WHEN TO RETRIEVE, WHEN TO STAY: UNCERTAINTY-AWARE TEMPORAL EVIDENCE ALLOCATION FOR STREAMING VIDEO-LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding requires Video Large Language Models (Video-LLMs) to reason over continuous visual streams under causal constraints. As the visual history grows, a bounded visual-processing budget requires evidence selection that balances temporal recency with query relevance. Recent-only selection excludes potentially relevant historical evidence, whereas Semantic-only retrieval can displace useful recent context when relevance scores are ambiguous. We introduce WRWS (When to Retrieve, When to Stay), a training-free framework for uncertainty-adaptive evidence allocation. A lightweight external vision-language encoder scores query relevance across the observed history, while an adaptive allocation module uses the normalized entropy of the similarity distribution as a proxy for retrieval uncertainty. WRWS favors semantic retrieval when relevance cues are reliable and strengthens the recency prior under uncertainty. Following a retrieve-first, encode-later pipeline, WRWS selects evidence before target-model visual encoding, such that only the selected observations are processed by the costly target Video-LLM. Experiments across four Video-LLM families and multiple model scales demonstrate competitive accuracy on StreamingBench and OVO-Bench. In our efficiency evaluation, WRWS reduces average vision-to-answer time to 47.93% of the state-of-the-art method. Code will be released.

1 INTRODUCTION

Multimodal large language models (MLLMs) have advanced video understanding (Li et al., 2024; Bai et al., 2025b), yet deploying them on continuously arriving video streams remains challenging. Unlike offline video understanding, streaming inference must answer queries using only observations available up to the query time, while potentially drawing on an increasingly long history (Lin et al., 2026; Niu et al., 2025). Processing this history with the target Video-LLM is expensive, even though only a subset of observations may ultimately be useful for a given query. This raises a central question: *under a bounded budget for expensive target-model visual processing, which causally available observations should receive target-model visual encoding?*

A natural training-free strategy is to retain the most recent observations (Shen et al., 2026). As illustrated in Fig. 1(a), such Recent-only selection provides a strong inductive bias for queries about the current scene, but earlier evidence becomes inaccessible once it falls outside the retained window. Expanding the window preserves more history, but also increases the visual-token load and the associated target-model processing cost. More fundamentally, a fixed Recent-only policy allocates the available budget solely according to temporal proximity, irrespective of which part of the observed history is relevant to the current query. Semantic retrieval offers a complementary strategy by selecting query-relevant observations from the causally available history (Di et al., 2025; Ning et al., 2025). However, as illustrated in Fig. 1(b), when query–observation similarities are ambiguous or insufficiently discriminative, semantic ranking provides a weak selection signal, and relying solely on similarity may displace otherwise useful recent context. The challenge is therefore not to choose recency or retrieval globally, but to determine *how strongly semantic relevance should override the recency prior for each query under a fixed evidence budget*.

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.

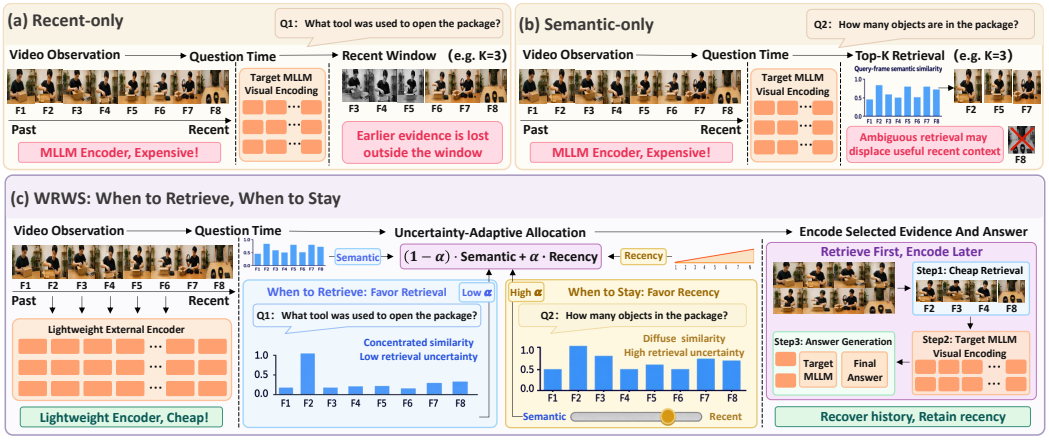


Figure 1: Motivation and key idea of WRWS. Under a bounded target-model visual-processing budget, **(a)** Recent-only selection may lose useful earlier evidence, while **(b)** Semantic-only retrieval may displace useful recent context when similarity is ambiguous. **(c)** WRWS adapts the semantic–recency trade-off to retrieval uncertainty and follows *retrieve first, encode later*, selecting evidence with lightweight representations before expensive target-model visual encoding.

These challenges are supported by two empirical observations in Sec. 2. First, target-model visual processing constitutes a major component of the measured vision-to-answer time (VTAT) runtime, motivating evidence selection before expensive target-model encoding. Second, Recent-only and Semantic-only exhibit substantial query-level complementarity under the same evidence budget, suggesting that no fixed policy is uniformly preferable across queries.

We therefore introduce **WRWS (When to Retrieve, When to Stay)**, a training-free framework for *adaptive temporal evidence allocation under a bounded target-model visual-processing budget*. As illustrated in Fig. 1(c), WRWS couples uncertainty-adaptive evidence allocation with a *retrieve-first, encode-later* pipeline. As observations arrive, a lightweight external vision encoder produces retrieval representations without invoking the target Video-LLM's expensive visual encoder. When a query arrives, WRWS estimates retrieval uncertainty from the query–observation similarity distribution and uses it to modulate the semantic–recency trade-off: concentrated similarity favors semantic relevance, whereas diffuse similarity strengthens the recency prior. The resulting policy selects a bounded set of observations before invoking the target Video-LLM only on the selected evidence. Because evidence selection is model-external, WRWS requires neither parameter updates nor access to target-model hidden states, attention statistics, or historical KV caches.

Evaluation spans four Video-LLM families and multiple model scales on streaming video understanding benchmarks (Lin et al., 2026; Niu et al., 2025), with additional experiments on offline long-video understanding. Across these settings, WRWS achieves competitive or improved online performance against strong training-free baselines. Compared with SimpleStream (Shen et al., 2026), WRWS reduces VTAT to approximately 48% while maintaining competitive downstream accuracy. Its model-external design is also compatible with vLLM (Kwon et al., 2023), demonstrating integration with an optimized multimodal serving backend.

Our contributions are threefold:

- We formulate streaming Video-LLM inference as temporal evidence allocation under a bounded target-model visual-processing budget. Query-level analysis reveals complementarity between Recent-only and Semantic-only selection. WRWS therefore uses retrieval uncertainty to adaptively balance semantic relevance and temporal recency for each query.
- WRWS selects Top-K evidence using lightweight external representations before invoking the target Video-LLM, thereby restricting expensive target-model visual processing to only the selected observations and keeping the per-query visual budget bounded.
- Across multiple Video-LLM families and model scales, WRWS achieves competitive or improved streaming performance while reducing VTAT to approximately 48% of SimpleStream under standard Transformers inference.

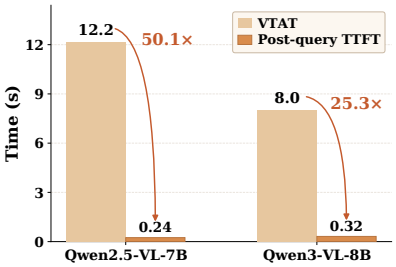


Figure 2: **Profiling target-VLM visual-processing cost.** VTAT is 50.1× and 25.3× inference TTFT for Qwen2.5-VL and Qwen3-VL.

Table 1: **Fixed evidence policies leave substantial query-level headroom.** Recent-only outperforms Semantic-only under the same evidence budget, while a per-query oracle consistently improves over both.

Backbone	Policy	Backward	Real-Time	Forward	Avg.
Qwen2.5-VL-7B	Semantic-only	47.12	59.08	39.17	48.46
	Recent-only	50.43	77.20	39.56	55.73
	Oracle	**59.21**	**81.77**	**45.63**	**62.20**
Qwen3-VL-8B	Semantic-only	48.46	62.87	43.95	51.76
	Recent-only	49.49	80.70	44.11	58.10
	Oracle	**60.56**	**84.45**	**52.25**	**65.75**

2 Empirical Observations

2.1 Do We Need to Process Every Incoming Observation with the Target VLM?

A straightforward streaming pipeline processes incoming observations with the target Video-LLM as they arrive, making their visual representations available when a query is issued. However, this commits expensive computation before knowing which observations will ultimately matter for a future query. We first quantify the cost of this encode-first strategy in practice.

Fig. 2 compares vision-to-answer time (VTAT) with inference time-to-first-token (TTFT). Both metrics end at the first generated answer token. VTAT is measured from the first visual observation, whereas inference TTFT starts when the target Video-LLM begins inference for answering. Under a fixed hardware and inference stack, VTAT therefore captures the cumulative runtime cost of processing the observed stream, while inference TTFT isolates the response cost after model inference. For Qwen2.5-VL-7B and Qwen3-VL-8B, VTAT is 12.18 s and 8.01 s, respectively, compared with only 0.243 s and 0.317 s for inference TTFT. Thus, VTAT is 50.1× and 25.3× the corresponding inference TTFT. Under our profiling protocol, this large gap shows that most of the measured vision-to-answer time is spent outside the target-model inference stage. A detailed VTAT breakdown in **Appendix** C further shows that target-VLM visual encoding is the largest component of this runtime. This profiling result motivates treating target-model visual processing as a limited inference-time resource rather than applying it indiscriminately to every incoming observation.

2.2 Is a Fixed Evidence Policy Sufficient?

Once expensive target-model encoding is deferred until query time, the next question is how to allocate this limited computation. A simple strategy is to prioritize the most recent observations, which provides a strong temporal prior for streaming queries. However, a fixed recency policy may overlook earlier observations that are more relevant to the current query.

