BEYOND RETRIEVAL: PROGRESSIVE LATENT MEMORY EVOLUTION FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding requires multimodal large language models to process continuous visual inputs and respond to user queries under strict causality and bounded memory. Existing approaches typically compress historical observations into an external memory bank and retrieve query-relevant evidence as additional visual context. Though effective, this store-and-retrieve paradigm keeps historical evidence as external visual context, preventing it from being internalized into a compact, evolving latent memory that can continuously guide streaming reasoning. To bridge this gap, we introduce **`LatentStream`**, a progressive latent working memory framework that shifts streaming memory from *store-and-retrieve* to *retrieve-and-internalize*. Specifically, **`LatentStream`** comprises three coordinated components. First, Query-Agnostic Hierarchical Streaming Memory organizes visual history into short-, mid-, and long-term levels through Jenks-guided adaptive consolidation. Once a query arrives, Hierarchical Latent Memory Evolution equips groups of latent memory tokens with progressively expanding memory receptive fields, enabling them to iteratively retrieve historical evidence from their corresponding scopes and internalize it into compact latent memory. Finally, Progressive Confidence-guided Latent Memory Optimization constructs a hierarchical progression reward from group-wise predictive entropy and jointly refines the latent memory tokens and retrieved evidence, encouraging increasingly confident streaming reasoning. Extensive experiments demonstrate that **`LatentStream`** achieves new state-of-the-art results on existing online and offline video benchmarks.

1 INTRODUCTION

Streaming video understanding requires models to continuously process incoming observations and respond to user questions that may be posed at any time [12, 55]. This capability is essential for a wide range of real-world applications, including live monitoring [15], autonomous driving [6, 4], smart glasses [62, 25], as well as embodied agents and robotic systems [44, 31].

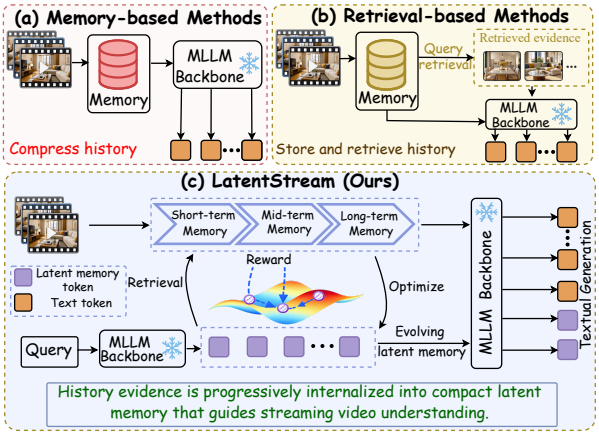


Figure 1: Motivation of our **`LatentStream`**: paradigm comparison of memory-based and retrieval-based methods with our retrieve-and-internalize method **`LatentStream`**.

To equip existing Video-LLMs [69, 3, 26, 46] for continuously arriving visual streams, recent efforts have explored streaming video understanding from multiple perspectives, ranging from dedicated model training [32, 18, 65] to streaming inference [56] and memory-based context management [54, 28, 50]. Among them, *training-free* approaches are more appealing as they directly adapt off-the-shelf Video-LLMs to streaming scenarios without additional parameter updates (Fig. 1a). Their key strategy is to manage the continuously accumulated visual context at inference time through frame selection [39], visual token pruning or merging [14, 60], KV-cache management [36, 9], and fixed-capacity or

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.

hierarchical memory [54]. More recent approaches [28, 50] further couple such memory management with query-aware retrieval, selectively recalling relevant evidence from the retained history once a query arrives (Fig. 1b). By filtering redundant observations, these methods effectively constrain the growth of historical context, reducing memory and computational overhead for video streams.

Despite these advances, existing methods typically treat streaming memory as an external bank of historical evidence, focusing primarily on what information to retain and what to retrieve once a query arrives [28, 54, 50]. Although query-aware memory retrieval allows the model to identify relevant visual evidence from history, the retrieved evidence is still exposed to the model as external visual context. Such a retrieval-centric paradigm essentially addresses which historical evidence should be accessed, while leaving unexplored how the accessed evidence can be internalized into a compact latent state that participates directly in subsequent video reasoning. Thus, query-agnostic streaming memory and query-conditioned reasoning remain loosely coupled: *there is no compact latent memory representation that can accumulate task-relevant historical evidence and continuously evolve alongside the video reasoning process in the model latent space*. This work posits that the model's latent space [30, 7, 47, 59] provides a natural substrate for bridging this gap.

Building on this view, we argue that streaming video reasoning calls for a **latent working memory**: rather than exposing retrieved visual evidence to the model as additional context, such a latent memory progressively internalizes task-relevant history into a fixed-length latent state that guides the streaming video reasoning process. Crucially, this latent state is not a one-shot compression of the retrieved history, but evolves with newly accessed evidence and, in turn, further guides the retrieval of relevant historical information. This creates an iterative interplay between external memory access and latent memory evolution, turning the conventional store-and-retrieve mechanism into a retrieve-and-internalize paradigm. This naturally raises our pivotal research question:



How can historical evidence in external streaming memory be adaptively internalized into a compact and evolving latent memory for streaming video reasoning?

To answer this question, we introduce `LatentStream`, a progressive latent working memory framework for streaming Video-LLMs (Fig. 1c). Instead of treating retrieved history as auxiliary context, `LatentStream` progressively internalizes task-relevant visual evidence into a compact, evolving latent memory that continuously guides streaming reasoning. At its core, `LatentStream` comprises three coordinated components. First, `LatentStream` constructs a ♣ **Query-agnostic Hierarchical Streaming Memory (HSM)** that organizes incoming observations into short-, mid-, and long-term memories under a fixed budget. A Jenks-guided adaptive consolidation strategy progressively reduces temporal and spatial redundancy while preserving informative evidence across extended video streams. Second, `LatentStream` introduces ♦ **Hierarchical Latent Memory Evolution (HME)** to internalize such hierarchical memory into Latent Memory Tokens (LMTs), which are divided into three groups with progressively expanding memory receptive fields. At each evolution iteration, the LMTs retrieve relevant evidence from their respective memory scopes and evolve jointly with the retrieved evidence, subsequently guiding the next round of historical access, forming iterative memory retrieval–update evolution. Third, to regulate this latent memory evolution, `LatentStream` develops ♥ **Progressive Confidence-guided Latent Memory Optimization (PMO)**, which constructs a hierarchical confidence progression reward from group-wise latent token predictive entropy that encourages increasingly confident reasoning as the accessible historical scope expands. Guided by this objective, the latent memory tokens progressively absorb task-relevant evidence through test-time optimization without modifying the Video-LLM parameters.

We extensively validate `LatentStream` on five online and offline video understanding benchmarks, and experimental results show that `LatentStream` achieves new state-of-the-art across diverse tasks (§4.2). Specifically, it achieves **64.2%** on OVO-Bench [34] and **76.9%** on StreamingBench [29] for streaming evaluation, while attaining **66.6%** on VideoMME [16], **74.0%** on MLVU [72], and **62.1%** on LongVideoBench [51] for offline long-video understanding. In §4.3, extensive ablation studies confirm the power of our idea and the efficacy of core model designs.

In summary, our main contributions are as follows: **First**, we introduce `LatentStream`, a **latent working memory framework** that advances conventional *store-and-retrieve* memory toward a *retrieve-and-internalize* paradigm, progressively internalizing task-relevant historical evidence into compact and evolving latent memory that can continuously guide streaming video reasoning. **Second**, we propose **Hierarchical Latent Memory Evolution**, which couples Jenks-guided hierarchical

streaming memory with three latent memory token groups of expanding memory receptive fields, enabling them to iteratively retrieve historical evidence from their corresponding scopes and internalize it into a compact, fixed-length latent memory. **Third**, we develop **Progressive Confidence-guided Latent Memory Optimization**, which adopts a hierarchical progression reward based on group-wise latent token predictive entropy and refines the latent memory token embeddings, encouraging increasingly confident streaming reasoning.

2 Related Work

Streaming Video Understanding. Unlike offline video understanding, streaming video understanding requires models to causally process continuously arriving observations and respond to user queries in real time. Existing studies fall into four groups. **(i)** *Proactive interaction* methods aim to determine what and when to respond through response prediction heads [1, 57], generative trigger tokens [53, 68], event-aware activation mechanisms [54, 19], or reinforcement learning [32]. **(ii)** *Streaming memory* methods [54, 50, 55] preserve useful historical information under bounded context and computation budgets, enabling models to access past observations during interaction. **(iii)** *Streaming thinking* methods [65, 32, 18, 45] further couple perception with reasoning, allowing intermediate reasoning states to evolve with incoming observations instead of postponing reasoning until a query arrives. **(iv)** In parallel, substantial efforts are devoted to *real-time inference*, reducing the cost of continuous video processing via selective model invocation [23, 13], visual token reduction [48, 52, 43], or KV-cache optimization [64, 24]. Together, these advances move video-language models toward causal, persistent, and real-time interaction in evolving visual environments.

Long-term Memory Management in Streaming Videos. A fundamental challenge in streaming video understanding is to preserve useful historical information from an unbounded visual stream under bounded memory and context budgets. Existing approaches fall into four categories. **(i)** *Hierarchical multi-level memory* [42, 55, 54] organizes historical observations across temporal scales, typically preserving detailed recent context while consolidating older observations into compact long-term representations. **(ii)** *Visual token compression and pruning* [60, 27] control memory growth by removing redundant visual tokens while retaining informative content. **(iii)** *KV-cache memory* [12, 33, 58, 64, 8] directly compresses, retrieves, or reuses cached internal states to bound GPU memory and avoid re-encoding historical observations. **(iv)** *Retrieval-augmented memory* [71, 28, 70] decouples long-term storage from the active reasoning context, maintaining historical visual features or compressed memories externally and retrieving query-relevant evidence on demand. However, existing streaming memory methods primarily consume retrieved history as auxiliary visual context or KV cache, rather than maintaining an explicit working state that evolves with accumulated evidence. In contrast, `LatentStream` internalizes retrieved evidence into compact latent memory tokens that iteratively evolve and guide subsequent memory access, allowing historical evidence to directly participate in streaming video reasoning.

3 Methodology

3.1 Framework Overview

We propose `LatentStream` (Fig. 2), a progressive latent working memory framework for streaming video understanding. First, **Query-agnostic Hierarchical Streaming Memory** (§3.2) continuously organizes incoming visual tokens into short-, mid-, and long-term memories via Jenks-Based Adaptive Memory Consolidation. Second, **Hierarchical Latent Memory Evolution** (§3.3) equips different groups of Latent Memory Tokens (LMTs) with progressively expanding memory receptive fields. Through selective latent-guided historical evidence retrieval and injection in the frozen Video-LLM, these LMTs progressively absorb critical visual evidence across temporal scales into a compact latent memory. Third, **Progressive Confidence-guided Latent Memory Optimization** (§3.4) estimates the prediction uncertainty associated with different LMT groups and constructs a hierarchical confidence reward, encouraging increasingly confident reasoning as the accessible historical scope expands. Guided by this objective, the additionally retrieved visual token embeddings and the LMT embeddings are jointly optimized at test time. After optimization, the retrieved visual tokens are removed from the generation context, and the final answer is generated from the bounded external memory, the query, and the optimized latent memory tokens.

