STREAMINGMEM-R1: LEARNING MEMORY EVOLUTION FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video understanding requires models to interpret incoming content causally, preserve information that may disappear from view, and respond at the appropriate time. Existing approaches typically retain fragmented historical representations and rely on predefined memory-update rules, limiting their ability to track evolving events and coordinate evidence retention with response decisions. To address these challenges, we propose **StreamingMEM-R1**, a framework that jointly learns memory evolution, evidence routing, and response timing. At each streaming step, the model performs structured memory evolution through two complementary modules: stable memory and dynamic memory. The evolved states parameterize a response policy that estimates whether an answer is warranted. For response steps, a routing variable specifies stable, dynamic, or joint memory as the answer context. We construct 100K supervised fine-tuning examples and 50K reinforcement-learning examples to teach these decisions explicitly. Streaming-aware fine-tuning preserves causal visual and semantic context, while memory-oriented reinforcement learning provides verifiable supervision for memory evolution and trajectory-level correctness. Experiments across four benchmarks demonstrate that StreamingMEM-R1 enables effective memory evolution, accurate long-horizon reasoning, and efficient streaming video understanding.

1 INTRODUCTION

Vision-language models (VLMs) have achieved rapid progress in video understanding, optical character recognition, and temporal reasoning (Team, 2026; Li et al., 2024a; Bai et al., 2025a). Streaming video understanding extends these capabilities to continuously arriving inputs: a model must interpret a video causally, retain evidence that may disappear from view, and answer questions using only information available at the response time (Chen et al., 2024a; Xiong et al., 2025; Yao et al., 2025). A brief on-screen date, for example, may be needed much later, whereas an ongoing action may span several chunks before it can be described correctly. Re-encoding an ever-growing history incurs increasing computation and memory, while retaining only a bounded recent context can discard evidence needed by future queries (Zhang et al., 2025; Di et al., 2025; Yang et al., 2025c; Zhang et al., 2026). Streaming models must therefore determine *what to retain*, *how to update it as events unfold*, and *when the available evidence is sufficient to respond*.

Existing methods follow two broad paradigms. Training-free approaches retrieve frames or KV states and manage cached history through handcrafted compression, eviction, or hierarchical policies (Di et al., 2025; Yang et al., 2025c; Zhang et al., 2026; Chen et al., 2026). Training-based approaches use streaming instructions, reasoning traces, reinforcement learning, or agentic control to improve perception and response behavior (Han et al., 2026; Liu et al., 2026c; Zheng et al., 2026; Wen et al., 2026). A detailed review is provided in Appendix B. As illustrated in Figure 1, both paradigms make important progress but leave two coupled memory challenges:

➠**(I) Memory representations without explicit semantic evolution.** Existing systems typically represent video history using selectively retained visual evidence or compressed latent and linguistic states (Zhang et al., 2025; Di et al., 2025; Ning et al., 2025; Yang et al., 2025c; Kim et al., 2025; Chen et al., 2025; Zhang et al., 2026; Liu et al., 2026c). Although these mechanisms preserve access to past observations, they do not explicitly encode how events unfold, which states remain active, or which facts should persist after the source frames leave the context window. Con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

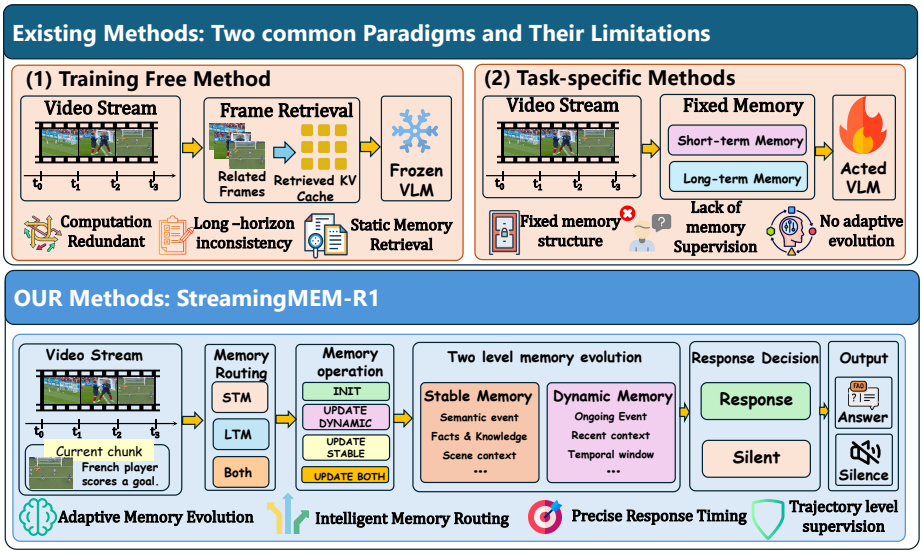


Figure 1: Existing approaches retain video history through retrieval or predefined memory updates. StreamingMEM-R1 learns to evolve complementary memories and select the evidence used for each response.

sequently, downstream reasoning may need to reconstruct temporal relations from partial or compressed evidence. StreamForest and EventMemAgent partially address this issue through persistent or hierarchical event memories, further motivating the explicit organization of temporally distributed evidence (Zeng et al., 2025; Wen et al., 2026).

➠**(II) Memory decisions without unified trajectory supervision.** A streaming model must make several coupled decisions: whether to update or retain memory, whether to remain silent or respond, and whether a response should use recent evidence or persistent historical evidence. Existing training objectives learn response timing, reasoning, or agentic control (Liu et al., 2026c;a; Zheng et al., 2026), while historical retention is commonly governed by FIFO buffering, redundancy-based selection, fixed cache hierarchies, or query-agnostic compression (Han et al., 2026; Zhang et al., 2026; Yang et al., 2025c). These components improve streaming behavior, but are not jointly supervised as a complete trajectory of memory transitions, response decisions, and answer-time evidence routing.

To address these challenges, we propose **StreamingMEM-R1**, which treats memory as a learned, chunk-level decision process. The model maintains a *stable memory* for persistent facts, completed events, and salient attributes, and a *dynamic memory* for ongoing events, recent transitions, and unresolved states. It updates these memories at every chunk, may remain silent when evidence is insufficient, and routes each response to dynamic memory, stable memory, or both. The framework comprises three coordinated components: ❶ **Memory-evolution trajectories.** We refine ThinkStream annotations with Qwen3.5-VL-35B-A3B (Team, 2026) to construct **100K SFT and 50K RL** samples that supervise memory transitions, response timing, and answer-time routing to finetune Qwen2.5-VL-7B (Bai et al., 2025b). ❷ **Streaming-aware SFT.** Logical sequence reordering, a dual-level KV cache, and causal attention masking preserve a bounded window of recent visual evidence together with persistent semantic history. ❸ **Memory-oriented RLVR.** Self-conditioned rollouts carry the model's generated memory across chunks and provide verifiable rewards for memory operations, routing, response timing, and answer correctness.

Experiments on OVOBench, StreamingBench, and RVS show that StreamingMEM-R1 improves streaming video understanding across current perception and historical reasoning. It exceeds the strongest reported baselines by 5.63% on OVOBench, 3.01% on StreamingBench, and 4.6 and 4.2 % on RVS-EGO and RVS-MOVIE, respectively. Our streaming inference system also reduces per-chunk latency by 83.6% and peak GPU memory by 61.0% relative to full-history recomputation.