Tab. 1 compares Recent-only with Semantic-only, which instead ranks observations by query-observation similarity under the same fixed evidence budget. Recent-only performs better on average for both Qwen2.5-VL-7B and Qwen3-VL-8B, reaching 55.73 and 58.10, versus 48.46 and 51.76 for Semantic-only, respectively. The advantage is especially pronounced on Real-Time tasks, where recent observations provide a particularly strong temporal prior.

The aggregate advantage of recency, however, does not mean that semantic retrieval is unnecessary. To examine their complementarity more directly, we construct a per-query oracle that selects the better result between Recent-only and Semantic-only for each query. The oracle reaches 62.20 with Qwen2.5-VL-7B and 65.75 with Qwen3-VL-8B, improving over Recent-only by 6.47 and 7.65 points, respectively. The gains are also consistent across all three task groups. The oracle gap reveals substantial query-level complementarity between the two fixed policies. This motivates WRWS to decide when to retrieve semantic evidence and when to retain recent context at query time.

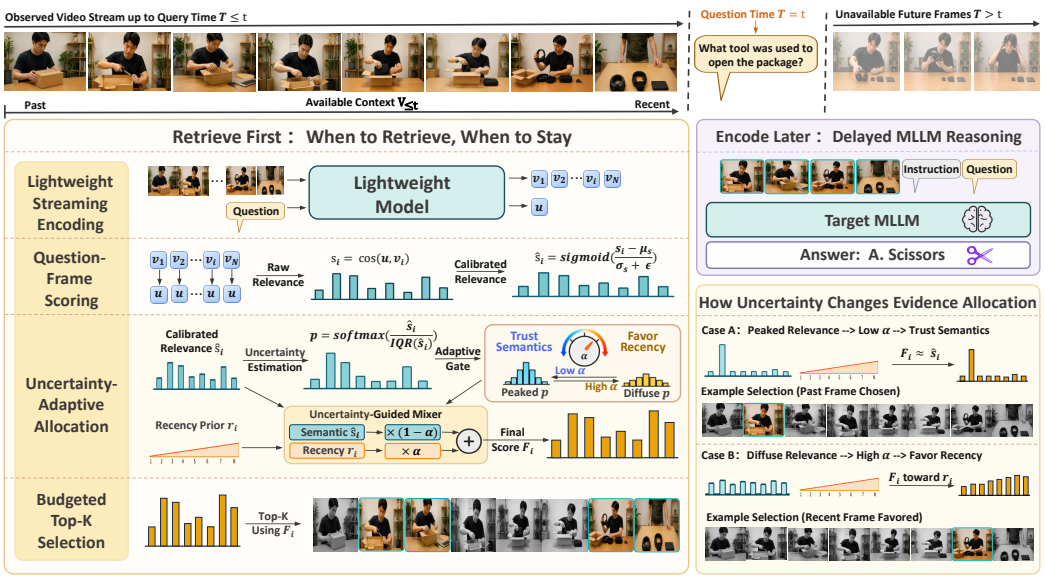


Figure 3: **Overview of WRWS.** WRWS uses lightweight representations to estimate query–observation relevance and retrieval uncertainty over the causally available history. The uncertainty adaptively balances semantic relevance and temporal recency to rank observations under a fixed Top-K budget. Only the selected observations are then encoded by the target Video-LLM, realizing our *retrieve-first, encode-later* strategy. The lower-right panel illustrates how peaked and diffuse relevance distributions shift evidence allocation toward semantics and recency, respectively.

3 Method

3.1 Problem Setup

Consider a video stream that produces a sequence of observations $\mathcal{H}_t = \{x_1, \ldots, x_t\}$ up to time t. When a query q_t arrives, a streaming Video-LLM must answer it using only the causally available history $\mathcal{H}_t$. Future observations $x_{t+1:}$ are unavailable.

Encoding every incoming observation with the target Video-LLM incurs expensive visual computation before the query's evidence needs are known. We therefore impose a bounded visual-processing budget at query time. Given the available history $\mathcal{H}_t$, the system selects a subset $\mathcal{S}_t \subseteq \mathcal{H}_t$ with $|\mathcal{S}_t| \le K$, where K denotes the maximum number of observations that may receive expensive target-VLM visual processing for each query. The selected observations in $\mathcal{S}_t$ are then encoded by the target Video-LLM and combined with q_t for answer generation, while the remaining observations bypass target-model visual encoding for that query.

The resulting problem is to allocate the target-model visual-processing budget across the causally available history at query time. Motivated by Sec. 2, the selection policy should account for both semantic relevance and temporal recency rather than committing to either fixed evidence policy. WRWS addresses this problem by using lightweight representations to retrieve candidate evidence and adapt the semantic–recency trade-off according to retrieval uncertainty. It then selects $\mathcal{S}_t$ and invokes the target Video-LLM only on these selected observations for answer generation.

3.2 Lightweight Query–Observation Retrieval

As illustrated in Fig. 3, WRWS first maintains lightweight representations of the observed stream, avoiding target Video-LLM invocation before evidence selection. As each observation x_i arrives, a lightweight vision-language encoder maps it to a visual representation $\mathbf{v}_i$, which is cached for subsequent retrieval. When query q_t arrives, the same model produces its text representation $\mathbf{u}$, and we compute the semantic relevance of each available observation by cosine similarity:

$$s_i = \cos(\mathbf{u}, \mathbf{v}_i), \qquad x_i \in \mathcal{H}_t. \tag{1}$$

Because similarity scores vary across queries, we calibrate them over the causally available history:

$$\hat{s}_i = \mathrm{sigmoid}\left(\frac{s_i - \mu_s}{\sigma_s + \epsilon}\right), \tag{2}$$

where μ_s and σ_s denote the mean and population standard deviation of $\{s_i\}_{x_i \in \mathcal{H}_t}$, and $\epsilon = 10^{-6}$ is a small constant for numerical stability. The calibrated score $\hat{s}_i$ provides a bounded semantic relevance signal for subsequent uncertainty estimation and adaptive semantic–recency allocation.

3.3 Uncertainty-Aware Temporal Evidence Allocation

Retrieval Uncertainty. Semantic retrieval is reliable when relevance is concentrated on a few observations, but becomes ambiguous when many observations receive similar scores. To quantify this ambiguity, we first convert the calibrated relevance scores into a probability distribution:

$$p_i = \frac{\exp\left((\hat{s}_i - m_t)/\tau_t\right)}{\sum_{j=1}^{N_t} \exp\left((\hat{s}_j - m_t)/\tau_t\right)}, \qquad m_t = \max_j \hat{s}_j, \tag{3}$$

where $\hat{\mathbf{s}} = \{\hat{s}_i\}_{i=1}^{N_t}$ and $N_t = |\mathcal{H}_t|$. For $N_t > 1$, the adaptive temperature is defined as

$$\tau_t = \begin{cases} \mathrm{IQR}(\hat{\mathbf{s}}), & \mathrm{IQR}(\hat{\mathbf{s}}) > \epsilon, \\ \mathrm{Std}(\hat{\mathbf{s}}), & \mathrm{IQR}(\hat{\mathbf{s}}) \le \epsilon \text{ and } \mathrm{Std}(\hat{\mathbf{s}}) > \epsilon. \end{cases} \tag{4}$$

If both dispersion measures are at most ϵ, the calibrated relevance scores are effectively indistinguishable and we use the uniform distribution $p_i = 1/N_t$. For $N_t = 1$, we set $p_1 = 1$. The adaptive temperature adjusts the softmax scale to the dispersion of calibrated relevance scores for each query. We then use normalized entropy as the retrieval uncertainty:

$$\alpha_t = \begin{cases} \dfrac{-\sum_{i=1}^{N_t} p_i \log p_i}{\log N_t}, & N_t > 1, \\ 0, & N_t = 1, \end{cases} \tag{5}$$

where $\alpha_t \in [0, 1]$. For a singleton history, no evidence-allocation decision is required, so we set $\alpha_t = 0$ by convention. As illustrated in the lower-right panel of Fig. 3, a peaked relevance distribution yields low α_t and places greater trust in semantic relevance, while a diffuse distribution yields high α_t and shifts the allocation toward recent observations.

Recency Prior. Semantic relevance alone does not capture the temporal prior of streaming queries, which often favors observations closer to the query time. We therefore assign each observation a recency score $r_i \in [0, 1]$ that increases monotonically with its temporal proximity to the query:

$$r_i = \begin{cases} \dfrac{i-1}{N_t - 1}, & N_t > 1, \\ 0, & N_t = 1, \end{cases} \qquad i = 1, \ldots, N_t. \tag{6}$$

Thus, more recent observations receive larger temporal weights, while earlier observations remain available for selection when supported by sufficiently strong semantic relevance.

Adaptive Semantic–Recency Allocation. We use the retrieval uncertainty α_t to adaptively combine semantic relevance and temporal recency:

$$F_i = (1 - \alpha_t)\hat{s}_i + \alpha_t r_i. \tag{7}$$

When retrieval is confident, a small α_t makes F_i primarily follow semantic relevance. As retrieval becomes more ambiguous, α_t increases and shifts the ranking toward recent observations. Through α_t, the score continuously interpolates between semantic relevance and temporal recency for each query in response to the estimated retrieval uncertainty.

3.4 Budgeted Selection and Delayed Target-VLM Encoding

Given the evidence scores $\{F_i\}_{i=1}^{N_t}$, we set $K_t = \min(K, N_t)$ and select the Top-K_t observations from the causally available history for subsequent target-model processing:

$$\mathcal{I}_t = \mathrm{TopK}_{\mathrm{t}\, i \in \{1,\ldots,N_t\}}\, F_i, \qquad \mathcal{S}_t = \{x_i \mid i \in \mathcal{I}_t\}. \tag{8}$$

Table 2: Performance comparison (%) on StreamingBench and OVO-Bench. Methods prefixed with "+" are applied to the corresponding base model. † denotes results reproduced following the official benchmark protocol. Gray rows denote WRWS. Best results are shown in **bold**.

