: 1=only recaps past content or answers a task, 3=broad future-oriented continuation/change, 5=clearly directs what to watch next.
- conciseness: 1=verbose or multiple tangled goals, 3=acceptable but wordy, 5=one compact high-density focus.

Return exactly this JSON object shape, with integer scores, reasons of at most 12 words, and one overall rationale of at most 16 words:
{"specificity": {"score": 1, "reason": "..."}, "visual_actionability": {"score": 1, "reason": "..."}, "forward_lookingness": {"score": 1, "reason": "..."}, "conciseness": {"score": 1, "reason": "..."}, "overall_rationale": "one short sentence"}.