Our contributions are summarized as follows:

- ☞ We formulate streaming memory as a learned evolution-and-routing process that coordinates complementary stable and dynamic semantic states.
- ☞ We develop causal trajectory supervision, streaming-aware SFT, and memory-oriented RLVR to jointly learn memory updates, evidence selection, response timing, and grounded answering.
- ☞ Extensive experiments on four benchmarks demonstrate that StreamingMEM-R1 significantly outperforms existing methods across diverse streaming video understanding tasks.

2 METHODS

2.1 OVERVIEW

StreamingMEM-R1 consists of three stages: **(I) memory-trajectory construction**, which supervises evolving short- and long-term memories, response decisions, and memory routing; **(II) streaming-aware SFT**, which combines sequence reordering, dual-level KV caching, and causal masking to preserve visual continuity; and **(III) memory-oriented RLVR**, which jointly optimizes memory operations, routing, response timing, and answer accuracy. The overall framework is illustrated in Figure 2.

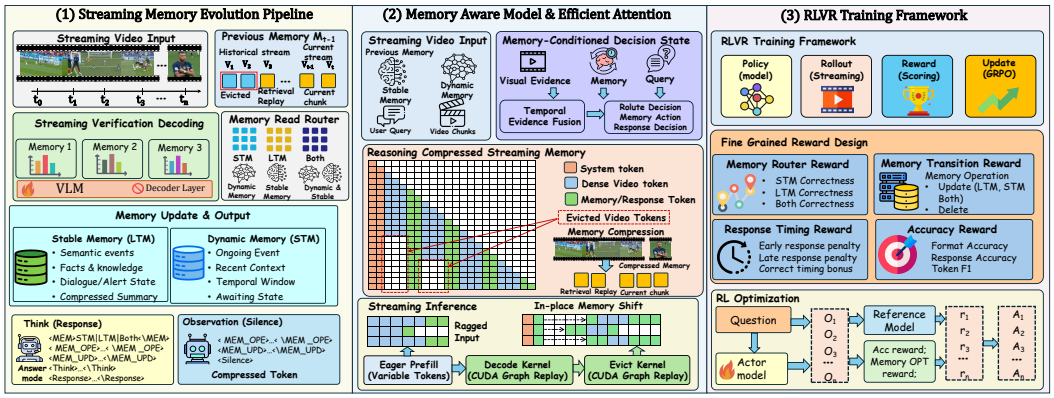


Figure 2: StreamingMEM-R1 overview: (I) evolving stable and dynamic memories with silence/response control; (II) memory-aware attention and compression for efficient streaming inference; (III) RLVR with rewards for memory routing, transitions, response timing, and accuracy.

2.2 MEMORY-TRAJECTORY CONSTRUCTION

We use **Qwen3.5-35B-A3B** Team (2026) as the teacher model to convert ThinkStream's chunk-level annotations into structured trajectories. The construction process has three steps: update memory for each video chunk, decide whether to respond, and retain only trajectories with verified answers. The resulting trajectories supervise both streaming-aware SFT and memory-oriented RLVR. Dataset sources and statistics are reported in Appendix D.1; source formats, causal prompts, annotation schemas, and validation rules are provided in Appendix A; and concrete serialized training examples are shown in Appendix G.

Observations and memory states. For a video V, query q, and ordered chunks $(v_t)_{t=1}^{T}$, T is the number of chunks and t is the current chunk index. We derive an observation o_t from annotations aligned with v_t, using only information available up to step t. After processing v_t, the memory state:

$$M_t = (M_t^S, M_t^D), \qquad M_0 = (\emptyset, \emptyset), \tag{1}$$

where M_t^S stores persistent events and facts as stable (long-term) memory, and M_t^D stores the current event or interaction as dynamic (short-term) memory. M_0 is the initial state, and $\emptyset$ denotes an empty memory. Thus, o_t provides new evidence, while M_{t-1} summarizes the preceding stream.

Memory updates. At each step, the teacher updates memory as

$$(a_t^{\text{mem}}, M_t) = F_\phi(o_t, M_{t-1}, q), \tag{2}$$

where F_ϕ is the teacher with parameters ϕ, a_t^{mem} is the memory operation, and M_t is the updated state. The main operations are `INIT`, which initializes both memories; `UPDATE_DYNAMIC`, which updates only M_t^D; and `UPDATE_BOTH`, which updates both M_t^S and M_t^D. Memory is updated at every step, independently of whether the model responds.

Response control and routing. The teacher next chooses $d_t \in \{\texttt{silent}, \texttt{response}\}$, where d_t indicates whether the available evidence is sufficient to answer. Define the shared memory output as $p_t = (a_t^{\text{mem}}, M_t)$ and the explicit silence token as $s_t = \texttt{<silent>}$. The complete step output is

$$z_t = \begin{cases} (p_t, s_t), & d_t = \texttt{silent}, \\ (p_t, y_t), & d_t = \texttt{response}, \end{cases} \tag{3}$$

where z_t is the serialized output at step t. A silent step contains only the memory update and s_t; a response step additionally contains

$$y_t = (r_t, c_t, b_t), \qquad (c_t, b_t) = G_\phi(o_t, M_t, q, r_t), \tag{4}$$

where y_t is the response payload; $r_t \in \{\texttt{STM}, \texttt{LTM}, \texttt{BOTH}\}$ routes the query to M_t^D, M_t^S, or both; c_t is the reasoning trace; b_t is the final answer; and G_ϕ is the teacher's response-generation procedure. Therefore, a_t^{mem} controls which memory is updated at every step, whereas r_t controls which memory is consulted only at response steps. The exact teacher schema and target serialization are given in Appendices A.3 and A.6.

Trajectory assembly and rejection sampling. For query q, we assemble the ordered trajectory

$$\mathcal{T} = \big((o_t, d_t, z_t)\big)_{t=1}^{T}, \tag{5}$$

where $\mathcal{T}$ is the complete trajectory and each tuple records the observation o_t, response decision d_t, and output z_t at step t. We verify multiple-choice answers by exact option matching and free-form answers by a separate semantic-equivalence judgment from the teacher.

Let $\mathcal{R}^*$ denote the nonempty set of response steps required by the source annotations and b_t^* the ground-truth answer at step t. We retain a candidate trajectory only if

$$\forall t \in \mathcal{R}^* : \quad d_t = \texttt{response} \ \wedge \ \text{Verify}(q, b_t, b_t^*) = 1, \tag{6}$$

where $\forall$ means "for every," Verify returns 1 for semantically equivalent answers (or matching multiple-choice options) and 0 otherwise, and $\wedge$ denotes logical conjunction. Failed candidates are rejected and regenerated. This prevents silent outputs from bypassing required answers, although answer correctness does not directly verify every intermediate memory update or response timing. Leakage-free answer supervision and independent video-evidence verification are detailed in Appendices A.4 and A.5.

2.3 Streaming-aware supervised fine-tuning

We train a student policy π_θ, with parameters θ, to predict the structured output z_t from video chunk v_t, query q, and the preceding stream context. The caption-derived observation o_t is used only for trajectory construction; the student learns from visual inputs. Our SFT design combines *reasoning-guided visual compression*, *visual-first cache reordering*, and *modality-aware causal masking* to retain historical semantics while preserving continuous, temporally aligned perception.