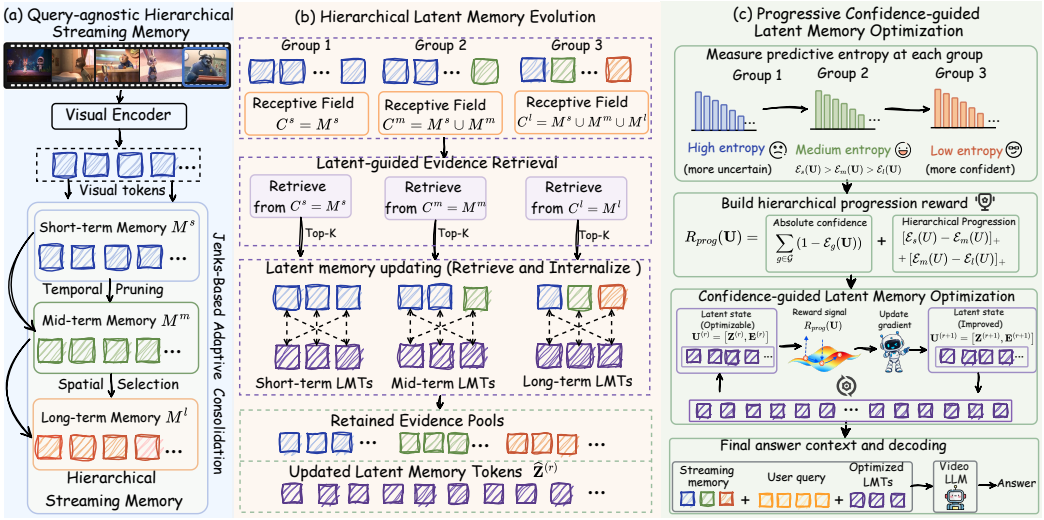


Figure 2: The overview of `LatentStream`. (a) Query-agnostic hierarchical streaming memory (§3.2) organizes incoming visual tokens into short-, mid-, and long-term memories via Jenks-based adaptive consolidation. (b) Hierarchical latent memory evolution (§3.3) retrieves and internalizes historical evidence into compact latent memory via progressively expanding memory receptive fields. (c) Progressive confidence-guided latent memory optimization (§3.4) refines the latent memory tokens with a hierarchical progression reward.

3.2 Query-agnostic Hierarchical Streaming Memory

Multi-level Streaming Memory Bank. To accommodate an ever-growing video stream within a bounded memory budget, we construct a query-agnostic hierarchical memory $\mathcal{M} = \{\mathcal{M}^g\}_{g\in\mathcal{G}}$, where $\mathcal{G} = \{s, m, l\}$ indexes the short-, mid-, and long-term levels. Following the hierarchical consolidation paradigm [54], incoming visual tokens are first stored densely in the short-term memory $\mathcal{M}^s$ to preserve recent perceptual details. As the stream evolves, historical representations are progressively migrated across memory levels through *Jenks-guided adaptive consolidation*: overflowing short-term evidence is selectively routed into $\mathcal{M}^m$ with different degrees of temporal preservation, while older mid-term evidence is consolidated into $\mathcal{M}^l$ under stronger spatial compression. Notably, this memory is constructed entirely online before the query arrives, providing a query-agnostic historical basis for the subsequent latent memory reasoning.

Jenks-Based Adaptive Memory Consolidation. To adapt memory compression to evolving redundancy patterns in streaming videos, we employ *Jenks Natural Breaks* [21] to derive data-dependent partitions directly from temporal and spatial score distributions. For short-to-mid memory transition, three-class Jenks partitioning is applied to temporal importance scores, yielding two breakpoints that divide historical evidence into `Drop`, `Compress`, and `Preserve` groups:

$$\rho_{\text{temp}}(s_i) = \begin{cases} \texttt{Drop}, & s_i < \tau_1, \\ \texttt{Compress}, & \tau_1 \le s_i < \tau_2, \\ \texttt{Preserve}, & s_i \ge \tau_2, \end{cases} \quad (\tau_1, \tau_2) = \text{Jenks}_3\left(\mathcal{S}_{\text{temp}}\right), \tag{1}$$

where $\mathcal{S}_{\text{temp}} = \{s_i\}$ is the temporal-importance distribution, and s_i is estimated from the local cosine-distance variation between a visual token and its spatial neighbors across adjacent frames following [54]. Low-score representations are discarded, intermediate ones are temporally compressed, and high-score representations are preserved. For the mid-to-long transition, we apply two-class Jenks partitioning to the spatial-distance distribution $\mathcal{S}_{\text{spat}} = \{d_{ij}\}$, where $d_{ij} = 1 - \cos(\mathbf{v}_i, \mathbf{v}_j)$ measures the feature distance between neighboring preserved tokens. Pairs assigned to the low-distance group are merged, whereas those in the high-distance group are retained individually. More details are provided in Appendix A.3.

3.3 Hierarchical Latent Memory Evolution

Although the hierarchical memory bank maintains a bounded summary of the streaming history, it remains an external and query-agnostic repository. To bridge external memory retention and query-conditioned reasoning, we introduce a set of hierarchical Latent Memory Tokens (LMTs) that

repeatedly retrieve historical evidence, interact with the retrieved visual tokens, and evolve in the latent space of a frozen MLLM.

Grouped LMTs with Expanding Memory Receptive Fields. Given the memory bank $\mathcal{M} = \{\mathcal{M}^g\}_{g\in\mathcal{G}}$, we partition the latent memory tokens at evolution iteration r into three groups:

$$\widehat{\mathbf{Z}}^{(r)} = \left[\widehat{\mathbf{Z}}_s^{(r)}; \widehat{\mathbf{Z}}_m^{(r)}; \widehat{\mathbf{Z}}_l^{(r)}\right], \qquad \widehat{\mathbf{Z}}_g^{(r)} \in \mathbb{R}^{K\times D}, \quad g \in \{s, m, l\}, \tag{2}$$

where K denotes the number of LMTs in each group, and D is the embedding dimension of the MLLM. To align the LMT hierarchy with the temporal organization of the external memory, we assign the three groups nested retrieval receptive fields:

$$\mathcal{C}^s = \mathcal{M}^s, \qquad \mathcal{C}^m = \mathcal{M}^s \cup \mathcal{M}^m, \qquad \mathcal{C}^l = \mathcal{M}^s \cup \mathcal{M}^m \cup \mathcal{M}^l. \tag{3}$$

Thus, these nested receptive fields expand the accessible history from recent observations to the complete memory bank, allowing broader-scope LMTs to integrate long-term evidence while retaining access to recent details. A bootstrap forward pass jointly contextualizes the initial LMT embeddings $\mathbf{Z}^{\text{init}}$ with the query and fixed external memory under a group-specific attention mask, yielding the query-conditioned initialization $\widehat{\mathbf{Z}}^{(0)}$ for subsequent retrieval–update iterations.

Latent-guided Historical Evidence Retrieval. As the latent memory progressively internalizes historical information, the evidence required for its subsequent evolution may change accordingly. To this end, we dynamically refresh the retrieved context at each evolution iteration. Specifically, at iteration r, each LMT group $\widehat{\mathbf{Z}}_g^{(r-1)}$ retrieves historical evidence exclusively from its corresponding memory receptive field $\mathcal{C}^g$. For each candidate visual token $\mathbf{m}_j \in \mathcal{C}^g$, we define $a_{g,j}^{(r)}$ as its maximum cosine similarity to the K LMTs in $\widehat{\mathbf{Z}}_g^{(r-1)}$. The retrieved evidence is then given by:

$$\mathcal{I}_g^{(r)} = \mathrm{TopK}_B\left(\left(a_{g,j}^{(r)}\right)_{\mathbf{m}_j\in\mathcal{C}_g}\right), \qquad \mathbf{E}_g^{(r)} = [\mathbf{m}_j]_{j\in\mathcal{I}_g^{(r)}}, \tag{4}$$

where $\mathrm{TopK}_B(\cdot)$ returns the indices of the B highest-scoring candidates, and $\mathbf{E}_g^{(r)}$ denotes the temporary evidence retrieved for group g. Although q is not directly employed as a retrieval vector, its semantics have already been encoded into $\widehat{\mathbf{Z}}^{(0)}$ through the bootstrap forward pass. The retrieval is therefore query-conditioned from the first evolution round. After each latent memory update, the relevance scores are recomputed from the evolved LMTs, allowing historical access to adapt progressively to what has already been internalized and what remains unresolved.

Group-Wise Evidence Injection and Latent Memory Updating. Given the candidate evidence $\mathbf{E}_g^{(r)}$ retrieved above, we instantiate an optimizable copy $\widetilde{\mathbf{E}}_g^{(r)}$. Let $\widehat{\mathbf{E}}_g^{(r-1)}$ denote the visual evidence pool accepted up to the previous iteration. To preserve the association between each LMT group and its retrieved evidence, we insert $\widehat{\mathbf{E}}_g^{(r-1)}$ and $\widetilde{\mathbf{E}}_g^{(r)}$ immediately after the corresponding LMT group $\widehat{\mathbf{Z}}_g^{(r-1)}$ in the candidate input $\mathbf{X}_{\text{cand}}^{(r)}$. Rather than modifying the preceding LMT activations through backward attention within the same forward pass, the retrieved evidence conditions the reward signal, whose gradient subsequently updates the optimizable LMT embeddings.

All injected evidence representations and LMT embeddings are jointly optimized under the progressive confidence objective (§3.4). Let $\mathcal{R}_{\text{cand}}^{(r)}$ denote the reward obtained with the current proposal and $\widehat{\mathcal{R}}^{(r-1)}$ the reward preserved from the previous iteration. The newly retrieved evidence is admitted into the visual evidence pool only when the candidate state yields a reward improvement:

$$\widehat{\mathbf{E}}_g^{(r)} = \begin{cases} \mathrm{Merge}\left(\widehat{\mathbf{E}}_g^{(r-1)}, \widetilde{\mathbf{E}}_g^{(r)}\right), & \text{if } \mathcal{R}_{\text{cand}}^{(r)} > \widehat{\mathcal{R}}^{(r-1)}, \\ \widehat{\mathbf{E}}_g^{(r-1)}, & \text{otherwise}, \end{cases} \tag{5}$$

where $\mathrm{Merge}(\cdot,\cdot)$ appends the newly selected visual evidence to the previously retained pool while removing duplicated entries. The reward gate controls only evidence-pool accumulation, while the updated LMTs guide the next retrieval iteration. Their optimization is detailed in §3.4.

3.4 Progressive Confidence-guided Latent Memory Optimization

Hierarchical Progression Reward Objective. The three LMT groups integrate query-relevant evidence from recent observations to the complete historical memory. However, broader retrieval may also introduce irrelevant evidence and destabilize latent evolution. We therefore exploit the frozen

Table 1: **Comparison with state-of-the-art methods on OVO-Bench [34]** (§4.2). Best results among open-source models are in **bold**, and the best results among training-free methods are underlined.