Model	#Frames	StreamingBench	OVO-Bench			
		Real-Time	Backward	Real-Time	Forward	Avg.
Human	–	91.46	92.33	93.20	92.90	92.81
Proprietary MLLMs						
Gemini 1.5 Pro (Team et al., 2024)	1 fps	75.69	62.54	69.32	57.15	63.00
GPT-4o (Hurst et al., 2024)	64	73.28	60.75	64.46	53.40	59.54
Claude 3.5 Sonnet (Anthropic, 2024)	20	72.44	–	–	–	–
Open-source Offline MLLMs						
LLaVA-Video-7B (Zhang et al., 2024)	64	–	40.40	63.52	54.82	52.91
Qwen2-VL-7B (Wang et al., 2024)	64	69.04	46.46	55.98	48.74	50.39
InternVL2-8B (Chen et al., 2024b)	16	63.72	43.44	60.39	46.60	50.15
LongVU-7B (Shen et al., 2025)	1 fps	–	35.01	57.61	47.50	46.71
Open-source Online / Streaming MLLMs						
VideoLLM-online-8B (Chen et al., 2024a)	2 fps	35.99	17.73	20.79	–	–
Flash-VStream-7B (Zhang et al., 2025)	1 fps	23.23	27.38	28.37	45.09	33.61
Dispider-7B (Qian et al., 2025)	1 fps	67.63	36.06	54.55	34.72	41.78
TimeChat-Online-7B (Yao et al., 2025)	1 fps	75.28	44.50	61.40	36.80	47.60
StreamForest-7B (Zeng et al., 2026)	1 fps	77.26	52.02	61.20	53.49	55.57
Streamo-7B (1 fps) (Xia et al., 2026)	1 fps	–	46.10	65.98	54.77	55.61
Training-free Offline-to-Online Methods						
LLaVA-OV-0.5B (Li et al., 2024)	64	59.64	34.59	49.70	41.08	41.79
+ ReKV (Di et al., 2025)	0.5 fps	57.39	33.06	43.77	–	–
+ HERMES (6K tokens) (Zhang et al., 2026)	0.5 fps	61.04	34.75	50.34	–	–
+ HERMES (4K tokens) (Zhang et al., 2026)	0.5 fps	**62.04**	34.80	50.72	–	–
+ WRWS (Ours)	0.5 fps	61.54	**38.57**	**53.12**	41.09	44.26
LLaVA-OV-7B (Li et al., 2024)	64	71.34	43.71	64.02	50.50	52.74
+ ReKV (Di et al., 2025)	0.5 fps	69.22	44.16	57.33	–	–
+ HERMES (6K tokens) (Zhang et al., 2026)	0.5 fps	72.63	48.80	65.07	–	–
+ HERMES (4K tokens) (Zhang et al., 2026)	0.5 fps	73.23	50.20	66.34	–	–
+ WRWS (Ours)	0.5 fps	**75.35**	**50.78**	**67.45**	**51.09**	**56.44**
Qwen2.5-VL-7B (Bai et al., 2025b)	1 fps	73.31	44.65	59.90	39.02	47.86
+ HERMES (6K tokens) (Zhang et al., 2026)	1 fps	78.72	48.10	68.42	–	–
+ HERMES (4K tokens) (Zhang et al., 2026)	1 fps	79.44	49.43	68.98	–	–
+ FluxMem (Xie et al., 2026)	1 fps	76.40	47.24	67.20	–	–
+ SAVEMem (Wu et al., 2026)	1 fps	76.00	50.44	74.93	–	–
+ SimpleStream† (Shen et al., 2026)	1 fps	78.47	50.43	**77.20**	39.56	55.73
+ WRWS (Ours)	1 fps	**79.47**	**50.59**	77.13	**40.23**	**55.98**
Qwen2.5-VL-32B (Bai et al., 2025b)	1 fps	74.27	50.33	64.40	–	–
+ HERMES (6K tokens) (Zhang et al., 2026)	1 fps	80.20	**57.71**	71.93	–	–
+ HERMES (4K tokens) (Zhang et al., 2026)	1 fps	80.08	55.42	72.37	–	–
+ SimpleStream† (Shen et al., 2026)	1 fps	**80.31**	49.03	**80.58**	42.05	**57.22**
+ WRWS (Ours)	1 fps	80.23	49.28	80.13	**42.25**	**57.22**
Qwen3-VL-8B (Bai et al., 2025a)	1 fps	80.51	44.00	65.00	46.30	51.77
+ SimpleStream† (Shen et al., 2026)	1 fps	80.59	48.84	**81.26**	42.55	57.55
+ WRWS (Ours)	1 fps	**81.11**	**50.00**	80.74	**45.05**	**58.59**
Qwen3-VL-30B-A3B (Bai et al., 2025a)	1 fps	80.75	63.24	75.43	59.15	65.94
+ SimpleStream† (Shen et al., 2026)	1 fps	83.51	61.45	**85.48**	44.12	63.68
+ WRWS (Ours)	1 fps	**83.71**	**62.35**	84.94	**45.77**	**64.35**
Qwen3.5-4B (Qwen Team, 2026)	1 fps	77.03	56.61	67.99	**52.49**	59.03
+ WRWS (Ours)	1 fps	**78.39**	**57.00**	**79.83**	45.24	**60.69**
Qwen3.5-35B-A3B (Qwen Team, 2026)	1 fps	**84.55**	62.56	72.89	**58.38**	64.61
+ WRWS (Ours)	1 fps	82.99	**65.07**	**82.79**	46.40	**64.75**

When $N_t \leq K$, all causally available observations are therefore selected. Only the selected observations in $\mathcal{S}_t$ are passed to the target Video-LLM with q_t for answering, while unselected observations bypass expensive target-model visual processing for the current query.

Evidence selection is therefore completed before query-specific target-VLM visual processing. The target model processes at most K observations per query, and no target-model states are required for retrieval. This model-external design keeps target-model inference unchanged and therefore **supports optimized serving backends such as vLLM.**

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate WRWS on StreamingBench (Lin et al., 2026) and OVO-Bench (Niu et al., 2025) for streaming video understanding, using all 12 OVO-Bench tasks across the Backward,

Table 3: Performance comparison (%) on offline benchmarks. Methods prefixed with "+" are applied to the corresponding base model. Gray rows denote WRWS.

Model	MLVU	EgoSchema	VideoMME Long	VideoMME Avg.
Proprietary MLLMs				
Gemini 1.5 Pro (Team et al., 2024)	–	69.32	62.54	66.41
GPT-4o (Hurst et al., 2024)	64.60	64.46	60.75	62.87
Open-source Offline MLLMs				
LLaVA-Video-7B (Zhang et al., 2024)	70.80	57.30	–	63.30
Qwen2-VL-7B (Wang et al., 2024)	–	66.70	–	63.30
Open-source Online MLLMs				
Dispider-7B (Qian et al., 2025)	61.70	55.60	–	57.20
TimeChat-Online-7B (Yao et al., 2025)	62.60	61.90	41.70	53.22
StreamForest-7B (Zeng et al., 2026)	70.00	–	–	61.40
Training-free Offline-to-Online Methods				
LLaVA-OV-7B (Li et al., 2024)	64.70	59.93	48.00	57.67
+ ReKV (Di et al., 2025)	68.50	**60.70**	46.89	57.74
+ HERMES (6K tokens) (Zhang et al., 2026)	–	60.23	**49.11**	58.44
+ WRWS (Ours)	**71.96**	60.48	47.90	**58.85**

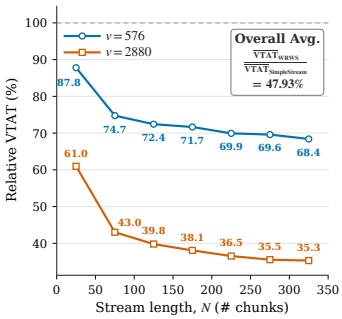


Figure 4: Relative VTAT across stream lengths for two common visual-token loads covering over 60% of samples. 47.93% is averaged over all samples.

Real-Time, and Forward groups. We additionally evaluate on EgoSchema (Mangalam et al., 2023), Video-MME (Fu et al., 2025), and MLVU (Zhou et al., 2025) for offline video understanding. We follow the official evaluation protocols unless otherwise specified.

Implementation. WRWS uses SigLIP (Zhai et al., 2023) for lightweight retrieval. Streaming inference is strictly causal and never accesses observations beyond the current streaming timestamp. Full implementation and evaluation details are provided in **Appendix** B.

4.2 Main Results

Online benchmarks. Tab. 2 compares WRWS with existing methods on StreamingBench and OVO-Bench. Across the four Qwen2.5/3 backbones, WRWS matches or improves the OVO-Bench average over SimpleStream, with gains of up to 1.04 points, while remaining comparable on StreamingBench. On LLaVA-OV-7B, WRWS achieves 75.35 on StreamingBench and 56.44 on OVO-Bench, outperforming the reported HERMES variants. WRWS further extends to the recent Qwen3.5 family, improving the OVO-Bench average over processing all sampled observations on both the 4B and 35B-A3B models. Results against HERMES under its 2-fps setting are provided in **Appendix** D.2.

Across backbones and scales. WRWS applies without additional training across LLaVA-OV, Qwen2.5-VL, Qwen3-VL, and Qwen3.5, spanning models from 0.5B to 35B-A3B. This suggests that its evidence-allocation mechanism is not tied to a specific Video-LLM family or model scale. Additional results across more model sizes and evidence budgets are provided in **Appendix** D.

Offline benchmarks. Tab. 3 further evaluates WRWS on offline video benchmarks. With LLaVA-OV-7B, WRWS achieves 71.96 on MLVU and a VideoMME average of 58.85, the highest among the compared training-free methods, while remaining competitive on EgoSchema and VideoMME.

4.3 Efficiency

Fig. 4 compares the VTAT of WRWS with SimpleStream on Qwen2.5-VL-7B as stream length increases. We visualize two common per-observation visual-token loads, which together account for over 60% of evaluation samples. Across all samples, WRWS requires only 47.93% of SimpleStream's VTAT on average. The relative VTAT further decreases with stream length for both loads, with larger savings under the higher visual-token load. This scaling behavior follows from avoiding expensive target-model visual processing over the full observed history as the stream grows. Additional TTFT and GPU-memory results are provided in **Appendix** F.