Reasoning-guided visual compression and cache reordering. We maintain a visual cache $\mathcal{K}_t^{\text{vis}}$ for recent video tokens and a semantic cache $\mathcal{K}_t^{\text{sem}}$ for generated outputs (Yang et al., 2026; Liu et al., 2026c). Before old visual tokens are removed, their salient information is summarized in $M_t = (M_t^S, M_t^D)$ and, at response steps, c_t. Thus, dense visual history is compressed into persistent semantic states. The visual and semantic caches are storage partitions and should not be confused with the stable and dynamic components of M_t, both of which belong to the semantic cache.

For a window of W chunks, recent visual states and historical semantic states are organized as

$$\begin{aligned} \mathcal{K}_t^{\text{vis}} &= \text{Tail}_W\big([\mathcal{K}_{t-1}^{\text{vis}}; \text{KV}(v_t)]\big), \\ \widetilde{\mathcal{K}}_t &= [\mathcal{K}_t^{\text{vis}}; \mathcal{K}_{t-1}^{\text{sem}}], \\ \mathcal{K}_t^{\text{sem}} &= [\mathcal{K}_{t-1}^{\text{sem}}; \text{KV}(z_t)]. \end{aligned} \tag{7}$$

Here, W is the visual-window size; $\mathrm{KV}(\cdot)$ computes key–value states; $[\cdot;\cdot]$ denotes ordered concatenation; and Tail_W retains the latest W visual chunks. The first line appends v_t and evicts older visual states, the second forms the decoding cache $\widetilde{\mathcal{K}}_t$ by placing recent visual states before previous semantic states, and the third appends the current output z_t to the semantic cache. Both caches are initially empty, excluding fixed prompt states. Logical positions are preserved after reordering, and historical visual replay can recover details omitted by semantic compression.

Modality-aware causal alignment. Cache reordering changes physical storage but not causal order. Motivated by causal video attention and specialized long-context attention mechanisms (Fei et al., 2024; Zheng et al., 2024; Basile et al., 2025; Xiao et al., 2024), we explicitly control visibility across modalities and time. Let i denote a query-token position and j a key-token position. For either position $l \in \{i, j\}$, $\tau(l)$ gives its chunk index and $\mu(l) \in \{\mathrm{vis}, \mathrm{sem}\}$ gives its visual or semantic modality. We define the additive attention mask as

$$A_{ij} = \begin{cases} 0, & j \le i,\ \mu(j) = \mathrm{vis},\ B_{ij} = 1, \\ 0, & j \le i,\ \mu(i) = \mu(j) = \mathrm{sem}, \\ -\infty, & \text{otherwise}, \end{cases} \tag{8}$$

where A_{ij} controls whether token i may attend to token j: 0 permits attention and $-\infty$ blocks it. The condition $j \le i$ enforces causal order. Visual keys additionally require the availability indicator $B_{ij} = 1$, defined as

$$B_{ij} = \begin{cases} 1, & 0 \le \tau(i) - \tau(j) < W \text{ or } j \in \mathcal{E}_i, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathcal{E}_i$ is the set of historical visual-token indices replayed for token i. Hence, a visual key is available if it lies within the latest W chunks or is explicitly replayed. $\mathcal{E}_i$ is nonempty only for answer tokens after an `LTM` or `BOTH` route. Semantic tokens may attend to earlier semantic tokens, whereas visual tokens cannot access semantic history. This mask matches visual eviction during inference and prevents future information leakage.

Response-conditional trajectory supervision. Both branches supervise the shared memory output $p_t = (a_t^{\mathrm{mem}}, M_t)$. Silent steps then terminate with `<silent>`, while response steps additionally supervise $y_t = (r_t, c_t, b_t)$. Consequently, routing and historical replay are activated only when the model responds.

For each trajectory, we minimize assistant-token negative log-likelihood:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{t=1}^{T} \sum_{k=1}^{|z_t|} \log \pi_\theta(z_{t,k} \mid q, z_{t,<k}, \mathcal{C}_{t,k}) \,. \tag{10}$$

Here, $\mathcal{L}_{\mathrm{SFT}}$ is the summed negative log-likelihood, so minimizing it increases the probability of target tokens. T is the number of chunks; k indexes output tokens; $|z_t|$ is the length of z_t; $z_{t,k}$ is its kth token; and $z_{t,<k}$ is its teacher-forced prefix. $\mathcal{C}_{t,k}$ denotes the visual and semantic context allowed by Equations 8–9. The conditional probability $\pi_\theta(z_{t,k} \mid \cdot)$ is assigned by the student policy, and $\log$ is the natural logarithm. Only output tokens contribute to the loss; the visual backbone is frozen, while the visual merger and language model are optimized. Training and inference use the same visibility rules. Detailed SFT hyperparameters, sampling settings, and context budgets are provided in Appendix D.2.

2.4 Memory-oriented RLVR

SFT teaches the model to imitate reference trajectories, but streaming inference is self-conditioned: every prediction depends on memories generated at preceding steps. To reduce this train–test discrepancy, we further optimize the SFT model with memory-oriented reinforcement learning with verifiable rewards (RLVR). Following the streaming group-relative optimization paradigm (Shao et al., 2024; Liu et al., 2026c), the policy explores complete trajectories and receives rule-based feedback on memory management and response behavior.

Self-conditioned streaming rollout. For each video–query pair (V, q), the rollout policy samples a group of trajectories $\{\mathcal{T}^{(g)}\}$. Within each trajectory, the generated memory is recursively carried to

the next video chunk:

$$z_t^{(g)} \sim \pi_{\text{old}}\Big(\cdot \mid v_t, q, \widehat{M}_{t-1}^{(g)}\Big), \qquad \widehat{M}_t^{(g)} = \text{ParseMem}\Big(z_t^{(g)}\Big), \qquad \widehat{M}_0^{(g)} = (\emptyset, \emptyset). \tag{11}$$

Thus, the rollout reproduces the actual streaming process rather than conditioning on ground-truth memory at every step. An incorrect update can influence subsequent routing, reasoning, and answers, allowing trajectory-level feedback to capture error propagation. After scoring all trajectories for the same input, GRPO increases the likelihood of trajectories whose rewards are higher relative to the group while regularizing the policy toward the SFT model.

Verifiable reward design. Each sampled trajectory is compared with its reference trajectory using four deterministic rewards. The *routing reward* $\mathcal{R}_{\text{route}}$ checks whether a response uses the required source in $\{\texttt{STM}, \texttt{LTM}, \texttt{BOTH}\}$ and rejects invalid routing fields at silent steps. The *operation reward* $\mathcal{R}_{\text{op}}$ evaluates the predicted memory operation at every chunk and verifies that the serialized stable and dynamic memories are consistent with that transition. The *timing reward* $\mathcal{R}_{\text{time}}$ evaluates whether the model correctly remains silent or responds as the stream unfolds. The *answer reward* $\mathcal{R}_{\text{ans}}$ measures correctness only at required response steps, using exact option agreement for multiple-choice questions and normalized token matching for free-form answers. Malformed outputs and missing required answers receive zero on the corresponding component.