Method	Size	Frames	Real-Time Visual Perception							Backward Tracing				Overall
			OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	
Proprietary Models														
Gemini 1.5 Pro [40]	–	1 fps	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.4	52.6	62.5	65.9
GPT-4o [20]	–	64	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	62.6
Open-source Offline MLLMs														
LLaVA-Video [69]	7B	64	69.8	59.6	66.4	50.6	72.3	61.4	63.3	51.2	64.2	9.7	41.7	52.5
Qwen2-VL [46]	7B	64	69.1	53.2	63.8	50.6	66.3	60.9	60.7	44.4	**66.9**	34.4	48.6	54.6
InternVL2 [11]	8B	64	68.5	58.7	69.0	44.9	67.3	56.0	60.7	43.1	61.5	27.4	44.0	52.4
LongVU [38]	7B	1 fps	55.7	49.5	59.5	48.3	68.3	63.0	57.4	43.1	66.2	9.1	39.5	48.5
LLaVA-OneVision [26]	8B	32	66.4	57.8	73.3	**53.4**	71.3	62.0	64.0	54.2	55.4	21.5	43.7	53.9
Open-source Online MLLMs (Training-Based)														
VideoLLM-Online [5] [CVPR 2024]	8B	2 fps	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	19.3
Dispider [37] [CVPR 2025]	7B	1 fps	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4	4.3	36.1	45.3
Flash-VStream [63] [ICCV 2025]	7B	1 fps	25.5	32.1	29.3	33.7	29.7	28.8	29.9	36.4	33.8	5.9	25.4	27.6
ViSpeak [17] [ICCV 2025]	7B	1 fps	75.2	58.7	71.6	51.1	74.3	66.9	66.3	**59.9**	48.7	64.0	57.5	61.9
TimeChat-Online [60] [ACM MM 2025]	7B	1 fps	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5	9.7	41.7	51.8
StreamForest [61] [NeurIPS 2025]	7B	1 fps	68.5	53.2	71.6	47.8	65.4	60.9	61.2	58.9	64.9	32.3	52.0	56.6
Streamo [53] [CVPR 2026]	7B	1 fps	79.2	57.8	**75.0**	49.4	64.4	**70.1**	66.0	54.6	52.0	31.7	46.1	56.1
ThinkStream [32] [ECCV 2026]	3B	1 fps	85.2	64.2	69.8	49.4	69.3	64.1	67.0	53.9	59.5	43.6	52.3	59.7
Open-source Online MLLMs (Training-Free)														
Qwen2.5-VL-3B† [3]	3B	1 fps	77.2	52.3	69.0	41.0	67.3	60.9	61.3	49.8	53.4	26.3	43.2	52.2
+ FluxMem	3B	1 fps	83.2	56.9	67.2	47.8	68.3	63.6	64.5	47.5	54.7	24.2	42.1	53.3
+ `LatentStream` (Ours)	3B	1 fps	84.6	57.8	70.7	46.1	71.3	64.1	65.8	51.2	53.4	51.6	52.1	59.0 (+6.8)
Qwen2.5-VL-7B† [3]	7B	1 fps	79.2	53.2	67.2	51.7	71.3	57.1	63.3	51.5	58.8	23.7	44.7	54.0
+ FluxMem [54] [CVPR 2026]	7B	1 fps	81.2	59.6	70.7	**53.4**	75.2	63.0	67.2	50.2	62.8	26.9	46.6	56.9
+ OASIS [28] [CVPR 2026]	7B	–	85.2	**72.5**	66.4	52.3	67.3	64.7	67.3	51.9	58.8	48.9	52.6	60.0
+ `LatentStream` (Ours)	7B	1 fps	**85.9**	61.5	73.3	46.6	**78.2**	65.2	**68.5**	50.8	60.8	**68.3**	**60.0**	**64.2** (+10.2)

MLLM's predictive uncertainty as intrinsic feedback, encouraging the latent memory to become increasingly confident as its accessible historical scope expands.

For an optimizable latent state $\mathbf{U}$ at evolution iteration r, we obtain the predictive distribution $\mathbf{p}_g(\mathbf{U})$ and compute its normalized top-δ entropy:

$$\mathcal{H}_g(\mathbf{U}) = -\frac{1}{\log \delta} \sum_{j \in \Omega_g(\mathbf{U})} \bar{p}_{g,j}(\mathbf{U}) \log \bar{p}_{g,j}(\mathbf{U}), \qquad \bar{p}_{g,j}(\mathbf{U}) = \frac{p_{g,j}(\mathbf{U})}{\sum_{v \in \Omega_g(\mathbf{U})} p_{g,v}(\mathbf{U})}, \tag{6}$$

where $\Omega_g(\mathbf{U}) = \mathrm{TopK}_\delta(\mathbf{p}_g(\mathbf{U}))$ denotes the set of the δ tokens with the highest probabilities. Since the three groups incorporate increasingly comprehensive historical evidence, the desired progression is $\mathcal{H}_s(\mathbf{U}) \geq \mathcal{H}_m(\mathbf{U}) \geq \mathcal{H}_l(\mathbf{U})$. To encourage both absolute confidence and consistent hierarchical progression, we define the hierarchical progression reward as:

$$\mathcal{R}_{\text{prog}}(\mathbf{U}) = \underbrace{\sum\nolimits_{g \in \mathcal{G}} (1 - \mathcal{H}_g(\mathbf{U}))}_{\text{absolute confidence}} + \underbrace{\big(-[\mathcal{H}_m(\mathbf{U}) - \mathcal{H}_s(\mathbf{U})]_+ - [\mathcal{H}_l(\mathbf{U}) - \mathcal{H}_m(\mathbf{U})]_+\big)}_{\text{hierarchical progression}}, \tag{7}$$

where $[x]_+ = \max(0, x)$. The first term reduces predictive uncertainty for all three LMT groups, while the second penalizes violations of the desired entropy ordering without encouraging unnecessarily large gaps once the ordering is satisfied.

Confidence-guided Latent Memory Optimization. At evolution iteration r, the latent memory tokens and retrieved evidence jointly form the optimizable latent state $\mathbf{U}^{(r,0)}$. We first perturb these continuous representations by sampling a Gaussian distribution $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ and forming $\mathbf{U}' = \mathbf{U} + \boldsymbol{\xi}$, where σ controls the magnitude of exploration. This induces the Gaussian policy $\pi_\sigma(\mathbf{U}' \mid \mathbf{U}) = \mathcal{N}(\mathbf{U}, \sigma^2 \mathbf{I})$. Starting from $\mathbf{U}^{(r,0)}$, we adopt a REINFORCE-based direct policy-gradient method [49] and iteratively update the latent state as:

$$\mathbf{U} \leftarrow \mathbf{U} + \eta \widehat{\nabla}_{\mathbf{U}} \mathcal{J}(\mathbf{U}), \tag{8}$$

where η is the learning rate and $\mathcal{J}(\mathbf{U}) = \mathbb{E}_{\mathbf{U}' \sim \pi_\sigma(\cdot|\mathbf{U})}[\mathcal{R}_{\text{prog}}(\mathbf{U}')]$ denotes the expected progression reward. Following the policy-gradient theorem, its gradient can be expressed as:

$$\nabla_{\mathbf{U}} \mathcal{J}(\mathbf{U}) = \mathbb{E}_{\mathbf{U}' \sim \pi_\sigma(\cdot|\mathbf{U})} \left[\mathcal{R}_{\text{prog}}(\mathbf{U}') \nabla_{\mathbf{U}} \log \pi_\sigma(\mathbf{U}' \mid \mathbf{U})\right] = \mathbb{E}\left[\mathcal{R}_{\text{prog}}(\mathbf{U}') \frac{\boldsymbol{\xi}}{\sigma^2}\right]. \tag{9}$$

Since $\mathbf{U}'$ contains both LMTs and group-wise retrieved evidence, this estimator provides evidence-conditioned joint updates without external supervision or changes to the MLLM parameters.

After R evolution iterations, all injected evidence copies are removed, and the final answer is decoded from $\left[\mathcal{M}; q; \widehat{\mathbf{Z}}^{(R)}\right]$, allowing the compact latent memory to guide more confident reasoning.

Table 2: **Comparison with state-of-the-art methods on StreamingBench [29] and offline video benchmarks [16, 72, 51]** (§4.2). Best results among open-source models are in **bold**, and the best results among training-free methods are underlined. † indicates the reproduced results.

Method	Size	Frames	Online Video	Offline Video					
			StreamingBench	VideoMME				MLVU	LongVideoBench
			Real-Time	Short	Medium	Long	All	M-Avg	Val
Open-source Offline MLLMs									
LongVA [67]	7B	128	60.0	61.1	50.4	46.2	52.6	56.3	–
LLaVA-Video [69]	7B	1 fps	–	–	–	–	63.3	70.8	–
LLaVA-OneVision [26]	7B	32	71.1	70.1	56.4	48.8	58.4	64.7	56.5
InternVL2.5 [10]	8B	64	–	–	–	–	64.2	68.9	60.0
LongVU [38]	7B	1 fps	–	–	–	–	60.6	65.4	–
Open-source Online MLLMs (Training-Based)									
VideoLLM-Online [5] [CVPR 2024]	8B	2 fps	36.0	–	–	–	–	–	–
Dispider [37] [CVPR 2025]	7B	1 fps	67.6	–	53.7	49.7	57.2	61.7	–
Flash-VStream [63] [ICCV 2025]	7B	1 fps	23.2	72.0	61.1	50.3	61.2	–	–
TimeChat-Online [60] [ACM MM 2025]	7B	1 fps	75.3	–	–	52.4	63.3	65.4	57.7
StreamForest [61] [NeurIPS 2025]	7B	1 fps	**77.3**	–	–	–	61.9	69.6	–
ThinkStream [32] [ECCV 2026]	3B	1 fps	75.0	–	–	–	61.9	–	56.4
Open-source Online MLLMs (Training-Free)									
ReKV [12] [ICLR 2025]	7B	0.5 fps	69.1	–	–	–	–	68.5	–
StreamChat [55] [ICLR 2025]	8B	1 fps	64.7	–	–	–	–	–	–
Qwen2.5-VL† [3]	7B	1 fps	73.9	73.8	62.4	53.8	63.3	67.9	60.7
+ FluxMem [54] [CVPR 2026]	7B	1 fps	76.4	76.9	65.1	54.0	65.3	73.1	61.1
+ OASIS [28] [CVPR 2026]	7B	–	70.6	–	–	–	–	–	–
+ `LatentStream` **(Ours)**	7B	1 fps	76.9	**77.2**	**67.4**	**55.1**	**66.6**	**74.0**	**62.1**

Table 3: **Impacts of core components** on OVO-Bench [34] and VideoMME [16] (§4.3).

HSM	PMO	HME	OVO-Bench	VideoMME
			54.0	63.3
✓			58.1	65.1
✓	✓		62.4	66.1
✓	✓	✓	**64.2**	**66.6**

Table 4: **Ablation of retrieve-and-internalize mechanism** on OVO-Bench [34] and VideoMME [16] (§4.3).

Method	OVO-Bench	VideoMME
HSM *Only*	58.1	65.1
+ Retrieved Visual Evidence	59.7	65.6
+ Initial Latent Memory Tokens	56.5	65.2
+ Evolved Latent Memory Tokens (**Ours**)	**64.2**	**66.6**

4 Experiment

4.1 Experiment Setup

Benchmarks. We evaluate `LatentStream` on two streaming video benchmarks and three offline long-video benchmarks. OVO-Bench [34] evaluates timestamp-aware streaming video understanding, covering historical retrieval, real-time perception, and proactive response. StreamingBench [29] evaluates real-time visual, omni-source, and contextual understanding over continuous video streams. Video-MME [16], MLVU [72], and LongVideoBench [51] evaluate perception, retrieval, and reasoning across varying video durations and temporal granularities.

Implementation Details. We build `LatentStream` on Qwen2.5-VL-3B and 7B backbones [3]. For online benchmarks, videos are sampled at 1 fps with at most 256 frames. The short- and mid-term memory capacities are set to 8 and 64 frames, respectively, while earlier observations are consolidated into long-term memory within a global budget of 2048 visual tokens. Each memory group uses $K=2$ latent memory tokens, with $B=8$ visual candidate patches injected at each iteration. We perform $R=4$ evolution iterations with a learning rate of $\eta = 1 \times 10^{-3}$ and a Gaussian perturbation scale of $\sigma = 10\%$. For offline benchmarks, videos are sampled at the same frame rate with 64 visual tokens per frame and at most $1,024$ frames. All experiments are conducted on eight NVIDIA H20 GPUs. Further details are provided in Appendix A.4.