5 Understanding Adaptive Evidence Allocation

Building on the observed complementarity between fixed evidence policies (Sec. 2), we examine how their relative utility varies across tasks and how WRWS resolves the trade-off at the query level.

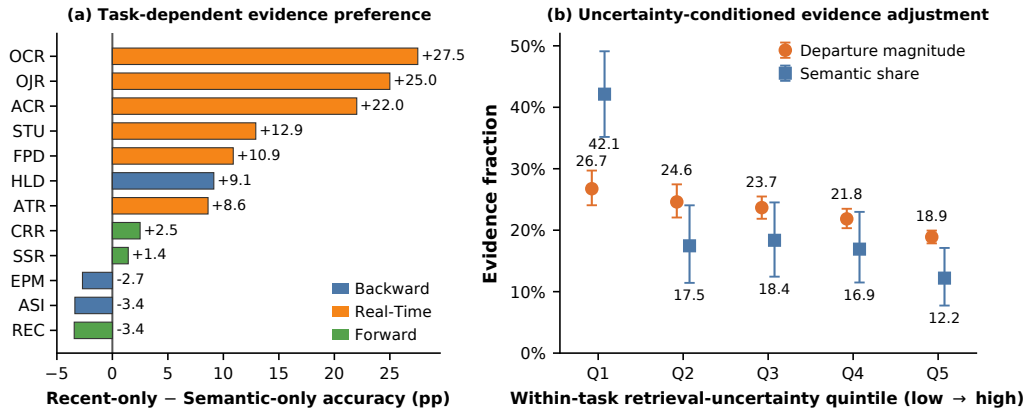


Figure 5: **Understanding adaptive evidence allocation.** **(a)** The relative utility of Recent-only and Semantic-only varies across temporal tasks. **(b)** When WRWS departs from Recent-only, lower retrieval uncertainty is associated with larger adjustments and a higher semantic share among the introduced evidence. Error bars denote 95% within-task stratified bootstrap confidence intervals.

5.1 Task-level Behavior

Fig. 5(a) compares Recent-only and Semantic-only across the 12 OVO-Bench tasks. Their relative utility varies across tasks. Recent-only performs better on all six Real-Time Visual Perception tasks, with gains of 8.6–27.5 percentage points, consistent with their reliance on current or nearby visual evidence. In contrast, Semantic-only performs better on EPM, ASI, and REC, which can require earlier or temporally distributed evidence. The preference is not determined solely by the benchmark-level task group: HLD favors Recent-only, while CRR and SSR show only modest differences. This task-level variation supports adaptive semantic-recency allocation at query time.

5.2 Sample-level Behavior

We next examine how retrieval uncertainty affects WRWS's final evidence selection. Let $\mathcal{A}_q$, $\mathcal{R}_q$, and $\mathcal{M}_q$ denote the evidence selected by WRWS, Recent-only, and Semantic-only for query q. WRWS retains a strong recency prior, matching Recent-only exactly on 79.9% of queries. For the remaining queries, we measure the departure magnitude

$$D_q = \frac{|\mathcal{A}_q \setminus \mathcal{R}_q|}{|\mathcal{A}_q|}, \tag{9}$$

which measures the extent of deviation from Recent-only, and the semantic replacement share

$$G_q = \frac{|(\mathcal{A}_q \setminus \mathcal{R}_q) \cap (\mathcal{M}_q \setminus \mathcal{R}_q)|}{|\mathcal{A}_q \setminus \mathcal{R}_q|}, \tag{10}$$

which measures the semantic alignment of the introduced evidence. Queries are grouped into retrieval-uncertainty quintiles independently within each task.

As shown in Fig. 5(b), the average departure magnitude decreases from 26.7% to 18.9% from the lowest- to highest-uncertainty quintile, while the semantic replacement share decreases overall from 42.1% to 12.2%. Thus, WRWS maintains a strong recency prior while making larger and more semantically aligned adjustments when retrieval is more confident.

6 Ablation Studies

Adaptive allocation. To assess the benefit of adaptive allocation, we compare it with representative fixed α values on Qwen2.5-VL and Qwen3-VL. As shown in Tab. 4(a), adaptive allocation outperforms the fixed settings reported in the main table on both backbones. The detailed α sweep in **Appendix** E further shows that the best fixed setting is backbone-dependent: $\alpha = 1.0$ for Qwen2.5-VL and $\alpha = 0.75$ for Qwen3-VL. Adaptive allocation removes the need to select such a backbone-specific trade-off and outperforms the best fixed setting by 0.25 and 0.34 points, respectively.

Table 4: **Ablation studies of WRWS.** † denotes the default setting. Best results are shown in **bold**.

(a) Adaptive evidence allocation

Allocation	Qwen2.5	Qwen3
$\alpha = 0.00$ (Semantic-only)	48.46	51.76
$\alpha = 0.50$	52.84	56.15
$\alpha = 1.00$ (Recent-only)	55.73	58.10
Adaptive†	**55.98**	**58.59**

(b) Recency prior

	Linear†	Plain	Exponential
Back.	**50.00**	48.46	49.71
RT	**80.74**	62.87	80.73
Fwd.	**45.05**	43.95	44.48
All	**58.59**	51.76	58.31

(c) Retrieval uncertainty

	Entropy†	Margin	Max	Std
Back.	**50.00**	49.49	49.49	49.94
RT	**80.74**	80.70	80.59	80.36
Fwd.	45.05	44.11	44.11	**45.39**
All	**58.59**	58.10	58.06	58.56

(d) Retrieval encoder

	SigLIP†	CLIP-L	SigLIP2	Target-VLM
Back.	**50.00**	49.37	49.59	48.59
RT	80.74	**80.91**	80.71	80.01
Fwd.	**45.05**	45.04	44.90	44.08
All	**58.59**	58.44	58.40	57.56

Recency prior. Tab. 4(b) compares linear and exponential recency weighting with a variant that removes it. Removing the recency prior reduces the score from 58.59 to 51.76, whereas linear and exponential weighting differ by only 0.28 points, suggesting that the presence of a recency prior matters more than its exact functional form. We use the simpler linear weighting by default.

Retrieval uncertainty. For retrieval uncertainty, we consider entropy, maximum confidence, margin, and standard deviation. As shown in Tab. 4(c), entropy and standard deviation, which characterize the similarity distribution as a whole, outperform maximum confidence and margin, which depend only on the top-ranked scores. We use entropy by default because it directly measures distributional uncertainty and achieves the best overall result.

Lightweight retrieval encoding. Tab. 4(d) compares three lightweight retrieval encoders with representations from the target Video-LLM. The lightweight encoders perform similarly, with SigLIP achieving the highest score of 58.59. Using target-model representations lowers the score to 57.56 despite requiring the expensive visual encoder of the Video-LLM. These results further support the use of lightweight external representations for retrieval before target-model visual processing.

7 Related Work

Streaming Video Understanding with Video-LLMs. Existing approaches enable streaming video understanding through streaming-specific training and architectures (Zhang et al., 2025; Huang et al., 2025; Chen et al., 2024a; Qian et al., 2024), or through training-free management of historical representations and redundant computation (Xiong et al., 2025; Di et al., 2025; Zhang et al., 2026; Xiao et al., 2026; Wang et al., 2026; Chen et al., 2026). The latter organize or compress memory and KV states, filter observations, or reuse visual representations to control context growth and computation. These methods optimize how observations are represented, retained, or reused within the target-model processing pipeline. WRWS focuses on an earlier decision: which causally available observations should receive expensive target-model visual processing in the first place.

Training-Free Temporal Evidence Selection. SimpleStream (Shen et al., 2026) establishes recent-frame retention as a strong training-free baseline, but allocates evidence solely by temporal proximity. Other methods adapt access to historical evidence: OASIS (Liang et al., 2026) uses preliminary MLLM reasoning over recent context and event summaries to trigger retrieval, while ShallowStream (Hao et al., 2026) uses target-model logits, KV states, and attention for routing and evidence selection. In contrast, WRWS adapts the semantic–recency trade-off before target-model visual encoding using retrieval uncertainty estimated from lightweight external representations. Its selection policy is fully model-external, requiring neither preliminary MLLM reasoning nor access to target-model attention or historical KV states.

8 Conclusion

WRWS addresses a central question in streaming Video-LLM inference: when should the model retrieve semantically relevant history, and when should it stay with recent context? Our experiments show that this choice varies across queries. Using retrieval uncertainty to make the choice allows WRWS to select evidence before target-model visual encoding, reducing average VTAT to 47.93% of SimpleStream while maintaining comparable downstream performance. The same formulation applies across multiple Video-LLM families and model scales without additional training.

AI USE STATEMENT

We used generative AI tools to aid and polish the writing of this paper, including improving grammar, clarity, readability, academic phrasing, and the organization of author-provided text. All AI-assisted edits were reviewed by the authors. The authors take full responsibility for the final content of this work, including all text, claims, results, and conclusions.

REPRODUCIBILITY STATEMENT

WRWS is a training-free inference framework that requires no parameter updates or training data. The main paper describes the problem formulation, algorithm design, and evaluation protocol, while the appendix provides additional implementation details and experimental settings. We provide the configurations of evaluated Video-LLMs, datasets, and inference settings to facilitate reproduction.

REFERENCES

Anthropic. Claude 3.5 sonnet model card addendum, 2024. URL https://www-cdn.anthropic.com/fed9cc193a14b84131812372d8d5857f8f304c52/Model_Card_Claude_3_Addendum.pdf.

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

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. *arXiv preprint arXiv:2502.13923*, 2025b.

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

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

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67 (12):220101, 2024b.

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.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Jitai Hao, Ke Yang, Qiang Huang, and Jun Yu. Shallowstream: Index shallow then answer deep for streaming video understanding. *arXiv preprint arXiv:2609.02780*, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338. IEEE, 2025.

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. *arXiv preprint arXiv:2410.21276*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. Oasis: On-demand hierarchical event memory for streaming video reasoning. *arXiv preprint arXiv:2604.17052*, 2026.

Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151. IEEE, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Zhenyu Ning, Guangda Liu, Qihao Jin, Wenchao Ding, Minyi Guo, and Jieru Zhao. LiveVLM: Efficient online video understanding via streaming-oriented KV cache and retrieval. *arXiv preprint arXiv:2505.15269*, 2025.

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

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

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

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

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 54582–54599, 2025.

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

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. *arXiv preprint arXiv:2403.05530*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18523–18533, 2026.

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

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*, pp. 31219–31229, 2026.

Junbin Xiao, Jiajun Chen, Tianxiang Sun, Xun Yang, and Angela Yao. Mukv: Multi-grained kv cache compression for long streaming video question-answering. *arXiv preprint arXiv:2605.22269*, 2026.

Yiweng Xie, Bo He, Junke Wang, Xiangyu Zheng, Ziyi Ye, and Zuxuan Wu. Fluxmem: Adaptive hierarchical memory for streaming video understanding. *arXiv preprint arXiv:2603.02096*, 2026.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. In *International Conference on Learning Representations*, volume 2025, pp. 69332–69351, 2025.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10807–10816, 2025.

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

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

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

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

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

A Overview

The supplementary material is organized as follows:

- **Detailed Experimental Setup (Sec. B).** We provide details on the target Video-LLMs, frame-sampling rates, retrieval encoder, evidence budgets, inference configuration, hardware, and latency measurement protocol.
- **Additional Analysis of Empirical Observations (Sec. C).** We provide a detailed VTAT breakdown and per-task results for Semantic-only, Recent-only, and the per-query oracle.
- **Additional Online Evaluation (Sec. D).** We report per-task results, comparisons with HERMES under the 2-fps setting, performance under different visual-processing budgets, and results across additional model scales.
- **Additional Analysis of Adaptive Evidence Allocation (Sec. E).** We provide detailed comparisons of adaptive and fixed evidence allocation, analyze how the preferred semantic–recency trade-off varies across models and tasks, and examine how retrieval uncertainty relates to the relative utility of the two evidence policies.
- **Additional Efficiency Analysis (Sec. F).** We provide detailed comparisons of VTAT, inference TTFT, and GPU memory usage.
- **Qualitative Analysis and Failure Cases (Sec. G).** We present representative examples of semantic–recency allocation and analyze failures arising from evidence selection and downstream Video-LLM prediction.
- **Limitations and Future Work (Sec. H).** We discuss retrieval-cache growth with stream length and the dependence of adaptive allocation on uncertainty estimated from external retrieval representations.

B Detailed Experimental Setup

Target Video-LLMs and frame sampling. We evaluate WRWS with eight target Video-LLMs: LLaVA-OneVision-0.5B and 7B, Qwen2.5-VL-7B and 32B, Qwen3-VL-8B and 30B-A3B, and Qwen3.5-4B and 35B-A3B. For streaming evaluation, we sample video observations at 0.5 fps for LLaVA-OneVision and 1 fps for the Qwen2.5-VL, Qwen3-VL, and Qwen3.5 families. For each target model, we use the same sampling rate across StreamingBench and OVO-Bench.

Retrieval and evidence budget. We use `google/siglip-so400m-patch14-384` (Zhai et al., 2023) as the lightweight retrieval encoder. For streaming benchmarks, we set the evidence budget K to 16 for LLaVA-OneVision, 4 for Qwen2.5-VL, and 6 for Qwen3-VL and Qwen3.5. These family-specific budgets keep the maximum number of visual tokens processed by the target Video-LLM at approximately 3K. We use the linear recency prior for all streaming experiments. The adaptive semantic–recency weighting introduces no additional tuned hyperparameters.

Offline evaluation. We evaluate EgoSchema, Video-MME, and MLVU with LLaVA-OneVision using the same sequential processing pipeline as in the streaming setting. We sample video observations at 0.5 fps and set $K = 32$, corresponding to approximately 6K target-model visual tokens and matching the 6K budget used by HERMES (Zhang et al., 2026). Since these benchmarks do not provide the same streaming-time recency semantics as online evaluation, we disable the recency prior by assigning an identical constant temporal weight to all candidate observations, making the temporal term effectively uniform. All other WRWS components remain unchanged.

Inference configuration. Our implementation builds on the SimpleStream codebase (Shen et al., 2026). We retain its target-model preprocessing and generation protocol, using the native processor of each Video-LLM and greedy decoding with at most 256 newly generated tokens. For streaming benchmarks, inference is strictly causal: at any streaming timestamp, WRWS only accesses observations that have arrived by that timestamp. All accuracy and efficiency results are obtained using standard Hugging Face Transformers inference.

Table 5: **Breakdown of VTAT for Qwen2.5-VL-7B under the encode-first baseline.** Target-VLM visual encoding accounts for 62.36% of the total VTAT. Other pipeline overhead is the residual after subtracting measured target-VLM visual encoding and inference TTFT from VTAT, which includes processing not captured by those two timers.

Metric	Target-VLM visual encoding	Other pipeline overhead	Inference TTFT	Total VTAT
Time (s)	**7.596**	4.342	0.243	12.181
Fraction (%)	**62.36**	35.65	1.99	100.00

Table 6: **Per-task breakdown of fixed evidence policies and the per-query oracle on OVO-Bench.** Semantic-only and Recent-only use the same fixed evidence budget. Oracle selects the better result between the two policies for each query. Gray rows denote the Oracle.

Model	Policy	Backward			Real-Time						Forward		
		EPM	ASI	HLD	OCR	ACR	ATR	STU	FPD	OJR	REC	SSR	CRR
Qwen2.5-VL-7B	Semantic-only	48.82	60.81	31.72	62.42	47.71	72.41	52.25	65.35	54.35	20.63	55.64	41.25
	Recent-only	52.19	58.78	40.32	91.95	67.89	81.90	67.42	75.25	78.80	18.48	56.44	43.75
	Oracle	**63.97**	**66.89**	**46.77**	**93.96**	**72.48**	**87.93**	**71.35**	**81.19**	**83.70**	**24.79**	**65.02**	**47.08**
Qwen3-VL-8B	Semantic-only	54.88	58.78	31.72	67.11	58.72	73.28	55.06	64.36	58.70	26.22	63.12	42.50
	Recent-only	52.19	55.41	40.86	94.63	80.73	81.90	67.98	75.25	83.70	22.78	64.55	45.00
	Oracle	**67.34**	**67.57**	**46.77**	**95.97**	**84.40**	**87.93**	**70.78**	**81.19**	**86.41**	**30.66**	**75.67**	**50.42**

Hardware and latency measurement. Experiments are conducted on two NVIDIA A100 80GB GPUs, while latency is measured on NVIDIA 4090 GPUs. We report two latency measures. Vision-to-answer time (VTAT) measures the elapsed time from the start of processing the first video observation to the generation of the first answer token, whereas inference time-to-first-token (TTFT) measures the elapsed time from the start of target Video-LLM inference for answering to the generation of the first answer token. We report the average VTAT and TTFT over all samples in OVO-Bench.

vLLM deployment. We additionally implement WRWS with vLLM to verify its compatibility with an optimized multimodal serving backend. The vLLM implementation is used only for the deployment study, with all other model configurations and WRWS settings kept identical to the standard setup in the main experiments. **All accuracy and efficiency results in the main experiments are obtained with the Transformers implementation.**

C Additional Analysis of Empirical Observations

We provide additional results for the two empirical observations in Sec. 2. We first decompose VTAT to examine where the latency of the encode-first pipeline is spent. We then report the full 12-task breakdown for Semantic-only, Recent-only, and the per-query Oracle.

C.1 VTAT Breakdown

Fig. 2 shows that Qwen2.5-VL-7B requires an average VTAT of 12.18 s under the encode-first pipeline, compared with only 0.243 s for inference TTFT. Tab. 5 further decomposes this latency.

Target-VLM visual encoding takes 7.596 s on average, accounting for 62.36% of the total VTAT. Inference TTFT accounts for only 0.243 s (1.99%). The remaining 4.342 s (35.65%) is the residual pipeline overhead after subtracting visual encoding and inference TTFT from VTAT. Target-VLM visual encoding is therefore the largest component of VTAT under our profiling setup. This result motivates identifying query evidence before invoking the target model's visual encoder, enabling visual processing to focus only on selected observations.

Table 7: **Per-task comparison (%) on OVO-Bench under online evaluation setting.** We report results across the Backward, Real-Time, and Forward task groups. HERMES results are taken from its original 6K-token setting. "–" indicates results not reported. Gray rows denote WRWS.