The overall trajectory reward is the direct sum of these complementary signals:

$$\mathcal{R}^{(g)} = \mathcal{R}_{\text{route}}^{(g)} + \mathcal{R}_{\text{op}}^{(g)} + \mathcal{R}_{\text{time}}^{(g)} + \mathcal{R}_{\text{ans}}^{(g)}. \tag{12}$$

This formulation introduces no learned reward model. It jointly encourages the policy to update the correct memory, select the appropriate evidence source, respond at the correct moment, and produce a grounded answer. Detailed RLVR optimization and rollout settings are provided in Appendix D.3.

3 Experiments

We evaluate StreamingMEM-R1 on three streaming video understanding benchmark families—OVOBench, StreamingBench, and RVS (Li et al., 2025b; Lin et al., 2026; Zhang et al., 2026)—to assess its overall effectiveness. Dataset definitions and baseline groups are detailed in Appendix C, while the complete training and compute configurations are reported in Appendix D. Beyond the main comparisons, we conduct ablation studies, duration-stratified and chunk-size analyses, reward-curve visualization, and runtime profiling to examine component contributions, long-horizon robustness, training dynamics, and streaming efficiency.

3.1 Main Results

As shown in Table 1, StreamingMEM-R1 achieves the highest overall score of 65.83, exceeding the strongest baseline, WAT, by 5.63 percentage points. It also achieves the best perception and reasoning averages of 72.35 and 59.31, improving over the respective strongest baselines by 5.32 and 7.01 points. The largest subtask gains occur in STU and ACR, with improvements of 13.43 and 8.99 points over the best competing scores, respectively. These results demonstrate improvements across both perception and reasoning rather than gains confined to a single evaluation category.

Table 2 shows that StreamingMEM-R1 achieves the highest model average of 80.71, outperforming WAT by 3.01 points and the strongest open-source offline model, Qwen2.5-VL-32B, by 6.44 points. Our method leads on six of the ten subtasks, with the largest margin in SU: 85.31 versus 73.17 for the strongest baseline. Together, the two benchmarks demonstrate consistent gains in aggregate streaming performance, although improvements are not uniform across subtasks; for example, EU and TR on StreamingBench remain below the best competing scores.

3.2 Evaluation on RVS and Reward Dynamics

RVS evaluation. Table 3 shows that StreamingMEM-R1 ranks first on both RVS subsets, outperforming the strongest baseline by 4.6 points on RVS-EGO and 4.2 points on RVS-MOVIE. This result extends the main-benchmark findings to retrospective visual search, where answers depend

Table 1: Results on OVOBench. Bold denotes the best score in each column. For streaming methods, subscripts indicate score differences from the best base-model result in each column (upper block): ↑ for gains and ↓ for drops.

Model	Perception							Reasoning				Overall
	OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	
LLaVA-Video-7B	69.13	58.72	68.83	49.44	**74.26**	59.78	63.52	56.23	57.43	7.53	40.40	51.96
Qwen2-VL-7B	60.40	50.46	56.03	47.19	66.34	55.43	55.98	47.81	35.48	**56.08**	46.46	51.22
LongVU-7B	53.69	53.21	62.93	47.75	68.32	59.78	57.61	40.74	59.46	4.84	35.01	46.31
Qwen2.5-VL-3B	76.51	44.03	67.24	42.13	68.31	61.96	60.03	50.50	53.38	22.04	41.98	51.00
Qwen2.5-VL-7B	67.79	55.05	67.24	42.13	66.34	60.87	59.90	51.52	58.78	23.66	44.65	52.28
Qwen2.5-VL-32B	77.18	58.72	68.10	50.56	**74.26**	57.61	64.40	58.59	62.84	29.57	50.33	57.37
Flash-VStream-7B	$24.16_{\downarrow 53.02}$	$29.36_{\downarrow 29.36}$	$28.45_{\downarrow 40.38}$	$33.71_{\downarrow 16.85}$	$25.74_{\downarrow 48.52}$	$28.80_{\downarrow 33.16}$	$28.37_{\downarrow 36.03}$	$39.06_{\downarrow 19.53}$	$37.16_{\downarrow 25.68}$	$5.91_{\downarrow 50.17}$	$27.38_{\downarrow 22.95}$	$27.88_{\downarrow 29.49}$
VideoLLM-online-8B	$8.05_{\downarrow 69.13}$	$23.85_{\downarrow 34.87}$	$12.07_{\downarrow 56.76}$	$14.04_{\downarrow 36.52}$	$45.54_{\downarrow 28.72}$	$21.20_{\downarrow 40.76}$	$20.79_{\downarrow 43.61}$	$22.22_{\downarrow 36.37}$	$18.80_{\downarrow 44.04}$	$12.18_{\downarrow 43.90}$	$17.73_{\downarrow 32.60}$	$19.26_{\downarrow 38.11}$
Dispider-7B	$57.72_{\downarrow 19.46}$	$49.54_{\downarrow 9.18}$	$62.07_{\downarrow 6.76}$	$44.94_{\downarrow 5.62}$	$61.39_{\downarrow 12.87}$	$51.63_{\downarrow 10.33}$	$54.55_{\downarrow 9.85}$	$48.48_{\downarrow 10.11}$	$55.41_{\downarrow 7.43}$	$4.30_{\downarrow 51.78}$	$36.06_{\downarrow 14.27}$	$45.31_{\downarrow 12.06}$
StreamForest-7B	$68.46_{\downarrow 8.72}$	$53.21_{\downarrow 5.51}$	$71.55_{\uparrow 2.72}$	$47.75_{\downarrow 2.81}$	$65.35_{\downarrow 8.91}$	$60.87_{\downarrow 1.09}$	$61.20_{\downarrow 3.20}$	**58.92**$_{\uparrow 0.33}$	$64.86_{\uparrow 2.02}$	$32.26_{\downarrow 23.82}$	$52.02_{\uparrow 1.69}$	$56.61_{\downarrow 0.76}$
Streamo-3B	$78.52_{\uparrow 1.34}$	$52.29_{\downarrow 6.43}$	$67.24_{\downarrow 1.59}$	$44.38_{\downarrow 6.18}$	$55.45_{\downarrow 18.81}$	**71.20**$_{\uparrow 9.24}$	$61.51_{\downarrow 2.89}$	$51.18_{\downarrow 7.41}$	$57.43_{\downarrow 5.41}$	$16.67_{\downarrow 39.41}$	$41.76_{\downarrow 8.57}$	$51.64_{\downarrow 5.73}$
WAT	$81.20_{\uparrow 4.02}$	$48.30_{\downarrow 10.42}$	$70.70_{\uparrow 1.87}$	$48.30_{\downarrow 2.26}$	$74.00_{\downarrow 0.26}$	$65.20_{\uparrow 3.24}$	$64.70_{\uparrow 0.30}$	$58.90_{\uparrow 0.31}$	$64.90_{\uparrow 2.06}$	$11.80_{\downarrow 44.28}$	$45.20_{\downarrow 5.13}$	$60.20_{\uparrow 2.83}$
ThinkStream	**85.23**$_{\uparrow 8.05}$	$64.22_{\uparrow 5.50}$	$69.82_{\uparrow 0.99}$	$49.43_{\downarrow 1.13}$	$69.31_{\downarrow 4.95}$	$64.13_{\uparrow 2.17}$	$67.03_{\uparrow 2.63}$	$53.87_{\downarrow 4.72}$	$59.46_{\downarrow 3.38}$	$43.55_{\downarrow 12.53}$	$52.30_{\uparrow 1.97}$	$59.66_{\uparrow 2.29}$
StreamingMEM-R1	$82.82_{\uparrow 5.64}$	**73.21**$_{\uparrow 14.49}$	**72.93**$_{\uparrow 4.10}$	**63.99**$_{\uparrow 13.43}$	$72.47_{\downarrow 1.79}$	$68.70_{\uparrow 6.74}$	**72.35**$_{\uparrow 7.95}$	$56.54_{\downarrow 2.05}$	**65.49**$_{\uparrow 2.65}$	$55.91_{\downarrow 0.17}$	**59.31**$_{\uparrow 8.98}$	**65.83**$_{\uparrow 8.46}$

Table 2: Results on StreamingBench. Bold denotes the best model score in each column, excluding human performance. Online-method subscripts show score differences from the best open-source offline model in each column: ↑ for gains and ↓ for drops.