4.2 Comparison with State-of-the-Arts

Results on Streaming Video Benchmarks. As shown in Tables 1 and 2, `LatentStream` consistently improves Qwen2.5-VL [3] on both benchmarks with the backbones frozen. On OVO-Bench [34], our `LatentStream` (7B) raises RT/BW from 63.3%/44.7% to **68.5%/60.0%**, achieving the best overall score among open-source methods (**+10.2%**). On StreamingBench [29], it achieves **76.9%** (**+3.0%**), outperforming all compared training-free methods. These gains further transfer to Qwen3-VL-8B [2] (Fig. 5 (a)), surpassing both the baseline and OASIS [28] across all metrics, particularly

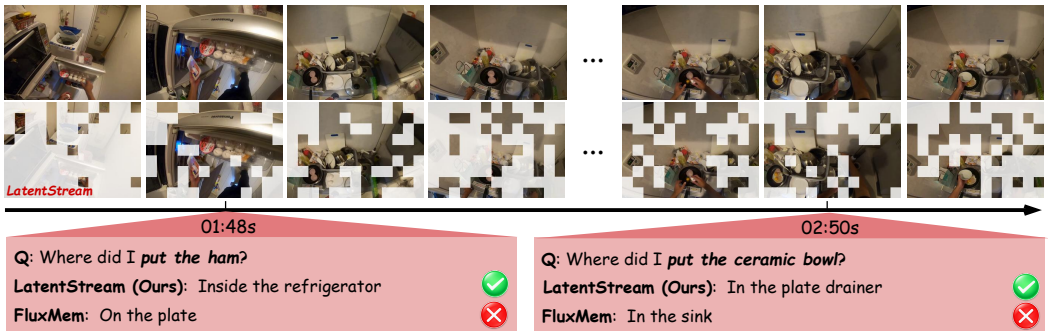


Figure 3: **Qualitative results and token compression visualization** on OVO-Bench [34] (§4.2).

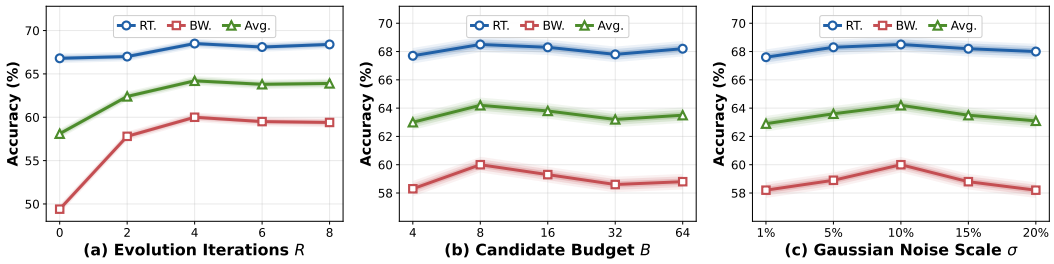


Figure 4: **Hyperparameter analysis on OVO-Bench [34]** (§4.3). (a) Effect of the number of evolution iterations R. (b) Effect of the candidate budget B. (c) Effect of Gaussian noise scale σ.

on OVO-Bench BW. The consistent improvements across model scales and backbones demonstrate the effectiveness of `LatentStream`. Detailed results are provided in Appendix E.

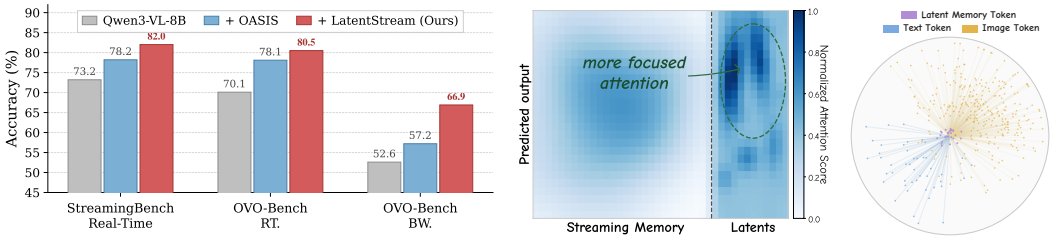
Results on Offline Video Benchmarks. Although designed for streaming video understanding, `LatentStream` generalizes well to offline long-video settings under a bounded memory budget. As shown in Table 2, it achieves the best results on VideoMME [16], MLVU [72], and LongVideoBench [51], outperforming both training-free and training-based methods. On VideoMME, the improvements consistently span short, medium, and long videos, demonstrating the effectiveness of hierarchical streaming memory and progressive latent memory evolution in preserving and internalizing task-relevant historical evidence beyond online settings.

Qualitative Results. Fig. 3 presents two representative backward-tracing cases on OVO-Bench [34] using Qwen3-VL-8B [2]. `LatentStream` correctly recovers the earlier object placements, whereas FluxMem predicts plausible but incorrect locations. The retained visual tokens further show that our hierarchical streaming memory preserves sparse spatial evidence under continual compression, enabling reliable long-range reasoning through progressive latent memory evolution.

4.3 Ablation Study

Key Component Analysis. Table 3 evaluates the three core components of `LatentStream`. Query-agnostic Hierarchical Streaming Memory (HSM) raises OVO-Bench [34]/VideoMME [16] from 54.0%/63.3% to **58.1%/65.1%**. Adding Progressive Confidence-guided Latent Memory Optimization (PMO) improves performance to **62.4%/66.1%** without retrieved visual evidence, while Hierarchical Latent Memory Evolution (HME) further reaches **64.2%/66.6%** by incorporating evidence from expanding memory scopes, confirming the complementarity of the three components.

Effectiveness of Retrieve-and-Internalize Latent Memory. To distinguish latent internalization from simple context augmentation, Table 4 contrasts four ways of exploiting historical information. Compared with the baseline conditioned only on Hierarchical Streaming Memory and the query, directly appending retrieved visual evidence following [66, 28] yields limited improvements. Introducing unoptimized latent memory tokens (LMTs) provides no improvement, indicating that the gains cannot be attributed to simply adding latent tokens. In contrast, evolved LMTs achieve the best performance on both benchmarks. These results demonstrate that retrieved evidence is effectively internalized into a compact latent working memory for streaming reasoning.



(a) (b) (c)

Figure 5: **Analysis of LatentStream** (§4.3). (a) **Generalization to Qwen3-VL-8B** on StreamingBench [29] and OVO-Bench [34]. (b) **Attention visualization** on an OVO-Bench [34] sample. Optimized latent memory tokens receive more focused attention. (c) **Visualization of the latent representation space** showing the geometric separation of latents, text tokens, and image tokens.

Table 5: **Ablation of progressive confidence-guided optimization** on OVO-Bench [34] and VideoMME [16] (§4.3).

Confidence Objective	OVO-Bench	VideoMME
w/o Latent Memory Optimization	58.1	65.1
Absolute Confidence Optimization *only*	62.5	65.5
Hierarchical Progression Reward (**Ours**)	**64.2**	**66.6**

Effect of Progressive Confidence-guided Latent Memory Optimization (PMO). To study the effectiveness of PMO (*cf.* Eq. 7), Table 5 compares it against no latent memory optimization and absolute confidence optimization alone. Absolute confidence optimization improves both benchmarks [34, 16], showing that predictive confidence provides effective self-supervision for refining latent memory. However, it optimizes each memory group independently. By further penalizing violations of the entropy ordering $\mathcal{H}_s \geq \mathcal{H}_m \geq \mathcal{H}_l$, our progression reward improves OVO-Bench and VideoMME by **1.7** and **1.1** points, confirming that confidence progression provides stronger test-time supervision for latent memory evolution.

Number of Evolution Iterations R. Fig. 4 (a) studies the effect of the evolution iteration number R. Performance improves consistently up to $R=4$, validating the effectiveness of retrieval–update optimization for latent memory evolution. Further iterations yield no additional benefit and introduce minor fluctuations. We therefore set $R=4$ to balance performance and efficiency.

Candidate Budget B. Fig. 4 (b) examines the candidate budget B at each iteration. Performance peaks at $B=8$ and declines with larger budgets, suggesting that a compact set of relevant evidence is sufficient to guide latent memory evolution, while excessive candidates hinder optimization.

Gaussian Noise Scale σ. Fig. 4 (c) studies the perturbation scale σ used for latent exploration. Increasing σ initially improves performance by enabling broader exploration, whereas excessive perturbations destabilize optimization. We therefore set $\sigma = 10\%$, which provides the best balance.

Attention Visualization. Fig. 5 (b) visualizes the attention patterns on an OVO-Bench [34] example. As expected, the optimized Latent Memory Tokens (LMTs) attract stronger attention from the predicted answer tokens than the visual tokens in the Hierarchical Streaming Memory (HSM). This observation indicates that confidence-guided optimization effectively activates LMTs.

Latent Representation Analysis. We visualize the final distributions of text tokens, visual tokens, and Latent Memory Tokens (LMTs) using t-SNE [41] to examine the effect of iterative optimization. As shown in Fig. 5 (c), the optimized LMTs form a tight cluster, well separated from both textual and visual embeddings. This distribution suggests that the optimized latent tokens become modality-independent, forming a unified cross-modal semantic representation.

5 Conclusion

In this work, we introduced `LatentStream`, a progressive latent working memory framework for streaming video understanding. Moving beyond conventional store-and-retrieve memory, `LatentStream` progressively internalizes task-relevant historical evidence into a compact latent memory that guides streaming reasoning. It integrates Hierarchical Streaming Memory, Hierarchical Latent Memory Evolution, and Progressive Confidence-guided Latent Memory Optimization. Extensive experiments across streaming and offline video benchmarks demonstrate the effectiveness of `LatentStream`. These results establish retrieve-and-internalize memory as a promising direction for bridging external memory with latent reasoning in streaming Video-LLMs.

AI USE STATEMENT

We used generative AI tools to assist with language proofreading, improve the presentation of the manuscript, support parts of the coding process, and aid in retrieval and discovery, such as identifying potentially relevant related work. Generative AI tools were not used to generate experimental results, perform data analysis, or draw scientific conclusions. All AI-assisted content was carefully reviewed and verified by the authors, who take full responsibility for the content of this manuscript.

REFERENCES

[1] Shehreen Azad, Vibhav Vineet, and Yogesh S Rawat. Streamready: Learning what to answer and when in long streaming videos. In *CVPR*, pp. 40494–40504, 2026. 3

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 7, 8, 21, 22

[3] 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. 1, 6, 7, 17, 20, 21, 22

[4] Tim Brödermann, Christos Sakaridis, Yuqian Fu, and Luc Van Gool. Cafuser: Condition-aware multimodal fusion for robust semantic perception of driving scenes. *IEEE Robotics and Automation Letters*, 10(4):3134–3141, 2025. 1

[5] 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 *CVPR*, pp. 18407–18418, 2024. 6, 7, 21, 22

[6] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024. 1, 23

[7] Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025. 2

[8] Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding. In *CVPR*, pp. 24675–24685, 2026. 3

[9] Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. In *AAAI*, volume 40, pp. 3120–3128, 2026. 1

[10] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 7

[11] 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, 2024. 6, 21

[12] Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, ZhongTao, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *ICLR*, volume 2025, pp. 42115–42127, 2025. 1, 3, 7, 22

[13] Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Qianxi Zhang, Donglin Bai, Zhibo Chen, and Ting Cao. Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. In *ICCV*, pp. 13448–13459, 2025. 3

[14] Evangelos Dorovatas, Soroush Seifi, Gunshi Gupta, and Rahaf Aljundi. Recurrent attention-based token selection for efficient streaming video-llms. In *NeurIPS*, pp. 144088–144114, 2025. 1

[15] Estera Dumitru and Stelian Spînu. A multi-task deep learning framework for real-time intelligent video surveillance with temporal event validation. *arXiv preprint arXiv:2607.03131*, 2026. 1

[16] 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 *CVPR*, pp. 24108–24118, 2025. 2, 7, 8, 9, 15, 21

[17] Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. Vispeak: Visual instruction feedback in streaming videos. In *ICCV*, pp. 21778–21788, 2025. 6, 21

[18] 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. *arXiv preprint arXiv:2603.12262*, 2026. 1, 3

[19] Zhenghui Guo, Yuanbin Man, Junyuan Sheng, Bowen Lin, Ahmed Ahmed, Bo Jiang, Boyuan Zhang, Miao Yin, Sian Jin, Omprakash Gnawali, et al. Event-vstream: Event-driven real-time understanding for long video streams. In *CVPR*, pp. 3060–3069, 2026. 3

[20] 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. 6, 21, 23

[21] George F Jenks and Fred C Caspall. Error on choroplethic maps: definition, measurement, reduction. *Annals of the Association of American Geographers*, 61(2):217–244, 1971. 4, 20