Model	Method	Backward			Real-Time						Forward		
		EPM	ASI	HLD	OCR	ACR	ATR	STU	FPD	OJR	REC	SSR	CRR
Qwen2.5-VL													
7B	SimpleStream	**52.19**	58.78	**40.32**	91.95	**67.89**	**81.90**	**67.42**	**75.25**	**78.80**	**18.48**	56.44	43.75
	WRWS	51.85	**60.14**	39.78	**92.62**	**67.89**	**81.90**	66.85	**75.25**	78.26	18.34	**56.92**	**45.42**
32B	SimpleStream	52.19	**54.05**	**40.86**	**93.96**	77.06	**86.21**	**71.35**	**72.28**	**82.61**	**19.91**	62.48	43.75
	WRWS	**54.55**	**54.05**	39.25	93.29	**77.98**	85.34	**71.35**	71.29	81.52	19.63	**62.96**	**44.17**
Qwen3-VL													
8B	SimpleStream	51.85	**55.41**	39.25	93.29	**83.49**	**83.62**	67.42	**78.22**	81.52	19.34	64.55	43.75
	WRWS	**54.55**	54.05	**41.40**	**94.63**	80.73	81.03	**68.54**	75.25	**84.24**	**22.78**	**64.86**	**47.50**
30B-A3B	SimpleStream	59.93	**64.19**	**60.22**	**95.97**	88.07	**87.93**	**70.22**	83.17	**87.50**	20.34	**69.95**	42.08
	WRWS	**62.63**	**64.19**	**60.22**	94.63	**88.99**	85.34	69.10	**85.15**	86.41	**22.78**	**69.95**	**44.58**
Qwen3.5													
4B	Base	**63.97**	**68.24**	37.63	80.54	62.39	73.28	52.81	74.26	64.67	**36.96**	60.10	**60.42**
	WRWS	58.59	60.81	**51.61**	**95.97**	**77.06**	**81.03**	**65.17**	**78.22**	**81.52**	20.63	**68.84**	46.25
35B-A3B	Base	**68.69**	**79.73**	39.25	84.56	71.56	76.72	58.99	79.21	66.30	**41.12**	72.34	**61.67**
	WRWS	64.98	71.62	**58.60**	**95.97**	**84.40**	**82.76**	**70.79**	**80.20**	**82.61**	20.92	**73.29**	45.00
LLaVA-OneVision													
7B	HERMES	**61.28**	58.78	26.34	72.48	**62.39**	69.83	47.75	**73.27**	64.67	–	–	–
	WRWS	59.93	**62.84**	**29.57**	**78.52**	59.63	**74.14**	**52.81**	**73.27**	**66.30**	27.36	67.57	58.33
0.5B	HERMES	**47.81**	**47.30**	9.14	57.05	49.54	**55.17**	32.58	**60.40**	47.28	–	–	–
	WRWS	47.47	44.59	**23.66**	**61.07**	**55.96**	**55.17**	**39.89**	54.46	**52.17**	15.04	62.80	45.42

C.2 Task-Level Analysis of Fixed Evidence Policies

Tab. 1 reports the aggregate comparison between Semantic-only, Recent-only, and the per-query Oracle. Tab. 6 gives the corresponding results for all 12 OVO-Bench tasks.

The relative performance of Semantic-only and Recent-only varies across tasks. For Qwen3-VL-8B, Recent-only performs better on all six Real-Time tasks, while Semantic-only performs better on EPM, ASI, and REC. Qwen2.5-VL-7B shows a similar pattern. Recent-only performs better on all six Real-Time tasks, whereas Semantic-only is stronger on ASI and REC.

The per-query Oracle improves over both fixed policies on every task for both backbones. Thus, the aggregate Oracle gains in Tab. 1 are not caused only by differences across task groups. Even within the same task, different queries can favor different evidence policies. This supports adapting the semantic-recency trade-off at the query level rather than fixing it globally or by task.

D Additional Online Evaluation

D.1 Detailed Results on the Main Online Benchmarks

Tab. 7 reports the per-task OVO-Bench results under the main online evaluation setting. For Qwen2.5-VL-7B, WRWS improves ASI, OCR, SSR, and CRR over SimpleStream, while giving similar results on several other tasks. At 32B, the improvements appear on EPM, ACR, SSR, and CRR, with comparable performance on ASI and STU. For Qwen3-VL-8B, WRWS performs better on EPM, HLD, OCR, STU, OJR, REC, SSR, and CRR. At 30B-A3B, it matches or outperforms SimpleStream on all three Backward tasks and improves ACR, FPD, REC, and CRR.

For Qwen3.5, the gains are concentrated on the Real-Time group. WRWS improves all six Real-Time tasks at both 4B and 35B-A3B. It also improves HLD and SSR at both scales, while several

Table 8: **Comparison with HERMES under the 2-fps online evaluation setting.** HERMES uses a 6K visual-token budget. WRWS selects $K = 6$ observations under its default Qwen3-VL streaming configuration. "B+R" denotes the average over the Backward and Real-Time groups. "–" indicates results not reported by HERMES. Gray rows denote WRWS.

Model	Method	Streaming	Backward	Real-Time	Forward	B+R	All
Qwen3-VL-8B	HERMES	**81.32**	46.78	73.21	–	60.00	–
	WRWS	80.39	**50.76**	**80.60**	44.17	**65.68**	58.51
Qwen3-VL-4B	HERMES	78.40	**54.00**	71.90	–	62.95	–
	WRWS	**80.07**	52.24	**78.64**	41.26	**65.44**	57.38

Table 9: **Per-task comparison with HERMES under the 2-fps online evaluation setting.** HERMES uses a 6K visual-token budget. "–" indicates results not reported. Gray rows denote WRWS.

Model	Method	Backward			Real-Time						Forward		
		EPM	ASI	HLD	OCR	ACR	ATR	STU	FPD	OJR	REC	SSR	CRR
Qwen3-VL-8B	HERMES	**54.88**	**70.95**	14.52	85.91	71.56	**81.03**	56.74	72.28	71.74	–	–	–
	WRWS	53.87	55.41	**43.01**	**94.63**	**83.49**	79.31	**71.35**	**73.27**	**81.52**	20.06	66.61	45.83
Qwen3-VL-4B	HERMES	54.55	**62.84**	44.62	81.88	73.39	**77.59**	58.43	73.27	66.85	–	–	–
	WRWS	**55.22**	54.73	**46.77**	**93.96**	**77.98**	76.72	**67.42**	**74.26**	**81.52**	17.19	65.34	41.25

other Backward and Forward tasks favor the base model. This task-level variation is consistent with the different temporal evidence requirements analyzed in Sec. 5.

For LLaVA-OneVision, we compare with the HERMES results reported under its 6K-token setting. Among the nine Backward and Real-Time tasks reported by HERMES, WRWS obtains higher accuracy on six tasks at 7B and five tasks at 0.5B.

D.2 Comparison with HERMES under the 2-fps Setting

We further compare WRWS with HERMES under the 2-fps online evaluation setting used in HERMES. As shown in Tab. 8, WRWS improves the Real-Time average from 73.21 to 80.60 on Qwen3-VL-8B and from 71.90 to 78.64 on Qwen3-VL-4B. The corresponding B+R averages increase from 60.00 to 65.68 and from 62.95 to 65.44. On StreamingBench, WRWS is slightly lower than HERMES at 8B (80.39 vs. 81.32) and higher at 4B (80.07 vs. 78.40).

Tab. 9 provides the per-task OVO-Bench results. On Qwen3-VL-8B, WRWS improves six of the nine Backward and Real-Time tasks, with particularly large gains on HLD, OCR, ACR, STU, and OJR. On Qwen3-VL-4B, WRWS improves seven of the nine reported tasks, including EPM, HLD, OCR, ACR, STU, FPD, and OJR. HERMES remains stronger on ASI and ATR at both model scales.

D.3 Effect of the Target-Model Visual-Processing Budget

We examine the effect of the target-model visual-processing budget by varying K from 2 to 16 on Qwen3-VL-30B-A3B. As shown in Tab. 10, increasing K from 2 to 4 improves the OVO-Bench average from 62.13 to 64.25. The overall score then changes only modestly as K increases from 4 to 8, reaching 64.35 at the default $K = 6$ and 64.77 at $K = 8$. With $K = 16$, the overall score further increases to 65.41. The gains mainly come from the Backward and Forward groups, whose averages increase to 64.53 and 50.41. In contrast, the Real-Time average decreases from 84.94 at $K = 6$ to 81.29 at $K = 16$. WRWS therefore remains effective across a range of visual-processing budgets, while the preferred budget varies with the temporal evidence requirements of different tasks.

The per-task results in Tab. 11 show the same trend. EPM, ASI, REC, and CRR generally benefit from a larger K, while several Real-Time tasks perform best with smaller or intermediate budgets. For example, ATR reaches its highest score at $K = 4$, while ACR and FPD peak at $K = 6$. Overall,

Table 10: **Performance under different target-model visual-processing budgets.** We vary the number of selected observations K for Qwen3-VL-30B-A3B. "B+R" denotes the average over the Backward and Real-Time groups. The gray row denotes the default setting ($K = 6$).

Model	Method	K	Streaming	Backward	Real-Time	Forward	B+R	All
30B-A3B	WRWS	2	78.83	60.46	83.03	42.91	71.74	62.13
	WRWS	4	83.51	62.17	**85.33**	45.26	**73.75**	64.25
	WRWS	6	83.71	62.35	84.94	45.77	73.59	64.35
	WRWS	8	83.67	62.35	84.62	47.33	73.49	64.77
	WRWS	16	**84.31**	**64.53**	81.29	**50.41**	72.91	**65.41**

Table 11: **Per-task performance under different target-model visual-processing budgets.** We vary the number of selected observations K for Qwen3-VL-30B-A3B. The gray row denotes the default setting ($K = 6$).

Model	Method	K	Backward			Real-Time						Forward		
			EPM	ASI	HLD	OCR	ACR	ATR	STU	FPD	OJR	REC	SSR	CRR
30B-A3B	WRWS	2	57.24	65.54	58.60	94.63	79.82	84.48	69.66	83.17	86.41	18.05	69.00	41.67
	WRWS	4	60.61	66.22	59.68	**95.97**	88.07	**87.07**	**70.22**	83.17	**87.50**	20.34	**70.43**	45.00
	WRWS	6	62.63	64.19	**60.22**	94.63	**88.99**	85.34	69.10	**85.15**	86.41	22.78	69.95	44.58
	WRWS	8	61.95	67.57	57.53	95.30	87.16	84.48	**70.22**	84.16	86.41	24.07	**70.43**	47.50
	WRWS	16	**66.33**	**70.27**	56.99	93.96	77.98	81.03	67.98	84.16	82.61	**27.36**	70.11	**53.75**

the results remain close for $K = 4$, 6, and 8, indicating that the default setting is not sensitive to a narrow choice of the visual-processing budget.

D.4 Results across Model Scales

We further compare WRWS with SimpleStream across additional model scales. For WRWS, we keep the evidence budget fixed within each model family, using $K = 4$ for Qwen2.5-VL and $K = 6$ for Qwen3-VL. **SimpleStream uses its best reported model-specific configuration.**

As shown in Tab. 12, WRWS remains close to SimpleStream across all evaluated scales. On Qwen2.5-VL, WRWS improves the overall OVO-Bench score at 3B, matches SimpleStream at 32B, and differs by only 0.07 points at 72B. The 72B comparison is particularly notable because WRWS uses $K = 4$, compared with $K = 16$ for SimpleStream. For Qwen3-VL, WRWS remains close to SimpleStream across the four evaluated scales, with differences ranging from -1.06 to $+0.67$ points. At 30B-A3B, WRWS exceeds SimpleStream by 0.67 points.