Model	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Avg.
Human	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46
Proprietary MLLMs											
Gemini 1.5 pro	79.02	80.47	83.54	79.67	**80.00**	**84.74**	77.78	64.23	71.95	48.70	75.69
GPT-4o	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28
Claude 3.5 Sonnet	73.33	80.47	84.09	82.02	75.39	79.53	61.11	61.79	69.32	43.09	72.44
Open-source offline MLLMs											
VILA-1.5-8B	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32
LongVA-7B	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96
LLaVA-NeXT-Video-32B	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96
Qwen2.5-VL-3B	67.21	68.75	75.39	79.17	72.96	72.27	71.30	61.38	71.59	26.06	67.96
Qwen2.5-VL-7B	77.93	76.56	78.55	80.86	76.73	76.95	80.56	65.45	65.72	52.85	73.31
Qwen2.5-VL-32B	76.29	79.69	78.55	83.50	76.10	79.44	80.56	61.38	68.27	59.07	74.27
Open-source online MLLMs											
Flash-VStream-7B	$25.89_{\downarrow 52.31}$	$43.57_{\downarrow 36.12}$	$24.91_{\downarrow 53.64}$	$23.87_{\downarrow 59.63}$	$27.33_{\downarrow 49.40}$	$13.08_{\downarrow 66.36}$	$18.52_{\downarrow 62.04}$	$25.20_{\downarrow 40.25}$	$23.87_{\downarrow 47.72}$	$48.70_{\downarrow 10.37}$	$23.23_{\downarrow 51.04}$
VideoLLM-online-8B	$39.07_{\downarrow 39.13}$	$40.06_{\downarrow 39.63}$	$34.49_{\downarrow 44.06}$	$31.05_{\downarrow 52.45}$	$45.96_{\downarrow 30.77}$	$32.40_{\downarrow 47.04}$	$31.48_{\downarrow 49.08}$	$34.16_{\downarrow 31.29}$	$42.49_{\downarrow 29.10}$	$27.89_{\downarrow 31.18}$	$35.99_{\downarrow 38.28}$
Dispider-7B	$74.92_{\downarrow 3.28}$	$75.53_{\downarrow 4.16}$	$74.10_{\downarrow 4.45}$	$73.08_{\downarrow 10.42}$	$74.44_{\downarrow 2.29}$	$59.92_{\downarrow 19.52}$	$76.14_{\downarrow 4.42}$	$62.91_{\downarrow 2.54}$	$62.16_{\downarrow 9.43}$	$45.80_{\downarrow 13.27}$	$67.63_{\downarrow 6.64}$
TimeChatOnline-7B	$80.20_{\uparrow 2.00}$	$82.00_{\uparrow 2.31}$	$79.50_{\uparrow 0.95}$	$83.30_{\downarrow 0.20}$	$76.10_{\downarrow 0.63}$	$78.50_{\downarrow 0.94}$	$78.70_{\downarrow 1.86}$	$64.60_{\downarrow 0.85}$	$69.60_{\downarrow 1.99}$	$58.00_{\downarrow 1.07}$	$75.40_{\uparrow 1.13}$
Streamforest-7B	$83.10_{\uparrow 4.90}$	**82.80**$_{\uparrow 3.11}$	$82.70_{\uparrow 4.15}$	$84.30_{\uparrow 0.80}$	$77.50_{\uparrow 0.77}$	$78.20_{\downarrow 1.24}$	$76.90_{\downarrow 3.66}$	$69.10_{\uparrow 3.65}$	**75.60**$_{\uparrow 4.01}$	$54.40_{\downarrow 4.67}$	$75.40_{\uparrow 1.13}$
WAT	$82.93_{\uparrow 4.73}$	$78.91_{\downarrow 0.78}$	$85.49_{\uparrow 6.94}$	$84.24_{\uparrow 0.74}$	$76.58_{\downarrow 0.15}$	$83.80_{\uparrow 4.36}$	$83.33_{\uparrow 2.77}$	$73.17_{\uparrow 7.72}$	$72.44_{\uparrow 0.85}$	$45.74_{\downarrow 13.33}$	$77.70_{\uparrow 3.43}$
ThinkStream	$83.74_{\uparrow 5.54}$	$70.31_{\downarrow 9.38}$	$76.03_{\downarrow 2.52}$	$82.69_{\downarrow 0.81}$	$76.73_{0.00}$	$78.19_{\downarrow 1.25}$	$76.85_{\downarrow 3.71}$	$70.73_{\downarrow 5.28}$	$74.43_{\downarrow 2.84}$	$45.21_{\downarrow 13.86}$	$75.00_{\uparrow 0.73}$
StreamingMEM-R1	**86.73**$_{\uparrow 8.53}$	$79.17_{\downarrow 0.52}$	**88.03**$_{\uparrow 9.48}$	**87.24**$_{\uparrow 3.74}$	$74.37_{\downarrow 2.36}$	$81.22_{\uparrow 1.78}$	**86.23**$_{\uparrow 5.67}$	**85.31**$_{\uparrow 19.86}$	$75.14_{\uparrow 3.55}$	**63.63**$_{\uparrow 4.56}$	**80.71**$_{\uparrow 6.44}$

heavily on preserving and recovering distant evidence. The remaining gap between the two subsets also suggests that complex movie narratives continue to pose a greater long-horizon memory challenge.

Reward dynamics. Figure 3 shows that all reward components improve over training despite local fluctuations. Routing and timing stabilize earlier, whereas answer correctness exhibits the largest overall improvement, suggesting that the policy first learns the output protocol and then refines grounded answering. The joint upward trend supports the role of memory-oriented RLVR in coordinating memory management and response behavior; benchmark results provide the corresponding evidence of downstream generalization.

Table 3: Comparison with streaming video understanding methods on RVS-EGO and RVS-MOVIE. The best results are highlighted in bold.

Method	RVS-EGO ↑	RVS-MOVIE ↑
LLaVA-OV-7B	56.2	43.0
ReKV	55.8	50.8
FlashVStream	57.0	53.1
InfiniPot-V	57.9	51.4
StreamMem	57.6	52.7
StreamingTOM	58.3	53.2
HERMES	60.3	54.4
StreamingMEM-R1	**64.9**	**58.6**

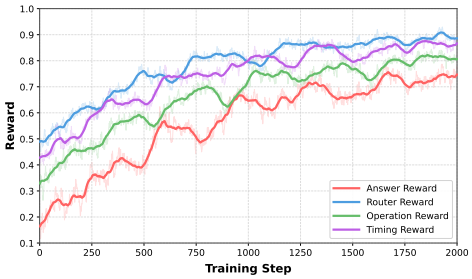


Figure 3: Memory-RLVR reward curves during training.