[22] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. 19, 23

[23] Junho Kim, Hosu Lee, James M Rehg, Minsu Kim, and Yong Man Ro. Stride: When to speak meets sequence denoising for streaming video understanding. *arXiv preprint arXiv:2603.27593*, 2026. 3

[24] Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. In *NeurIPS*, pp. 138983–139013, 2025. 3

[25] Lik-Hang Lee and Pan Hui. Interaction methods for smart glasses: A survey. *IEEE access*, 6: 28712–28732, 2018. 1, 23

[26] 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. 1, 6, 7, 21, 22

[27] Ruanjun Li, Yuedong Tan, Yuanming Shi, and Jiawei Shao. Videoscan: Enabling efficient streaming video understanding via frame-level semantic carriers. *arXiv preprint arXiv:2503.09387*, 2025. 3

[28] 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. 1, 2, 3, 6, 7, 8, 21, 22

[29] 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*, pp. 12147–12151, 2026. 2, 7, 9, 15, 21, 22

[30] Chengzhi Liu, Yuzhe Yang, Yue Fan, Qingyue Wei, Sheng Liu, and Xin Eric Wang. Reasoning within the mind: Dynamic multimodal interleaving in latent space. In *CVPR*, pp. 9225–9236, 2026. 2, 19

[31] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 2025. 1, 23

[32] 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. *arXiv preprint arXiv:2603.12938*, 2026. 1, 3, 6, 7, 21, 22

[33] Zhenyu Ning, Guangda Liu, Qihao Jin, Chengwei Li, Wenchao Ding, Minyi Guo, and Jieru Zhao. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. *arXiv preprint arXiv:2505.15269*, 2025. 3

[34] 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 *CVPR*, pp. 18902–18913, 2025. 2, 6, 7, 8, 9, 15, 20, 21, 22

[35] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 20

[36] Zhanzhong Pang, Dibyadip Chatterjee, Fadime Sener, and Angela Yao. Decouple and cache: Kv cache construction for streaming video understanding. *arXiv preprint arXiv:2605.01858*, 2026. 1

[37] 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 *CVPR*, pp. 24045–24055, 2025. 6, 7, 21, 22

[38] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024. 6, 7, 21

[39] Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *arXiv preprint arXiv:2604.02317*, 2026. 1

[40] 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. 6, 21

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 9

[42] Chao Wang, Xudong Tan, Jianjian Cao, Kangcong Li, and Tao Chen. Curvestream: Boosting streaming video understanding in mllms via curvature-aware hierarchical visual memory management. *arXiv preprint arXiv:2603.19571*, 2026. 3

[43] Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. Streambridge: Turning your offline video large language model into a proactive streaming assistant. In *NeurIPS*, volume 38, pp. 132332–132359, 205. 3

[44] Kuanning Wang, Ke Fan, Yuqian Fu, Siyu Lin, Hu Luo, Daniel Seita, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Ocra: Object-centric learning with 3d and tactile priors for human-to-robot action transfer. *arXiv preprint arXiv:2603.14401*, 2026. 1

[45] Lu Wang, Zhuoran Jin, Yupu Hao, Yubo Chen, Kang Liu, Yulong Ao, and Jun Zhao. Think while watching: Online streaming segment-level memory for multi-turn video reasoning in multimodal large language models. *arXiv preprint arXiv:2603.11896*, 2026. 3

[46] 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. 1, 6, 21

[47] Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *CVPR*, pp. 12030–12040, 2026. 2

[48] 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 *CVPR*, pp. 18523–18533, 2026. 3

[49] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. 6, 18

[50] 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. 1, 2, 3

[51] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *NeurIPS*, volume 37, pp. 28828–28857, 2024. 2, 7, 8, 15

[52] Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. In *NeurIPS*, volume 37, pp. 109922–109947, 2024. 3

[53] Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. In *CVPR*, pp. 31219–31229, 2026. 3, 6, 21

[54] 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. 1, 2, 3, 4, 6, 7, 21, 22

[55] 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 *ICLR*, volume 2025, pp. 69332–69351, 2025. 1, 3, 7, 22

[56] Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. In *ICLR*, volume 2026, pp. 61463–61475, 2026. 1

[57] Weicai Yan, Yuhong Dai, Qi Ran, Haodong Li, Wang Lin, Tao Jin, Xing Xie, Hao Liao, and Jianxun Lian. Proact-vl: A proactive videollm for real-time ai companions. *arXiv preprint arXiv:2603.03447*, 2026. 3

[58] 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. *arXiv preprint arXiv:2508.15717*, 2025. 3

[59] Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *CVPR*, pp. 33510–33520, 2026. 2

[60] 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 *ACM MM*, pp. 10807–10816, 2025. 1, 3, 6, 7, 21, 22

[61] 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. In *NeurIPS*, volume 38, pp. 75804–75835, 2026. 6, 7, 21, 22

[62] Deheng Zhang, Yuqian Fu, Runyi Yang, Yang Miao, Tianwen Qian, Xu Zheng, Guolei Sun, Ajad Chhatkuli, Xuanjing Huang, Yu-Gang Jiang, et al. Egonight: Towards egocentric vision understanding at night with a challenging benchmark. In *ICLR*, pp. 887–901, 2026. 1

[63] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *ICCV*, pp. 21059–21069, 2025. 6, 7, 21, 22

[64] Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *ACL*, pp. 8411–8430, 2026. 3

[65] Jialiang Zhang, Junlong Tong, Junyan Lin, Hao Wu, Yirong Sun, Yunpu Ma, and Xiaoyu Shen. Think-as-you-see: Streaming chain-of-thought reasoning for large vision-language models. *arXiv preprint arXiv:2603.02872*, 2026. 1, 3

[66] Kairui Zhang, Zhenyu Yang, Bing Wang, Shengsheng Qian, and Changsheng Xu. Querystream: Advancing streaming video understanding with query-aware pruning and proactive response. In *ICLR*, 2026. 8

[67] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. URL https://arxiv.org/abs/2406.16852. 7, 22

[68] Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. Proactive assistant dialogue generation from streaming egocentric videos. In *EMNLP*, pp. 12055–12079, 2025. 3

[69] 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. 1, 6, 7, 21

[70] Yulin Zhang, Cheng Shi, and Sibei Yang. Weavetime: Stream from earlier frames into emergent memory in videollms. *arXiv preprint arXiv:2602.22142*, 2026. 3

[71] Zicheng Zhao, Kangyu Wang, Shijie Li, Rui Qian, Weiyao Lin, and Huabin Liu. Cogstream: Context-guided streaming video question answering. In *AAAI*, volume 40, pp. 13332–13341, 2026. 3

[72] 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 *CVPR*, pp. 13691–13701, 2025. 2, 7, 8, 15

SUMMARY OF THE APPENDIX

This supplementary material provides additional details and analysis for the ICLR 2027 submission, titled *"Beyond Retrieval: Progressive Latent Memory Evolution for Streaming Video Understanding"*. The appendix is organized as follows:

- §A provides benchmark, hierarchical memory construction, and implementation details.
- §B presents the pseudo-code of progressive latent memory optimization.
- §C provides theoretical analyses of latent optimization and evidence internalization.
- §D reports additional quantitative results and ablation studies.
- §E provides the full results on streaming video benchmarks.
- §F presents additional qualitative case studies.
- §G discusses limitations, future work, and broader impact.

A EXPERIMENTAL DETAILS

A.1 BENCHMARK DESCRIPTIONS

OVO-Bench. OVO-Bench [34] contains 644 videos and approximately 3,100 timestamped queries for online video understanding. Its 12 tasks cover Real-Time Visual Perception, Backward Tracing, and Forward Active Responding. Following prior memory-oriented evaluations, we report the mean performance over the Real-Time and Backward categories, while excluding the response-timing-oriented Forward category.

StreamingBench. StreamingBench [29] comprises 900 videos and 4,500 human-annotated question–answer pairs across 18 tasks. It evaluates real-time visual, omni-source, and contextual understanding, with questions presented at different timestamps during each video. We report accuracy on its Real-Time Visual Understanding subset.

Video-MME. Video-MME [16] contains 900 videos totaling 254 hours and 2,700 multiple-choice questions across six domains and 30 subdomains. The videos range from 11 seconds to one hour and are divided into short, medium, and long subsets. We adopt the setting without subtitles and report both duration-wise and overall accuracy.

MLVU. MLVU [72] consists of 1,730 videos and 3,102 questions, with durations ranging from 3 minutes to over 2 hours. Its nine tasks assess holistic, single-detail, and multi-detail long-video understanding. We report M-Avg, the mean accuracy over its seven multiple-choice tasks.

LongVideoBench. LongVideoBench [51] contains 3,763 videos and 6,678 multiple-choice questions, with videos lasting up to one hour. It evaluates long-context retrieval and reasoning through 17 categories of referring-reasoning questions over interleaved video and subtitle inputs. We report accuracy on the public validation split containing 752 videos and 1,337 questions.

A.2 PROBLEM FORMULATION

Let $\mathcal{V} = \{I_t\}_{t=1}^{\infty}$ denote a potentially unbounded video stream, where I_t is the visual observation arriving at timestamp t. In the *online streaming video question answering* (StreamingVQA) task, a natural-language query q may be issued at an arbitrary timestamp T. Given the observed video prefix $\mathcal{V}_{\le T} = \{I_t\}_{t=1}^{T}$, a streaming video model π_θ is required to predict an answer:

$$\widehat{\mathbf{a}} = \arg\max_{\mathbf{a}} p_{\pi_\theta}\left(\mathbf{a} \mid q, \mathcal{V}_{\le T}\right). \tag{10}$$

Unlike offline video question answering, StreamingVQA is governed by two defining constraints: (i) *causal observation*, where the model can only access observations up to timestamp T without revisiting future content; and (ii) *post-hoc querying*, where q is revealed only at timestamp T, requiring the preceding stream to be processed without foreknowledge of the eventual query. Meanwhile, as the observed history grows indefinitely, the system must operate under a fixed resource budget that does not scale with the stream duration. The objective is therefore to preserve sufficient historical information for accurate answer generation while satisfying the causal, post-hoc, and resource-bounded requirements of streaming video processing.

A.3 Hierarchical Streaming Memory Construction Details

Online Hierarchical Organization. We construct the streaming memory online as $\mathcal{M} = \{\mathcal{M}^g\}_{g\in\mathcal{G}}$, where $\mathcal{G} = \{s, m, l\}$ denotes the short-, mid-, and long-term levels. Incoming visual tokens are stored densely in $\mathcal{M}^s$. Upon overflow, the oldest short-term grid is temporally routed into $\mathcal{M}^m$, while the oldest overflowing mid-term grid is spatially consolidated into $\mathcal{M}^l$. These first-in-first-out transitions occur before the query arrives, ensuring query-agnostic construction. For the model with temporal patch size $p_t = 2$, frame capacities N_s and N_m correspond to $C_s = \lfloor N_s/p_t \rfloor$ and $C_m = \lfloor N_m/p_t \rfloor$ temporal grids.

Global Budget Enforcement. Beyond these per-level capacities, we enforce a global budget of $B_{\text{mem}} = 2{,}048$ visual tokens after each memory update. Whenever $\sum_{g\in\mathcal{G}} |\mathcal{M}^g| > B_{\text{mem}}$, we iteratively discard the lowest-scoring tokens from $\mathcal{M}^l$ according to the temporal-importance score in Eq. 11 until the budget is restored. For a spatially merged token, we retain the score of its anchor i^* in Eq. 16. This operation leaves the short- and mid-term memories unchanged and confines budget-driven forgetting to already consolidated history.