The per-task results in Tab. 13 further show that the differences between WRWS and SimpleStream vary across tasks and model sizes. WRWS improves several Forward tasks, while SimpleStream remains stronger on some Backward and Real-Time tasks. Overall, WRWS maintains comparable performance across model scales without model-specific adjustment of K within each model family.

E Additional Analysis of Adaptive Evidence Allocation

E.1 Adaptive vs. Fixed Evidence Allocation

We further compare adaptive allocation with fixed α values on Qwen2.5-VL-7B and Qwen3-VL-8B. As shown in Tab. 14, the best fixed α differs across backbones. Qwen2.5-VL-7B performs best with $\alpha = 1.0$, reaching 55.73, whereas Qwen3-VL-8B favors $\alpha = 0.75$, reaching 58.25. Adaptive allocation does not require selecting a model-specific α and achieves higher overall scores on both backbones, with 55.98 on Qwen2.5-VL-7B and 58.59 on Qwen3-VL-8B.

Tab. 15 provides the corresponding per-task results. The preferred fixed α varies across tasks on both backbones. For Qwen2.5-VL-7B, EPM and ASI favor $\alpha = 0.5$ and REC favors $\alpha = 0.25$,

Table 12: **Comparison with SimpleStream across model scales.** For each target model, we compare against the best reported SimpleStream configuration, which may use a model-specific evidence budget K. WRWS keeps $K = 4$ for all Qwen2.5-VL models and $K = 6$ for all Qwen3-VL models. "B+R" denotes the average over the Backward and Real-Time groups. Gray rows denote WRWS.

Model	Method	K	Streaming	Backward	Real-Time	Forward	B+R	All
Qwen2.5-VL								
3B	SimpleStream	4	77.83	**48.50**	**74.64**	43.19	**61.57**	55.44
	WRWS	4	**78.07**	47.27	74.56	**45.21**	60.91	**55.68**
32B	SimpleStream	4	**80.31**	49.03	**80.58**	42.05	**64.81**	**57.22**
	WRWS	4	80.23	**49.28**	80.13	**42.25**	64.65	57.22
72B	SimpleStream	16	**84.31**	**61.46**	79.52	**42.07**	70.49	**61.02**
	WRWS	4	82.35	60.89	**81.25**	40.72	**71.07**	60.95
Qwen3-VL								
2B	SimpleStream	4	76.55	**44.66**	**75.87**	43.77	**60.27**	**54.77**
	WRWS	6	**76.91**	44.60	74.51	**44.64**	59.55	54.58
4B	SimpleStream	16	**80.79**	**54.72**	77.20	**42.99**	**65.96**	**58.31**
	WRWS	6	80.51	50.81	**78.51**	42.44	64.66	57.25
32B	SimpleStream	8	**83.31**	64.88	83.05	**45.64**	73.96	**64.52**
	WRWS	6	83.03	**65.15**	**83.11**	44.27	**74.13**	64.18
30B-A3B	SimpleStream	4	83.51	61.45	**85.48**	44.12	73.46	63.68
	WRWS	6	**83.71**	**62.35**	84.94	**45.77**	**73.59**	**64.35**

Table 13: **Per-task comparison with SimpleStream across model scales.** SimpleStream uses its best reported model-specific evidence budget K, while WRWS keeps $K = 4$ for Qwen2.5-VL and $K = 6$ for Qwen3-VL. Gray rows denote WRWS.

Model	Method	K	Backward			Real-Time						Forward		
			EPM	ASI	HLD	OCR	ACR	ATR	STU	FPD	OJR	REC	SSR	CRR
Qwen2.5-VL														
3B	SimpleStream	4	**52.19**	**56.76**	**36.56**	88.59	73.39	**75.86**	**59.55**	**73.27**	**77.17**	17.62	68.20	43.75
	WRWS	4	51.85	56.08	33.87	**89.26**	**74.31**	**75.86**	**59.55**	72.28	76.09	**17.77**	**68.68**	**49.17**
32B	SimpleStream	4	52.19	**54.05**	**40.86**	**93.96**	77.06	**86.21**	**71.35**	**72.28**	**82.61**	**19.91**	62.48	43.75
	WRWS	4	**54.55**	**54.05**	39.25	93.29	**77.98**	85.34	**71.35**	71.29	81.52	19.63	**62.96**	**44.17**
72B	SimpleStream	16	**64.65**	**72.97**	46.77	93.96	70.64	82.76	67.42	**79.21**	**83.15**	**23.07**	55.64	**47.50**
	WRWS	4	59.60	68.24	**54.84**	**95.30**	**83.49**	**83.62**	**69.66**	72.28	**83.15**	18.05	**59.94**	44.17
Qwen3-VL														
2B	SimpleStream	4	51.85	**52.03**	**30.11**	92.62	**76.15**	**80.17**	**51.12**	75.25	**79.89**	18.62	71.86	40.83
	WRWS	6	**52.19**	**52.03**	29.57	**93.29**	70.64	77.59	**51.12**	**77.23**	77.17	**20.06**	**72.18**	**41.67**
4B	SimpleStream	16	**60.61**	**62.16**	41.40	92.62	75.23	**77.59**	62.92	**78.22**	76.63	**19.77**	62.96	**46.25**
	WRWS	6	55.22	54.73	**42.47**	**93.29**	**77.06**	75.00	**67.42**	76.24	**82.07**	18.34	**67.73**	41.25
32B	SimpleStream	8	55.56	**67.57**	**71.51**	**95.30**	**85.32**	**85.34**	72.47	**77.23**	82.61	**18.05**	**71.38**	**47.50**
	WRWS	6	**56.90**	**67.57**	70.97	**95.30**	84.40	**85.34**	**73.60**	75.25	**84.78**	15.33	71.22	46.25
30B-A3B	SimpleStream	4	59.93	**64.19**	**60.22**	**95.97**	88.07	**87.93**	**70.22**	83.17	**87.50**	20.34	**69.95**	42.08
	WRWS	6	**62.63**	**64.19**	**60.22**	94.63	**88.99**	85.34	69.10	**85.15**	86.41	**22.78**	**69.95**	**44.58**

whereas several Real-Time tasks prefer stronger recency weighting. Qwen3-VL-8B shows a similar trend, with smaller α values favored by EPM and ASI and stronger recency weighting by several Real-Time tasks. The task-dependent optima indicate that no single fixed semantic–recency trade-off works uniformly across tasks.

Table 14: **Comparison of adaptive and fixed evidence allocation on OVO-Bench.** Fixed α values interpolate between Semantic-only ($\alpha = 0$) and Recent-only ($\alpha = 1$). Adaptive denotes the query-adaptive allocation used by WRWS. "B+R" denotes the average over the Backward and Real-Time groups. Gray rows denote Adaptive.

Model	Allocation	Backward	Real-Time	Forward	B+R	All
Qwen2.5-VL-7B	$\alpha = 0.00$ (Semantic-only)	47.12	59.08	39.17	53.10	48.46
	$\alpha = 0.25$	47.97	64.46	**40.72**	56.21	51.05
	$\alpha = 0.50$	50.22	68.34	39.96	59.28	52.84
	$\alpha = 0.75$	49.97	73.95	38.58	61.96	54.16
	$\alpha = 1.00$ (Recent-only)	50.43	**77.20**	39.56	63.82	55.73
	Adaptive	**50.59**	77.13	40.23	**63.86**	**55.98**
Qwen3-VL-8B	$\alpha = 0.00$ (Semantic-only)	48.46	62.87	43.95	55.67	51.76
	$\alpha = 0.25$	48.80	69.72	45.90	59.26	54.81
	$\alpha = 0.50$	49.56	73.32	45.57	61.44	56.15
	$\alpha = 0.75$	49.90	78.39	**46.47**	64.14	58.25
	$\alpha = 1.00$ (Recent-only)	49.49	80.70	44.11	65.09	58.10
	Adaptive	**50.00**	**80.74**	45.05	**65.37**	**58.59**

Table 15: **Per-task comparison of adaptive and fixed evidence allocation on OVO-Bench.** Fixed α values range from Semantic-only ($\alpha = 0$) to Recent-only ($\alpha = 1$). Gray rows denote Adaptive. Best results for each backbone are shown in **bold**.

Model	Allocation	Backward			Real-Time						Forward		
		EPM	ASI	HLD	OCR	ACR	ATR	STU	FPD	OJR	REC	SSR	CRR
Qwen2.5-VL-7B	$\alpha = 0.00$	48.82	60.81	31.72	62.42	47.71	72.41	52.25	65.35	54.35	20.63	55.64	41.25
	$\alpha = 0.25$	52.19	59.46	32.26	74.50	49.54	74.14	56.74	69.31	62.50	**20.92**	57.07	44.17
	$\alpha = 0.50$	**54.21**	**64.19**	32.26	81.21	58.72	75.86	59.55	67.33	67.39	20.20	56.76	42.92
	$\alpha = 0.75$	52.53	60.81	36.56	89.93	66.06	80.17	64.04	72.28	71.20	18.19	**57.55**	40.00
	$\alpha = 1.00$	52.19	58.78	**40.32**	91.95	**67.89**	**81.90**	**67.42**	**75.25**	**78.80**	18.48	56.44	43.75
	Adaptive	51.85	60.14	39.78	**92.62**	**67.89**	**81.90**	66.85	**75.25**	78.26	18.34	56.92	**45.42**
Qwen3-VL-8B	$\alpha = 0.00$	54.88	58.78	31.72	67.11	58.72	73.28	55.06	64.36	58.70	**26.22**	63.12	42.50
	$\alpha = 0.25$	**55.89**	58.78	31.72	77.18	68.81	76.72	60.11	70.30	65.22	**26.22**	67.73	43.75
	$\alpha = 0.50$	55.22	**60.14**	33.33	83.89	74.31	77.59	65.17	68.32	70.65	24.93	**68.04**	43.75
	$\alpha = 0.75$	54.21	59.46	36.02	91.95	79.82	79.31	66.29	**75.25**	77.72	23.50	65.50	**50.42**
	$\alpha = 1.00$	52.19	55.41	40.86	**94.63**	**80.73**	**81.90**	67.98	**75.25**	83.70	22.78	64.55	45.00
	Adaptive	54.55	54.05	**41.40**	**94.63**	**80.73**	81.03	**68.54**	**75.25**	**84.24**	22.78	64.86	47.50

E.2 Uncertainty-Conditioned Utility of Semantic and Recent Evidence

As retrieval uncertainty increases, Recent-only exhibits a larger advantage over Semantic-only. In the lowest-uncertainty quintile, the two policies are nearly tied (62.09 vs. 62.91, $\Delta_{\mathrm{S-R}} = -0.82$ points). The gap widens to -9.59 and -7.82 points in the two highest-uncertainty quintiles. This trend suggests that retrieval uncertainty is associated with the relative downstream utility of semantic and recent evidence. Semantic retrieval remains more competitive under low uncertainty, while high uncertainty favors a stronger recency prior.