3.3 Ablation Study

Figure 4 evaluates four components of StreamingMEM-R1. Removing any component lowers accuracy on both benchmarks, supporting the contribution of each to the complete framework. Full task-level results are reported in Appendix E.1.

Memory routing and evolution. Removing either the memory router or the explicit memory operations consistently degrades both benchmarks, with the router ablation causing a drop of up to 5.21 points. These results validate two distinct decisions in our framework: routing determines *which evidence to consult*, whereas memory operations determine *how retained information evolves*.

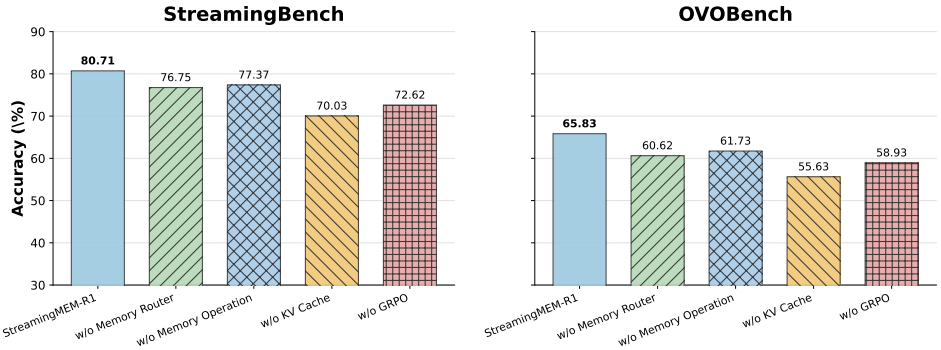


Figure 4: Ablation study on StreamingBench and OVOBench.

Dual-level streaming KV modeling. Removing the complete dual-level streaming KV mechanism, which jointly maintains recent visual KV states in $\mathcal{K}^{\text{vis}}$ and accumulated textual-semantic KV states in $\mathcal{K}^{\text{sem}}$—causes the largest degradation, exceeding 10 points on both benchmarks. Without this mechanism, the model loses direct access to fine-grained recent observations as well as the compressed memory of earlier chunks, weakening both cross-chunk state tracking and long-range evidence integration. This result identifies coordinated visual–semantic KV modeling as a central source of the model's video-understanding capability, rather than merely an inference-side cache optimization.

Trajectory-level optimization. Removing GRPO causes a substantial drop of up to 8.09 points, showing that supervised initialization alone is insufficient. Optimizing self-conditioned trajectories is therefore important for correcting errors that propagate through memory updates, routing, and later responses. Overall, the ablations support the complementary roles of context preservation, explicit memory control, and trajectory-level policy optimization.

3.4 Chunk-size Sensitivity and Video-duration Robustness

Chunk-size sensitivity. Figure 5 compares several fixed chunk lengths with adaptive chunking. Performance generally improves as fixed chunks provide more local context, while adaptive chunking adds roughly two points over the strongest fixed setting. This suggests that content-dependent

Table 4: Efficiency analysis of the proposed streaming inference components. Latency is measured per video chunk, and peak memory denotes the maximum GPU memory consumption.

Variant	Latency/Chunk ↓	Peak Memory ↓	Speedup ↑	Accuracy ↑
Full Recompute	8.74 s	70.8 GB	1.00×	80.23
+ Streaming Window	2.83 s	35.4 GB	3.09×	80.49
+ In-place KV Shift	2.09 s	32.8 GB	4.18×	80.64
+ CUDA Graph Decode	1.66 s	29.1 GB	5.27×	80.68
Full StreamingMEM-R1	**1.43 s**	**27.6 GB**	**6.11×**	**80.71**

boundaries better balance local completeness and timely memory updates than a single global chunk size; the associated latency trade-off is not measured here.

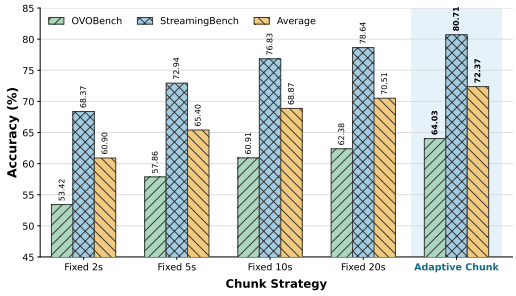


Figure 5: Chunk-size sensitivity.

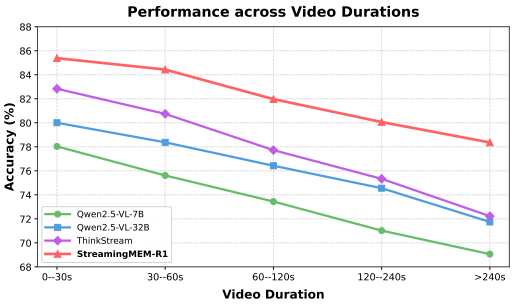


Figure 6: Performance across video durations.

Video-duration robustness. Figure 6 compares performance across five video-duration bins, from 0–30 seconds to over 240 seconds. Although accuracy decreases with duration for all methods, StreamingMEM-R1 consistently outperforms Qwen2.5-VL-7B, Qwen2.5-VL-32B, and ThinkStream, with a smaller decline between the shortest and longest bins. Its lead over the strongest competing model also widens as duration increases. This pattern is consistent with the intended role of evolving memory in preserving useful historical evidence, while the remaining decline shows that longer video streams remain challenging. Qualitative examples of cross-chunk reasoning and delayed historical retrieval are provided in Appendix F.

3.5 Efficiency Analysis

Table 4 shows that StreamingMEM-R1 improves efficiency without sacrificing predictive quality. Relative to full-history recomputation, it reduces per-chunk latency by 83.6% and peak GPU memory by 61.0%, while also achieving the best accuracy among the evaluated variants. The result demonstrates that bounded streaming context is beneficial not only computationally but also for maintaining relevant evidence.

The incremental comparison shows that the streaming window contributes the largest initial saving, while in-place KV shifting and CUDA Graph decoding provide additional complementary gains. Accuracy remains effectively stable across these optimized variants, indicating that the speed and memory improvements do not come from weakening the model's predictions. Additional efficiency results are reported in Appendix E.2.

4 Conclusion

We introduced StreamingMEM-R1, a unified framework that learns stable and dynamic memory evolution, response timing, and answer-time evidence routing through causal trajectories, streaming-aware SFT, and memory-oriented RLVR. Experiments across four streaming benchmarks demonstrate strong long-horizon understanding and historical evidence retrieval while substantially reducing inference latency and memory consumption.

AI USE STATEMENT

In this work, generative AI models were used to assist with language editing, proofreading, and presentation refinement. The authors reviewed and revised all AI-assisted text and remain fully responsible for the accuracy, originality, and scientific integrity of the final manuscript. The multimodal teacher model used for constructing training trajectories is described as part of the proposed data-generation pipeline rather than as an authoring tool.