Short-to-Mid Temporal Routing. Let $\mathbf{v}_{t,p}$ be the visual token at temporal grid t and position p, and let $\bar{\mathbf{v}}$ be its ℓ_2-normalized feature. For each token leaving $\mathcal{M}^s$, we search its 3×3 neighborhood in the preceding and succeeding grids. Writing $i = (t, p)$, we compute

$$\begin{aligned} d^{\pm}_{t,p} &= 1 - \max_{p'\in\mathcal{N}_{3\times3}(p)} \bar{\mathbf{v}}^{\top}_{t,p}\bar{\mathbf{v}}_{t\pm1,p'}, \\ s_i \equiv s_{t,p} &= \max\big(d^{-}_{t,p}, d^{+}_{t,p}\big)\,. \end{aligned} \tag{11}$$

The backward distance is cached upon entry and the forward distance is computed upon eviction; their maximum highlights tokens that differ from either adjacent grid.

Given $\mathcal{S}_{\text{temp}} = \{s_i\}$, three-class Jenks partitioning produces two data-dependent breakpoints:

$$\begin{aligned} \rho_{\text{temp}}(s_i) &= \begin{cases} \texttt{Drop}, & s_i < \tau_1, \\ \texttt{Compress}, & \tau_1 \le s_i < \tau_2, \\ \texttt{Preserve}, & s_i \ge \tau_2, \end{cases} \\ (\tau_1, \tau_2) &= \text{Jenks}_3(\mathcal{S}_{\text{temp}})\,. \end{aligned} \tag{12}$$

`Drop` removes low-score tokens and `Preserve` keeps high-score tokens unchanged. For `Compress`, let p^* be the most similar position in the succeeding grid. We update

$$\widetilde{\mathbf{v}}_{t,p} = \tfrac{1}{2}\left(\mathbf{v}_{t,p} + \mathbf{v}_{t+1,p^*}\right). \tag{13}$$

The fused token remains at (t, p) and the matched token is unchanged. A forward similarity above $\gamma_{\text{sim}} = 0.8$ directly triggers `Drop`; missing neighbors or degenerate Jenks partitions trigger conservative preservation.

Mid-to-Long Spatial Clustering. For an overflowing mid-term grid, we construct an eight-neighbor graph over the surviving tokens. Each spatially adjacent pair (i, j) forms a candidate edge with

$$d_{ij} = 1 - \cos(\mathbf{v}_i, \mathbf{v}_j), \qquad \mathcal{S}_{\text{spat}} = \{d_{ij}\}. \tag{14}$$

Two-class Jenks partitioning minimizes within-class variation and produces an adaptive breakpoint τ_s:

$$\begin{aligned} \rho_{\text{spat}}(d_{ij}) &= \begin{cases} \texttt{Merge}, & d_{ij} < \tau_s, \\ \texttt{Separate}, & d_{ij} \ge \tau_s, \end{cases} \\ \tau_s &= \text{Jenks}_2(\mathcal{S}_{\text{spat}})\,. \end{aligned} \tag{15}$$

We retain only low-distance `Merge` edges; their connected components define clusters, while high-distance edges separate distinct regions. For each non-singleton cluster $\mathcal{C}$, we compute its mean and select the closest token as the anchor:

$$\begin{aligned} \boldsymbol{\mu}_{\mathcal{C}} &= \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \mathbf{v}_i, \\ i^* &= \arg\min_{i\in\mathcal{C}} \|\mathbf{v}_i - \boldsymbol{\mu}_{\mathcal{C}}\|_2^2\,. \end{aligned} \tag{16}$$

The anchor remains at its original position with feature $\boldsymbol{\mu}_{\mathcal{C}}$; other cluster members are discarded and singletons remain unchanged. Surviving tokens retain their original temporal order and positional encodings.

Table 6: **Implementation and inference hyperparameters of `LatentStream`** (§A.4).

Hyperparameter	Value
Backbone	Qwen2.5-VL and Qwen3-VL
Video sampling rate	1 fps
Maximum online frames	256
Maximum offline frames	1,024
Offline visual tokens per frame	64
Temporal patch size p_t	2
Short-term memory capacity N_s	8 frames
Mid-term memory capacity N_m	64 frames
Global memory budget	2,048 visual tokens
Temporal Jenks partition	3 classes
Spatial Jenks partition	2 classes
Number of LMT groups	3
LMTs per group K	2
Total number of LMTs	6
LMT initialization	`<|endoftext|>`
Evolution iterations R	4
Candidate budget B	8
Entropy candidate size δ	10
Gaussian noise scale σ	0.10
Noise decay factor γ_{noise}	0.95
Learning rate η	1×10^{-3}
Hardware	8× NVIDIA H20

A.4 More Implementation Details

We implement **`LatentStream`** using the official Qwen2.5-VL and Qwen3-VL backbones [3], with both the visual encoder and language model kept frozen throughout inference. For online benchmarks, videos are uniformly sampled at 1 fps with at most 256 frames, whereas offline videos contain at most 1,024 sampled frames and are represented using 64 visual tokens per frame. The short- and mid-term memory capacities are set to 8 and 64 frames, respectively, and the overall streaming memory is bounded to 2,048 visual tokens. We divide the latent memory tokens (LMTs) into three groups, each containing $K = 2$ tokens, and perform $R = 4$ evolution iterations. At each iteration, every group retrieves $B = 8$ visual candidates from its corresponding memory receptive field. The latent states are optimized with a learning rate of $\eta = 1 \times 10^{-3}$ and Gaussian perturbation scale $\sigma = 0.10$. Only the LMT embeddings and temporary copies of retrieved evidence are optimized; all retrieved evidence is removed before final answer generation. The LMTs are initialized anew for each query, optimized only during the corresponding query-answering process, and discarded after the answer is generated. Unless otherwise specified, the same configuration is used across all benchmarks and backbones. Detailed hyperparameters are summarized in Table 6.

B Pseudo-code of Latent Memory Optimization Procedure

In this work, we introduce **`LatentStream`**, a progressive latent working memory framework for streaming video understanding, as summarized in Algorithm 1. By integrating query-agnostic hierarchical streaming memory, hierarchical latent memory evolution, and progressive confidence-guided optimization, **`LatentStream`** progressively internalizes task-relevant historical evidence into compact latent memory tokens. The entire procedure is performed at test time with the Video-LLM frozen, while reward-improving evidence is selectively retained to guide subsequent memory evolution. Extensive experiments demonstrate that **`LatentStream`** consistently improves both streaming and offline video understanding, establishing progressive latent memory evolution as an effective alternative to conventional store-and-retrieve memory. To guarantee reproducibility, our code will be released.

Algorithm 1 `LatentStream`: Progressive Latent Memory Optimization

Input: Frozen Video-LLM π_θ, hierarchical memory $\mathcal{M}$, query q, initial LMTs $\mathbf{Z}^{\text{init}}$, iterations R, LMT number K, candidate budget B, entropy size δ, and hyperparameters $(\eta, \sigma, \gamma_{\text{noise}})$
Define $\mathcal{G} = \{s, m, l\}$, $\mathcal{C}^s = \mathcal{M}^s$, $\mathcal{C}^m = \mathcal{M}^s \cup \mathcal{M}^m$, and $\mathcal{C}^l = \mathcal{M}^s \cup \mathcal{M}^m \cup \mathcal{M}^l$ ▷ expand historical access across groups
$\widehat{\mathbf{Z}}^{(0)} \leftarrow \text{Bootstrap}_{\pi_\theta}(\mathcal{M}, q, \mathbf{Z}^{\text{init}}; \{\mathcal{C}^g\}_{g\in\mathcal{G}})$ ▷ query-conditioned initialization with group-specific masks
$\widehat{\mathbf{E}}_g^{(0)} \leftarrow \varnothing,\ \forall g;\ \widehat{\mathcal{R}}^{(0)} \leftarrow \mathcal{R}_{\text{prog}}(\widehat{\mathbf{Z}}^{(0)})$ ▷ initialize evidence pools and admission threshold
for $r = 1, \ldots, R$ **do**
  $\sigma_r \leftarrow \sigma\gamma_{\text{noise}}^{r-1}$ ▷ anneal the exploration noise
  for all $g \in \mathcal{G}$ **do**
    $a_{g,j}^{(r)} \leftarrow \max_{k\in\{1,\ldots,K\}} \cos(\widehat{\mathbf{z}}_{g,k}^{(r-1)}, \mathbf{m}_j),\ \forall \mathbf{m}_j \in \mathcal{C}^g;\ \mathcal{I}_g^{(r)} \leftarrow \text{TopK}_B(\{a_{g,j}^{(r)}\}_j)$ ▷ retrieve evidence relevant to the evolved LMTs
    $\widetilde{\mathbf{E}}_g^{(r)} \leftarrow \text{Clone}([\mathbf{m}_j]_{j\in\mathcal{I}_g^{(r)}})$ ▷ create optimizable evidence copies
  end for
  $\mathbf{U}^{(r)} \leftarrow \text{Pack}_{g\in\mathcal{G}}(\widehat{\mathbf{Z}}_g^{(r-1)}, \widehat{\mathbf{E}}_g^{(r-1)}, \widetilde{\mathbf{E}}_g^{(r)})$ ▷ place evidence after its corresponding LMT group
  Sample $\boldsymbol{\xi}^{(r)} \sim \mathcal{N}(\mathbf{0}, \sigma_r^2\mathbf{I})$; $\mathbf{U}' \leftarrow \mathbf{U}^{(r)} + \boldsymbol{\xi}^{(r)}$ ▷ explore a neighboring latent state
  Compute group-wise entropies $\{\mathcal{H}_g(\mathbf{U}')\}_{g\in\mathcal{G}}$ and hierarchical progression reward $\mathcal{R}_{\text{noise}}^{(r)} \leftarrow \mathcal{R}_{\text{prog}}(\mathbf{U}')$ ▷ measure absolute and hierarchical confidence
  $\widehat{\nabla}_{\mathbf{U}}\mathcal{J} \leftarrow \mathcal{R}_{\text{noise}}^{(r)}\boldsymbol{\xi}^{(r)}/\sigma_r^2$; $\mathbf{U}_{\text{cand}}^{(r)} \leftarrow \mathbf{U}^{(r)} + \eta\widehat{\nabla}_{\mathbf{U}}\mathcal{J}$ ▷ jointly optimize LMTs and retrieved evidence
  $(\mathbf{Z}_{\text{cand}}^{(r)}, \{\mathbf{E}_{g,\text{old}}^{(r)}, \mathbf{E}_{g,\text{new}}^{(r)}\}_{g\in\mathcal{G}}) \leftarrow \text{Unpack}(\mathbf{U}_{\text{cand}}^{(r)});\ \mathcal{R}_{\text{cand}}^{(r)} \leftarrow \mathcal{R}_{\text{prog}}(\mathbf{U}_{\text{cand}}^{(r)})$ ▷ evaluate the optimized candidate
  $\widehat{\mathbf{Z}}^{(r)} \leftarrow \mathbf{Z}_{\text{cand}}^{(r)}$ ▷ continue latent optimization without greedy rollback
  if $\mathcal{R}_{\text{cand}}^{(r)} > \widehat{\mathcal{R}}^{(r-1)}$ **then**
    $\widehat{\mathbf{E}}_g^{(r)} \leftarrow \text{Merge}(\mathbf{E}_{g,\text{old}}^{(r)}, \mathbf{E}_{g,\text{new}}^{(r)}),\ \forall g;\ \widehat{\mathcal{R}}^{(r)} \leftarrow \mathcal{R}_{\text{cand}}^{(r)}$ ▷ retain reward-improving evidence
  else
    $\widehat{\mathbf{E}}_g^{(r)} \leftarrow \widehat{\mathbf{E}}_g^{(r-1)},\ \forall g;\ \widehat{\mathcal{R}}^{(r)} \leftarrow \widehat{\mathcal{R}}^{(r-1)}$ ▷ discard non-improving evidence
  end if
end for
$\widehat{\mathbf{a}} \leftarrow \text{Decode}_{\pi_\theta}([\mathcal{M}; q; \widehat{\mathbf{Z}}^{(R)}])$ ▷ remove all retrieved evidence before generation
return $\widehat{\mathbf{a}}$

C Theoretical Analysis

We characterize four properties of `LatentStream`: the stochastic optimization behavior of the confidence objective, the bounded structure of hierarchical progression supervision, the capacity induced by nested memory receptive fields, and the effect of retrieved evidence on latent updating. These analyses concern the surrogate objective and the underlying optimization mechanism; they do not provide guarantees of answer correctness or optimal evidence retrieval.