F Additional Efficiency Analysis

Tab. 17 provides a detailed efficiency comparison between WRWS and SimpleStream on OVO-Bench using Qwen2.5-VL-7B. Across all 3,035 samples, WRWS reduces the average VTAT from 12.18 s to 5.84 s, corresponding to 47.93% of the SimpleStream latency. The reduction is consistent across the three task groups, with VTAT ratios of 40.92%, 43.44%, and 57.74% for Backward, Real-Time, and Forward tasks, respectively. The largest relative reduction is observed on Backward tasks, while Forward tasks retain a larger fraction of the baseline latency.

Table 16: **Uncertainty-conditioned relative utility of semantic and recent evidence on OVO-Bench.** Queries are partitioned into five retrieval-uncertainty quintiles independently within each task and then micro-aggregated across tasks. $\Delta_{\mathrm{S-R}}$ denotes Semantic-only accuracy minus Recent-only accuracy in percentage points. Confidence intervals are obtained using 5,000 within-task stratified bootstrap repetitions. Bold indicates confidence intervals that exclude zero.

Uncertainty	# Queries	Semantic-only	Recent-only	$\Delta_{\mathrm{S-R}}$	95% CI
Q1 (lowest)	612	62.09	62.91	-0.82	$[-3.93, 2.45]$
Q2	609	52.71	55.83	-3.12	$[-6.57, 0.16]$
Q3	608	49.18	52.63	-3.45	$[-6.91, 0.00]$
Q4	605	43.14	52.73	$\mathbf{-9.59}$	$\mathbf{[-13.39, -5.79]}$
Q5 (highest)	601	41.60	49.42	$\mathbf{-7.82}$	$\mathbf{[-11.48, -4.16]}$

Table 17: **Detailed efficiency comparison with SimpleStream on OVO-Bench.** VTAT measures the elapsed time from the start of processing the first video observation to the generation of the first answer token, while inference TTFT measures the elapsed time from the start of target Video-LLM inference for answering to the generation of the first answer token. "Ratio" reports WRWS relative to SimpleStream, where a lower VTAT ratio indicates larger latency savings. GPU memory is measured over the full evaluation.

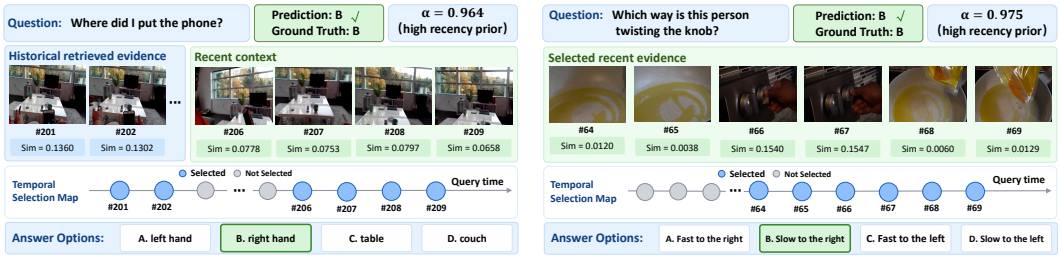
Group	# Samples	VTAT (s)			Inference TTFT (s)			GPU Memory (GB)	
		WRWS	SimpleStream	Ratio	WRWS	SimpleStream	Ratio	WRWS	SimpleStream
Overall	3035	**5.84**	12.18	**47.93%**	0.2431	0.2429	100.08%	20.76	**17.28**
Backward	631	**6.37**	15.56	**40.92%**	0.2788	0.2793	99.82%		
Real-Time	837	**7.17**	16.50	**43.44%**	0.2863	0.2866	99.91%		
Forward	1567	**4.92**	8.51	**57.74%**	0.2057	0.2050	100.35%		

Inference TTFT measures the elapsed time from the start of target Video-LLM inference for answering to the generation of the first answer token. The overall TTFT is 0.2431 s for WRWS and 0.2429 s for SimpleStream, with similarly small differences across the three task groups. After the query arrives, both methods perform the same target-model visual encoding and answer generation under the same evidence budget. Thus, their nearly identical TTFT is expected, and the remaining sub-millisecond differences are likely attributable to normal runtime measurement noise.

WRWS uses 20.76 GB of GPU memory, compared with 17.28 GB for SimpleStream. The additional memory mainly comes from the SigLIP retrieval encoder and the cached representations of historical observations. In our implementation, these representations are kept on GPU throughout evaluation, so the retrieval cache grows with the observed video history. The per-observation storage remains lightweight, as WRWS retains only a single global retrieval embedding for each observation, whereas target-model visual representations typically contain multiple visual tokens per observation. Furthermore, the retrieval cache can also be **offloaded to CPU memory**, as commonly done for historical representations in streaming systems, to further reduce its GPU footprint. We report the all-GPU setting because it avoids additional host–device transfers and offloading logic in the serving pipeline. Like SimpleStream, WRWS forwards only the selected K observations to the target Video-LLM for each query, regardless of the observed video length. Its target-model visual context therefore remains bounded by the evidence budget. When the retrieval cache is offloaded to CPU memory, WRWS also maintains a length-stable GPU-memory profile similar to SimpleStream, in contrast to methods whose GPU footprint grows with the accumulated visual history, such as TimeChat-Online (Yao et al., 2025). Under the all-GPU setting reported here, WRWS reduces average VTAT by 52.1% at the cost of an additional 3.48 GB of GPU memory.

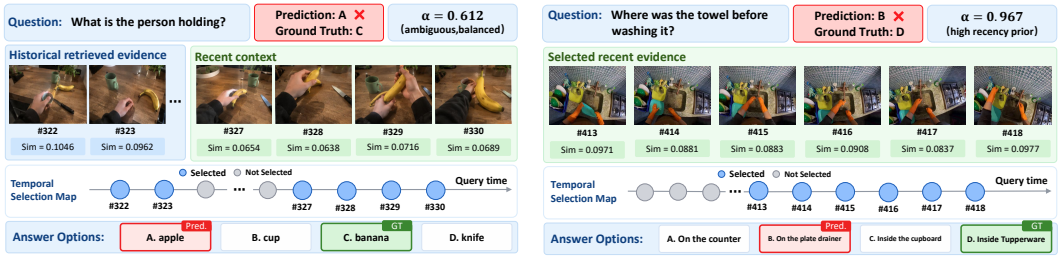
G Qualitative Analysis and Failure Cases

We further examine representative examples to understand how WRWS allocates evidence across time and where its remaining errors arise.



(a) **Retrieving history while keeping recent context.** (b) **Staying recent when local evidence is sufficient.**

Figure 6: **Qualitative examples of adaptive evidence allocation. (a)** When earlier observations are more relevant to the query, WRWS retrieves historical evidence while keeping recent context. **(b)** When the relevant action occurs near the query time, WRWS shifts the evidence budget toward recent observations. WRWS answers both queries correctly.



(a) **Failure despite relevant selected evidence.** (b) **Failure when earlier evidence is missed.**

Figure 7: **Qualitative failure cases. (a)** WRWS selects both earlier and recent observations in which the banana is visible, yet the target Video-LLM predicts "apple" instead of "banana". This suggests that evidence selection alone does not eliminate downstream visual recognition or reasoning errors. **(b)** When WRWS concentrates its budget on recent observations, the earlier evidence needed to recover the towel's previous location is not selected, leading to an incorrect answer.

Figs. 6a and 6b illustrate two successful allocation patterns. In Fig. 6a, several earlier observations receive stronger semantic similarity than the recent context and are retained together with recent frames. This allows WRWS to recover historical evidence without discarding local context near the query time. Fig. 6b shows the opposite regime: the evidence needed to answer the query is already contained in the recent observations, and WRWS keeps the allocation concentrated near the query time. Together, these examples illustrate the intended behavior of the adaptive semantic–recency trade-off rather than committing to a fixed temporal policy.

The failure cases reveal two different sources of error. In Fig. 7a, answer-relevant visual evidence is already present in the selected observations, yet the target Video-LLM still predicts an incorrect option. The failure therefore cannot be explained solely by evidence selection, and instead suggests a limitation in downstream visual recognition or reasoning. By contrast, Fig. 7b exposes a selection failure. The query requires information from an earlier stage of the video, while the selected set is dominated by recent observations around the current washing activity. The relevant historical evidence is consequently absent from the target model's input. These cases show that uncertainty-adaptive allocation improves temporal evidence selection but does not eliminate errors arising from either evidence allocation or the reasoning capability of the underlying Video-LLM.

H Limitations and Future Work

WRWS currently retains lightweight retrieval representations for the observed history, so the retrieval-side cache grows with stream length. While these representations are substantially more compact than target-model visual states and can be offloaded from GPU memory, combining WRWS with cache compression or eviction could further improve scalability to very long streams. In addition, the current allocation policy estimates retrieval uncertainty from the similarity distribution of an external encoder. Future work may explore improved uncertainty calibration and richer retrieval signals to further refine adaptive evidence allocation.