ETHICS STATEMENT

StreamingMEM-R1 is designed for causal video understanding over continuously arriving visual content. Potential applications include assistive monitoring, interactive video agents, and long-horizon video analysis. Such applications may involve sensitive information, including faces, identities, activities, locations, and private conversations. Models trained on video data may also inherit biases from their backbone models, training sources, and annotation procedures, and may produce incorrect, incomplete, or overconfident memories and responses. In particular, preserving historical evidence can increase the risk of unintended retention or disclosure of private information. The model's outputs should therefore not be treated as verified evidence or used as the sole basis for high-impact decisions. Responsible deployment should include consent and privacy protections, compliance with the licenses and usage terms of source videos and pretrained models, secure handling of stored memories, human oversight, and clear disclosure when generated responses are used.

REPRODUCIBILITY STATEMENT

The main paper describes the StreamingMEM-R1 architecture and training framework in Section 2, including memory-trajectory construction, stable–dynamic memory evolution, response timing, evidence routing, streaming-aware supervised fine-tuning, and memory-oriented reinforcement learning. The streaming KV-cache update and causal attention mechanisms are specified in Equations 7–9, while the self-conditioned rollout and trajectory-level reward formulation are given in Equations 11–12.

Appendix A describes the source records and causal prompt construction. The teacher annotation schema, answer verification procedure, independent visual-evidence verification, and output serialization are provided in Appendices A.3, A.4, A.5, and A.6, respectively. Dataset sources, baseline configurations, and training-data statistics are reported in Appendices C and D.1, with detailed statistics in Table 5.

The optimization settings for supervised fine-tuning and RLVR are given in Appendices D.2 and D.3, and the hardware and computational environment are reported in Appendix D.4. Evaluation protocols, ablations, and efficiency measurements are provided in Appendices E–E.2, including the full ablation results in Tables 6 and 7, and the overall efficiency results in Table 8. Together, these descriptions specify the data construction, model training, streaming inference, and evaluation procedures needed to reproduce the reported experiments.

REFERENCES

Vatsal Agarwal, Saksham Suri, Matthew Gwilliam, Pulkit Kumar, and Abhinav Shrivastava. Going down memory lane: Scaling tokens for video stream understanding with dynamic kv-cache memory. *arXiv preprint arXiv:2602.18434*, 2026.

Anthropic. Claude 3.5 sonnet. 2024. URL `https://www.anthropic.com/news/claude-3-5-sonnet`.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang

Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025b. URL https://api.semanticscholar.org/CorpusID:276449796.

Lorenzo Basile, Valentino Maiorca, Diego Doimo, Francesco Locatello, and Alberto Cazzaniga. Head pursuit: Probing attention specialization in multimodal transformers. *arXiv preprint arXiv:2510.21518*, 2025.

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

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024b.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. Streamingtom: Streaming token compression for efficient video understanding. *arXiv preprint arXiv:2510.18269*, 2025.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3120–3128, 2026.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024c.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval (2025). *URL https://arxiv. org/abs/2503.00540*, 2025.

Jiajun Fei, Dian Li, Zhidong Deng, Zekun Wang, Gang Liu, and Hui Wang. Video-ccam: Enhancing video-language understanding with causal cross-attention masks for short and long videos. *arXiv preprint arXiv:2408.14023*, 2024.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025.

Zifan Han, Hongbo Sun, Jinglin Xu, Canhui Tang, Yulong Lei, Xuchong Zhang, Hongbin Sun, Zhongjiang He, and Hao Sun. Wat: Online video understanding needs watching before thinking. *arXiv preprint arXiv:2603.13412*, 2026.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13504–13514, 2024.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding. In *European conference on computer vision*, pp. 709–727. Springer, 2020.

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.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *arXiv preprint arXiv:2506.15745*, 2025.

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*, 2024a.

Kangcong Li, Peng Ye, Lin Zhang, Chao Wang, Huafeng Qin, and Tao Chen. Freshmem: Brain-inspired frequency-space hybrid memory for streaming video understanding. *arXiv preprint arXiv:2602.01683*, 2026.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102, 2025a.

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

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26689–26699, 2024.

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

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024.

Jinming Liu, Jianguo Huang, Zhaoyang Jia, Jiahao Li, Xiaoyi Zhang, Zongyu Guo, Bin Li, Wenjun Zeng, Yan Lu, and Xin Jin. An efficient streaming video understanding framework with agentic control. *arXiv preprint arXiv:2605.17921*, 2026a.

Shuming Liu, Chen Zhao, Tianqi Xu, and Bernard Ghanem. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3318–3327, 2025.

Yuxin Liu, Peiqin Zhuang, and Yali Wang. Streamems: Streaming video understanding with self-evolving memory scheme for vision-language models. *arXiv preprint arXiv:2608.27881*, 2026b.

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*, 2026c.

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

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

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24045–24055, 2025.

Hongyu Qu, Guangming Yao, Ling Xing, Xiaobin Hu, Rongxing Ding, Guibin Zhang, Fan Zhang, Yi Yuan, Xiangbo Shu, and Shuicheng Yan. Beyond retrieval: Progressive latent memory evolution for streaming video understanding. *arXiv preprint arXiv:2609.04131*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024.

Hongyuan Tao, Bencheng Liao, Shaoyu Chen, Haoran Yin, Qian Zhang, Wenyu Liu, and Xinggang Wang. Infinitevl: Synergizing linear and sparse attention for highly-efficient, unlimited-input vision-language models. *arXiv preprint arXiv:2512.08829*, 2025a.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18992–19001, 2025b.

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.

Qwen Team. Qwen3.5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

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. *arXiv preprint arXiv:2505.05467*, 2025a.

Han Wang, Yuxiang Nie, Yongjie Ye, Yanjie Wang, Shuai Li, Haiyang Yu, Jinghui Lu, and Can Huang. Dynamic-vlm: Simple dynamic visual token compression for videollm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20812–20823, 2025b.

Junxi Wang, Te Sun, Jiayi Zhu, Junxian Li, Haowen Xu, Zichen Wen, Xuming Hu, Zhiyu Li, and Linfeng Zhang. Streammeco: Long-term agent memory compression for efficient streaming video understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13234–13251, 2026.

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.

Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long streaming video understanding with recurrent memory bridges. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24170–24181. IEEE, 2025c.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3272–3283, 2025d.

Siwei Wen, Zhangcheng Wang, Xingjian Zhang, Lei Huang, and Wenjun Wu. Eventmemagent: Hierarchical event-centric memory for online video understanding with adaptive tool use. *arXiv preprint arXiv:2602.15329*, 2026.

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

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*, 2024.

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

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

Boshen Xu, Zihan Xiao, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Jian Luan, and Qin Jin. Timeviper: A hybrid mamba-transformer vision-language model for efficient long video understanding. *arXiv preprint arXiv:2511.16595*, 2025.

Haolin Yang, Feilong Tang, Lingxiao Zhao, Xiang An, Ming Hu, Huifa Li, Xinlin Zhuang, Yifan Lu, Xiaofeng Zhang, Abdalla Swikir, et al. Streamagent: Towards anticipatory agents for streaming video understanding. *arXiv preprint arXiv:2508.01875*, 2025a.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025b.

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*, 2025c.

Zhenyu Yang, Kairui Zhang, Bing Wang, Shengsheng Qian, and Changsheng Xu. Livestarpro: Proactive streaming video understanding with hierarchical memory for long-horizon streams. *arXiv preprint arXiv:2606.17798*, 2026.

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

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. *arXiv preprint arXiv:2509.24871*, 2025.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 21059–21069, 2025.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. *arXiv preprint arXiv:2601.14724*, 2026.

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*, 2024a.

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*, 2024b.