Proposition 1 (Expected change of the smoothed confidence objective). Let $\mathbf{U}$ denote the jointly optimizable LMT–evidence state. We define the Gaussian-smoothed progression objective and its stochastic estimator as

$$J_\sigma(\mathbf{U}) = \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}\left[\mathcal{R}_{\text{prog}}(\mathbf{U}+\boldsymbol{\xi})\right],$$
$$\widehat{\mathbf{g}} = \mathcal{R}_{\text{prog}}(\mathbf{U}+\boldsymbol{\xi})\frac{\boldsymbol{\xi}}{\sigma^2}. \tag{17}$$

The score-function identity [49] gives

$$\mathbb{E}\left[\widehat{\mathbf{g}}\right] = \nabla J_\sigma(\mathbf{U}). \tag{18}$$

Suppose that J_σ is L_J-smooth and that

$$\mathbb{E}\left[\left\|\widehat{\mathbf{g}} - \nabla J_\sigma(\mathbf{U})\right\|^2\right] \le \nu^2. \tag{19}$$

For the update $\mathbf{U}^+ = \mathbf{U} + \eta\widehat{\mathbf{g}}$, we have

$$\mathbb{E}\left[J_\sigma(\mathbf{U}^+)\right] \ge J_\sigma(\mathbf{U}) + \eta\left(1 - \frac{L_J\eta}{2}\right)\left\|\nabla J_\sigma(\mathbf{U})\right\|^2$$
$$- \frac{L_J\eta^2}{2}\nu^2. \tag{20}$$

Proof. The score-function identity follows from

$$\nabla_{\mathbf{U}} \log \mathcal{N}\left(\mathbf{U}+\boldsymbol{\xi}; \mathbf{U}, \sigma^2 \mathbf{I}\right) = \frac{\boldsymbol{\xi}}{\sigma^2}, \tag{21}$$

which establishes the unbiasedness of $\widehat{\mathbf{g}}$. By the L_J-smoothness of J_σ,

$$\begin{aligned} J_\sigma\left(\mathbf{U}+\eta\widehat{\mathbf{g}}\right) \geq J_\sigma(\mathbf{U}) + \eta\left\langle \nabla J_\sigma(\mathbf{U}), \widehat{\mathbf{g}}\right\rangle \\ - \frac{L_J \eta^2}{2}\left\|\widehat{\mathbf{g}}\right\|^2. \end{aligned} \tag{22}$$

Taking expectations, using Eq. 18, and applying Eq. 19 yield Eq. 20. □

The bound implies an expected increase in the smoothed confidence objective whenever

$$\left(1-\frac{L_J\eta}{2}\right)\left\|\nabla J_\sigma(\mathbf{U})\right\|^2 > \frac{L_J\eta}{2}\nu^2. \tag{23}$$

This proposition concerns only the entropy-based confidence surrogate. Although predictive confidence provides useful intrinsic feedback [22, 30], improving this surrogate does not formally guarantee improved answer correctness, which is evaluated empirically in our experiments.

Hierarchical progression supervision. For the normalized entropies $\mathcal{H}_s, \mathcal{H}_m, \mathcal{H}_l \in [0,1]$, the progression reward satisfies

$$\begin{aligned} \mathcal{R}_{\text{prog}} = \sum_{g\in\{s,m,l\}} (1-\mathcal{H}_g) - [\mathcal{H}_m - \mathcal{H}_s]_+ - [\mathcal{H}_l - \mathcal{H}_m]_+, \\ 0 \leq \mathcal{R}_{\text{prog}} \leq 3, \qquad \mathcal{H}_s \geq \mathcal{H}_m \geq \mathcal{H}_l \implies \mathcal{R}_{\text{prog}} = 3 - \mathcal{H}_s - \mathcal{H}_m - \mathcal{H}_l. \end{aligned} \tag{24}$$

The objective is therefore bounded and numerically stable. The first term applies explicit entropy-reduction pressure to all three groups, while the hinge terms penalize violations of the desired confidence ordering. Once the ordering is satisfied, both hinge terms vanish and every group remains subject to the absolute confidence objective. The reward thus discourages confidence degradation across expanding memory scopes without favoring artificially large inter-group entropy gaps.

Observation 2 (Capacity-level nesting of feasible evidence). Let $\mathfrak{E}_B(\mathcal{C}^g)$ denote the family of feasible evidence subsets containing at most B visual tokens from receptive field $\mathcal{C}^g$. For a fixed latent state $\mathbf{U}$, define the oracle attainable utility as

$$V_g^\star(\mathbf{U}) = \max_{\mathbf{E}\in\mathfrak{E}_B(\mathcal{C}^g)} J_\sigma(\mathbf{U};\mathbf{E}). \tag{25}$$

Because

$$\mathcal{C}^s \subseteq \mathcal{C}^m \subseteq \mathcal{C}^l, \tag{26}$$

the corresponding feasible evidence families are nested:

$$\mathfrak{E}_B(\mathcal{C}^s) \subseteq \mathfrak{E}_B(\mathcal{C}^m) \subseteq \mathfrak{E}_B(\mathcal{C}^l). \tag{27}$$

Consequently,

$$V_s^\star(\mathbf{U}) \leq V_m^\star(\mathbf{U}) \leq V_l^\star(\mathbf{U}). \tag{28}$$

This observation is a capacity-level property of the nested receptive fields. The practical retrieval procedure ranks visual tokens according to their maximum cosine similarity to the current LMTs and selects the Top-B candidates. It does not solve the oracle maximization in Eq. 25. Therefore, Eq. 28 neither guarantees that the utility of the retrieved Top-B evidence increases across memory levels nor that the practical HME procedure recovers an optimal evidence subset. The effectiveness of the retrieval rule is evaluated empirically.

Reward-gated evidence admission. The running threshold used for evidence admission satisfies

$$\widehat{\mathcal{R}}^{(r)} = \max\left\{\widehat{\mathcal{R}}^{(r-1)}, \mathcal{R}_{\text{cand}}^{(r)}\right\} \geq \widehat{\mathcal{R}}^{(r-1)}. \tag{29}$$

This monotonicity applies only to the running evidence-admission threshold. It prevents newly retrieved evidence from persistently accumulating when the candidate score fails to exceed the previously recorded threshold. Since the LMTs continue along the stochastic optimization trajectory

without rollback, Eq. 29 does not imply that the reward of the evolving LMT–evidence state is monotonic across iterations. This asymmetric design makes persistent evidence accumulation conservative without greedily truncating the transient LMT optimization trajectory.

Evidence-conditioned latent updating. Although retrieved evidence cannot retroactively change preceding LMT activations in the same causal forward pass, it conditions the subsequent optimization direction. Specifically, for a twice-differentiable objective,

$$\nabla_{\mathbf{Z}} J_\sigma(\mathbf{Z}, \mathbf{E}) - \nabla_{\mathbf{Z}} J_\sigma(\mathbf{Z}, \mathbf{0}) = \int_0^1 \nabla^2_{\mathbf{ZE}} J_\sigma(\mathbf{Z}, t\mathbf{E})\, \mathbf{E}\, dt. \tag{30}$$

Whenever the mixed derivative is nonzero, retrieved evidence changes the subsequent LMT optimization direction. This identity characterizes how temporarily injected evidence can influence latent updating before being removed; it does not establish that the retrieved evidence is optimal or that the resulting update improves answer correctness. Together, evidence-conditioned updating permits historical information to be internalized into the latent memory, while reward-gated admission prevents non-improving evidence from persistently accumulating in the evidence pool. Whether these mechanisms translate into improved downstream performance is evaluated empirically.

D Additional Experimental Results

D.1 Efficiency Analysis

We evaluate the efficiency of `LatentStream` with Qwen2.5-VL-7B on OVO-Bench [34]. As shown in Table 7, `LatentStream` improves performance from 54.0 to **64.2**, while reducing peak memory from 30.80 GB to **21.97 GB** and TPOT from 6.45 ms to **3.16 ms**. These gains arise because the compact hierarchical memory limits the accumulated visual context and reduces autoregressive decoding overhead. Although latent memory evolution increases TTFT from 7.63 s to 8.41 s, `LatentStream` provides a favorable accuracy–efficiency trade-off.

Table 7: **Efficiency comparison** (§D.1) on OVO-Bench [34] with Qwen2.5-VL-7B [3] as the backbone.

Method	Peak Mem↓	TTFT↓	TPOT↓	Perf.↑
Qwen2.5-VL-7B [3]	30.80GB	7.63s	6.45ms	54.0
+ `LatentStream` **(Ours)**	21.97GB	8.41s	3.16ms	64.2

D.2 Number of Latent Memory Tokens for Each Group

We study the effect of varying the number of Latent Memory Tokens (LMTs) in each memory group from $K = 0$ to $K = 4$. As shown in Fig. 6, introducing one LMT per group improves the overall accuracy from 58.1% to 63.1%, with a particularly substantial gain of 8.8 points on Backward Tracing (BW), demonstrating the importance of latent memory for internalizing historical evidence. Increasing K to 2 further improves overall accuracy to 64.2%. This indicates that a small amount of additional latent capacity is beneficial for capturing complementary historical cues while keeping the optimization space sufficiently compact and stable. Further increasing K yields no additional benefit, suggesting that excessive latent capacity may introduce redundant representations and complicate test-time optimization. We therefore set $K = 2$ by default, which provides the best overall performance with a compact latent memory.

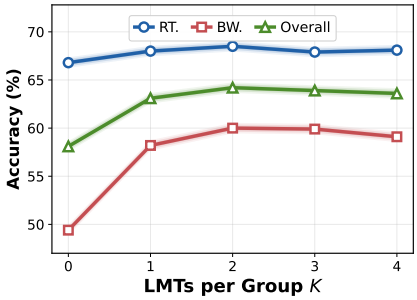


Figure 6: **Effect of the number of Latent Memory Tokens (LMTs) per group** on OVO-Bench [34] (§D.2).

D.3 Effect of Jenks-Based Adaptive Memory Consolidation

We evaluate Jenks-based adaptive consolidation using a controlled variant that replaces only Jenks Natural Breaks [21] with Otsu thresholding [35] while keeping all other components unchanged. As shown in Table 8, Jenks improves Real-Time Visual Perception (RT), Backward Tracing (BW), and overall performance on OVO-Bench [34] from 68.0/58.8/63.4 to 68.5/60.0/64.2, respectively.

Table 8: **Effect of Jenks-based adaptive memory consolidation** (§D.3). All experiments use Qwen2.5-VL-7B [3] under the same frame rate and memory budget. In the controlled Otsu variant, we replace only the Jenks-based partitioning in LatentStream while keeping all other components and hyperparameters unchanged.

Method	Consolidation	OVO-Bench			StreamingBench	VideoMME
		RT.	BW.	Overall	Real-Time	All
FluxMem [54]	Otsu	67.2	46.6	56.9	76.4	65.3
LatentStream w/ Otsu	Otsu	68.0	58.8	63.4	76.7	66.1
LatentStream **(Ours)**	Jenks	**68.5**	**60.0**	**64.2**	**76.9**	**66.6**

Table 9: **Detailed results on OVO-Bench [34]** (§E). OCR, ACR, ATR, STU, FPD, and OJR denote Optical Character Recognition, Action Recognition, Attribute Recognition, Spatial Understanding, Future Prediction, and Object Recognition, respectively. EPM, ASI, and HLD denote Episodic Memory, Action Sequence Identification, and Hallucination Detection, respectively. Best results among all open-source models are in **bold**, while the best results among training-free methods are underlined. † indicates results reproduced by us, and "–" indicates that the corresponding result is not reported.

Method	Size	Frames	Real-Time Visual Perception							Backward Tracing				Overall
			OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	
Proprietary Models														
Gemini 1.5 Pro [40]	–	1 fps	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.4	52.6	62.5	65.9
GPT-4o [20]	–	64	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	62.6
Open-source Offline MLLMs														
LLaVA-Video [69]	7B	64	69.8	59.6	66.4	50.6	72.3	61.4	63.3	51.2	64.2	9.7	41.7	52.5
Qwen2-VL [46]	7B	64	69.1	53.2	63.8	50.6	66.3	60.9	60.7	44.4	**66.9**	34.4	48.6	54.6
InternVL2 [11]	8B	64	68.5	58.7	69.0	44.9	67.3	56.0	60.7	43.1	61.5	27.4	44.0	52.4
LongVU [38]	7B	1 fps	55.7	49.5	59.5	48.3	68.3	63.0	57.4	43.1	66.2	9.1	39.5	48.5
LLaVA-OneVision [26]	8B	32	66.4	57.8	73.3	53.4	71.3	62.0	64.0	54.2	55.4	21.5	43.7	53.9
Open-source Online MLLMs (Training-Based)														
VideoLLM-Online [5] [CVPR 2024]	8B	2 fps	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8	12.2	17.7	19.3
Dispider [37] [CVPR 2025]	7B	1 fps	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4	4.3	36.1	45.3
Flash-VStream [63] [ICCV 2025]	7B	1 fps	25.5	32.1	29.3	33.7	29.7	28.8	29.9	36.4	33.8	5.9	25.4	27.6
ViSpeak [17] [ICCV 2025]	7B	1 fps	75.2	58.7	71.6	51.1	74.3	66.9	66.3	59.9	48.7	64.0	57.5	61.9
TimeChat-Online [60] [ACM MM 2025]	7B	1 fps	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5	9.7	41.7	51.8
StreamForest [61] [NeurIPS 2025]	7B	1 fps	68.5	53.2	71.6	47.8	65.4	60.9	61.2	58.9	64.9	32.3	52.0	56.6
Streamo [53] [CVPR 2026]	7B	1 fps	79.2	57.8	75.0	49.4	64.4	70.1	66.0	54.6	52.0	31.7	46.1	56.1
ThinkStream [32] [ECCV 2026]	3B	1 fps	85.2	64.2	69.8	49.4	69.3	64.1	67.0	53.9	59.5	43.6	52.3	59.7
Open-source Online MLLMs (Training-Free)														
Qwen2.5-VL-3B† [3]	3B	1 fps	77.2	52.3	69.0	41.0	67.3	60.9	61.3	49.8	53.4	26.3	43.2	52.2
+ FluxMem	3B	1 fps	83.2	56.9	67.2	47.8	68.3	63.6	64.5	47.5	54.7	24.2	42.1	53.3
+ `LatentStream` **(Ours)**	3B	1 fps	84.6	57.8	70.7	46.1	71.3	64.1	65.8	51.2	53.4	51.6	52.1	59.0
Qwen2.5-VL-7B† [3]	7B	1 fps	79.2	53.2	67.2	51.7	71.3	57.1	63.3	51.5	58.8	23.7	44.7	54.0
+ FluxMem [54] [CVPR 2026]	7B	1 fps	81.2	59.6	70.7	53.4	75.2	63.0	67.2	50.2	62.8	26.9	46.6	56.9
+ OASIS [28] [CVPR 2026]	7B	–	85.2	72.5	66.4	52.3	67.3	64.7	67.3	51.9	58.8	48.9	52.6	60.0
+ `LatentStream` **(Ours)**	7B	1 fps	85.9	61.5	73.3	46.6	**78.2**	65.2	68.5	50.8	60.8	68.3	60.0	64.2
Qwen3-VL-8B† [2]	8B	1 fps	71.1	65.1	75.9	64.6	75.3	70.7	70.1	52.5	60.1	46.8	52.6	61.4
+ OASIS [28] [CVPR 2026]	8B	–	92.0	80.7	81.0	67.4	67.3	79.9	78.1	62.0	60.1	47.3	57.2	67.7
+ `LatentStream` **(Ours)**	8B	1 fps	**93.3**	**82.6**	**82.8**	**70.8**	70.3	**82.6**	**80.5**	**64.3**	64.9	**72.6**	**66.9**	**73.6**

The larger gain on BW suggests that data-dependent natural breaks better preserve sparse historical evidence during repeated temporal and spatial consolidation. Jenks also consistently improves StreamingBench [29] and VideoMME [16], respectively. Although the controlled Otsu variant already outperforms FluxMem [54], the consistent gains of the full model demonstrate that Jenks provides more effective adaptive partitioning under a fixed memory budget.

E Full Results on Streaming Video Benchmarks

Tables 9 and 10 provide the complete task-wise results on OVO-Bench [34] and StreamingBench [29], respectively. We report results with both Qwen2.5-VL [3] and Qwen3-VL [2] backbones while retaining the same comparison methods, evaluation protocols, and method groupings as in the main paper.

F Case Study

Fig. 7 presents two additional representative cases from a long indoor video on OVO-Bench [34] using Qwen3-VL-8B [2]. `LatentStream` correctly grounds the recipe source to the open cookbook

Table 10: **Detailed results on the Real-Time Visual Understanding subset of StreamingBench [29]** (§E). OP, CR, CS, ATP, EU, TR, PR, SU, ACP, and CT denote Object Perception, Causal Reasoning, Clip Summarization, Attribute Perception, Event Understanding, Text-Rich Understanding, Prospective Reasoning, Spatial Understanding, Action Perception, and Counting, respectively. Best results among all open-source models are in **bold**, while the best results among training-free methods are underlined. † indicates results reproduced by us, and "–" indicates that the corresponding result is not reported.

Method	Size	Frames	Real-Time Visual Understanding										All
			OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	
Open-source Offline MLLMs													
LongVA [67]	7B	128	70.0	63.3	61.2	70.9	62.7	59.5	61.1	53.7	54.7	34.7	60.0
LLaVA-OneVision [26]	7B	32	80.4	74.2	76.0	80.7	72.7	71.7	67.6	65.5	65.7	45.1	71.1
Open-source Online MLLMs (Training-Based)													
VideoLLM-Online [5] [CVPR 2024]	8B	2 fps	39.1	40.1	34.5	31.1	46.0	32.4	31.5	34.2	42.5	27.9	36.0
Dispider [37] [CVPR 2025]	7B	1 fps	74.9	75.5	74.1	73.1	74.4	59.9	76.1	62.9	62.2	45.8	67.6
Flash-VStream [63] [ICCV 2025]	7B	1 fps	25.9	43.6	24.9	23.9	27.3	13.1	18.5	25.2	23.9	48.7	23.2
TimeChat-Online [60] [ACM MM 2025]	7B	1 fps	80.8	79.7	80.8	83.3	74.8	78.8	78.7	64.2	68.8	58.0	75.3
StreamForest [61] [NeurIPS 2025]	7B	1 fps	83.1	82.8	82.7	84.3	77.5	78.2	76.9	69.1	75.6	54.4	77.3
ThinkStream [32] [ECCV 2026]	3B	1 fps	83.7	70.3	76.0	82.7	76.7	78.2	76.9	70.7	74.4	45.2	75.0
Open-source Online MLLMs (Training-Free)													
ReKV [12] [ICLR 2025]	7B	0.5 fps	–	–	–	–	–	–	–	–	–	–	69.1
StreamChat [55] [ICLR 2025]	8B	1 fps	–	–	–	–	–	–	–	–	–	–	64.7
Qwen2.5-VL† [3]	7B	1 fps	78.3	80.5	79.8	82.4	75.5	80.4	74.1	62.6	67.6	51.1	73.9
+ FluxMem [54] [CVPR 2026]	7B	1 fps	80.2	81.1	81.4	85.3	78.0	83.8	80.6	65.9	69.6	52.1	76.4
+ OASIS [28] [CVPR 2026]	7B	–	–	–	–	–	–	–	–	–	–	–	70.6
+ `LatentStream` (Ours)	7B	1 fps	80.2	81.9	85.8	**85.6**	78.6	**86.6**	67.5	**81.5**	67.5	67.9	76.9
Qwen3-VL† [2]	8B	1 fps	76.8	77.2	77.3	80.7	70.4	75.2	80.6	64.2	65.8	49.2	73.2
+ OASIS [28] [CVPR 2026]	8B	–	–	–	–	–	–	–	–	–	–	–	78.2
+ `LatentStream` (Ours)	8B	1 fps	**86.7**	**85.8**	**86.4**	**85.6**	**85.5**	82.9	**87.0**	72.4	**78.4**	**69.2**	**82.0**

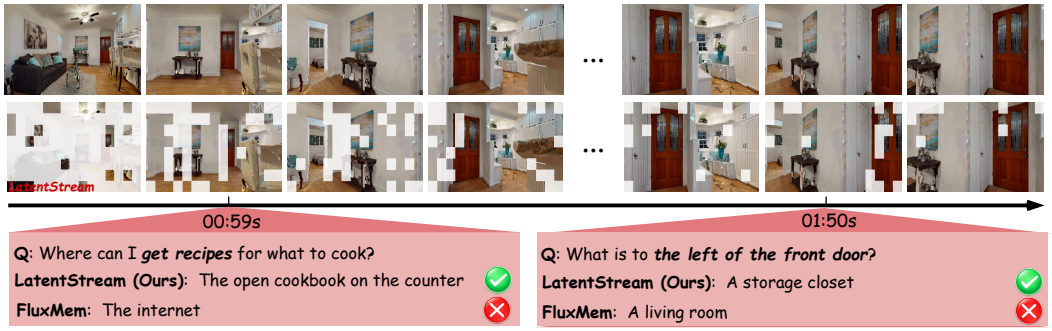


Figure 7: **Qualitative results and token compression visualization** on OVO-Bench [34] (§F).

on the kitchen counter and identifies the storage closet to the left of the front door, whereas FluxMem produces plausible but contextually unsupported answers. The retained visual tokens further show that our hierarchical streaming memory preserves sparse visual and spatial evidence under continual compression, enabling reliable long-range reasoning through progressive latent memory evolution.

G Conclusion

Limitations and Future Work. Despite its effectiveness, `LatentStream` still has several limitations. First, Progressive Confidence-guided Latent Memory Optimization (PMO) adopts a fixed four-round evolution schedule, whose repeated retrieval and latent updating increase the time to first token (TTFT). Future work will explore a single-step optimizer that directly predicts the final Latent Memory Token (LMT) state, together with confidence-adaptive early stopping for queries of different difficulty. Second, Query-agnostic Hierarchical Streaming Memory (HSM) performs irreversible compression before the query arrives and may therefore discard subtle, localized, or static evidence that becomes relevant to a later query. This limitation may be alleviated through uncertainty- and novelty-aware consolidation or recoverable multi-resolution memory under the same bounded budget. Finally,

as PMO uses predictive entropy as a surrogate for reasoning quality, future work may incorporate calibrated uncertainty or lightweight self-verification to reduce incorrect yet overconfident latent updates [22].

Broader Impact. `LatentStream` provides a training-free approach to maintaining bounded visual memory for streaming Video-LLMs. On the positive side, its reduced memory footprint and decoding cost may facilitate continuous video understanding on resource-constrained platforms, with potential applications in assistive smart glasses, long-duration video analysis, embodied agents, and robotic systems [25, 31, 6]. On the negative side, continuously storing and processing visual history may expose sensitive information or enable unauthorized surveillance [20]. Moreover, erroneous memory consolidation or overconfident latent optimization may discard decisive evidence and lead to unreliable responses. Such failures are particularly concerning in safety-critical applications such as autonomous driving and robotic control [6, 31]. Practical deployment should therefore incorporate explicit user consent, appropriate data-retention policies, privacy-preserving memory processing, uncertainty calibration, and human oversight. `LatentStream` should be used as an assistive component rather than an autonomous decision maker in high-stakes scenarios.