Yikai Zheng, Xin Ding, Yifan Yang, Shiqi Jiang, Hao Wu, Qianxi Zhang, Weijun Wang, Ting Cao, and Yunxin Liu. Em-garde: A propose-match framework for proactive streaming video understanding. *arXiv preprint arXiv:2603.19054*, 2026.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*, 2024.

A Data-generation and Verification Prompts

We employ Qwen3.5-35B-A3B as the multimodal teacher to convert annotated streaming videos into causal memory trajectories. This section describes the source data, prompt-construction procedure, teacher output schema, and verification rules. Dynamic values are represented by placeholders, while the fixed instructions follow those used in our data-generation pipeline.

A.1 Source Data Types

The synthesis pipeline accepts two complementary types of source records.

Native streaming records. Each record contains a video, timestamped user–assistant conversations, and optional timestamped event descriptions. Depending on the interaction mode, the desired behavior can be:

- **Continuous output**: describe a newly observed event without repeatedly narrating unchanged content;
- **Event trigger**: remain silent until a specified event becomes visually satisfied, respond once, and preserve its reported state;
- **Real-time dialogue**: answer once sufficient causal evidence becomes available.

Responses may be free-form text or multiple-choice answers. Both formats are retained in the constructed trajectories. During trajectory filtering, a multiple-choice response must resolve to the canonical option and exactly match the dataset gold label, whereas a free-form response is retained only when a separate teacher-model judgment finds it semantically equivalent to the dataset gold answer. Thus, gold-answer verification is format-specific rather than restricted to multiple-choice records.

Verified memory-augmented records. These records additionally contain an audited memory route and temporally localized evidence:

$$\mathcal{D}_{\text{aug}} = \left(q, y, r^*, E^H, E^C\right), \tag{13}$$

where q is the query, y is the gold answer, $r^* \in \{\text{LTM}, \text{BOTH}\}$ is the target route, and E^H and E^C denote historical and current evidence, respectively. Such records are admitted only after independent video-evidence verification.

A.2 Causal Prompt Construction

A video is divided into temporally ordered chunks $\mathcal{V} = \{v_1, \ldots, v_T\}$. For each chunk v_t, we construct one teacher request:

$$P_t = \left(P_{\text{static}}, v_t, \mathcal{R}_t, S_t\right), \tag{14}$$

where P_{static} contains the shared annotation policy and few-shot examples, $\mathcal{R}_t$ contains optional historical retrieval videos, and S_t is the chunk-specific causal state.

Static instruction. The shared system prompt instructs the teacher as follows:

> You are a data annotation teacher for a streaming video agent. Convert the provided causal streaming state into one compact two-level memory-evolution target. Use only the attached

> current video chunk, the previous memory state, the currently available query, and supplied past responses. Do not infer or reveal future events.
>
> The attached frames are the primary evidence for the current time window. Queries, source response timestamps, candidate labels, and common-sense assumptions are not visual evidence. Preserve temporal relations such as before/after, ongoing/completed, and awaiting/reported.
>
> Declare which evidence is necessary for the decision. Current visual evidence and previous Dynamic Memory are short-term evidence. Stable Memory is long-term evidence and may be retrieved only when indispensable. Return valid JSON only and do not include Markdown.

The prompt also defines the evidence-to-route mapping:

$$\texttt{INIT_CURRENT} \rightarrow \texttt{STM}, \tag{15}$$

$$\texttt{SHORT_TERM_PROCESSING} \rightarrow \texttt{STM}, \tag{16}$$

$$\texttt{STABLE_RETRIEVAL} \rightarrow \texttt{LTM}, \tag{17}$$

$$\texttt{CURRENT_PLUS_STABLE} \rightarrow \texttt{BOTH}. \tag{18}$$

The route is determined by the evidence actually used, independently of which memory slots are subsequently modified.

Chunk-specific causal state. The dynamic textual payload is instantiated as:

```
{
  "sample_id": <sample identifier>,
  "metadata": {
    "video_path": <video path>,
    "interaction_mode": <continuous, trigger, or dialogue>,
    "content_dimension": <task category>,
    "temporal_scope": <current, past, future, or mixed>,
    "response_format": <free-form or multiple-choice>
  },
  "current_time_window": [<chunk start>, <chunk end>],
  "user_query": {
    "timestamp": <query timestamp or null>,
    "text": <query or empty>,
    "available_in_current_window": <boolean>
  },
  "causal_state": {
    "previous_memory_state": <complete previous memory>,
    "previous_stable_memory": <stable-memory content>,
    "previous_dynamic_memory": <dynamic-memory content>,
    "past_assistant_responses": <responses before this chunk>
  },
  "route_supervision": <causally available constraints>,
  "response_timing_hints": {
    "source_timestamp_is_hint_only": true,
    "candidate_turns": <available response-turn IDs>
  }
}
```

A query is included only after its annotated arrival time. Before that time, query-related metadata is hidden and the teacher performs query-independent memory maintenance. Candidate response turns contain identifiers and already reached timestamps but not their answer text; therefore, they can guide temporal alignment without leaking the target answer.

Visual inputs. Every request contains the current video chunk. For an STM decision, no historical video is attached. For LTM or BOTH, the pipeline retrieves a bounded set of strictly earlier video windows:

$$\mathcal{R}_t = \mathrm{TopK}\left(\{v_j : j < t\}, \mathrm{sim}(q, e_j)\right), \tag{19}$$

where e_j is an offline historical evidence index. Only the selected video frames and their timestamps are shown to the teacher; the indexed evidence text is withheld. The input order is explicitly specified as `CURRENT`, followed by `HISTORICAL_1`, . . ., `HISTORICAL_K`.

A.3 Teacher Annotation Schema

The teacher returns one JSON object:

```
{
  "route_basis": "INIT_CURRENT | SHORT_TERM_PROCESSING |
                  STABLE_RETRIEVAL | CURRENT_PLUS_STABLE",
  "route_purpose": "INITIALIZE | CURRENT_TRACKING |
                    HISTORICAL_QUERY | RESPONSE_LIFECYCLE |
                    MEMORY_EVOLUTION",
  "mem_action": "STM | LTM | BOTH",
  "uses_current_visual": <boolean>,
  "uses_prior_dynamic": <boolean>,
  "need_response": <boolean>,
  "selected_response_turn_id": <integer or null>,
  "trigger_state": "NOT_APPLICABLE | AWAITING |
                    SATISFIED | REPORTED",
  "stm_obs": <observation from the current video>,
  "prior_dynamic_obs": <necessary dynamic-memory fact or empty>,
  "ltm_obs": <necessary stable-memory fact or empty>,
  "updated_memory": "Stable Memory: Semantic Events: ...
                     Dynamic Memory: Ongoing Event: ...",
  "stable_write_reason": "NONE | SALIENT_ATTRIBUTE | OCR_TEXT |
                          COUNT_IDENTITY_RELATION | COMPLETED_EVENT |
                          PHASE_BOUNDARY",
  "stable_update_evidence": <supporting evidence or empty>,
  "response_evidence": <concise causal evidence>,
  "assistant_response": <free-form answer or empty>
}
```

Stable Memory retains compact facts that remain useful after leaving the current visual window, including completed events, phase boundaries, OCR text, counts, identities, and salient attributes. Dynamic Memory represents the current ongoing event, recent transition, or unresolved evidence. The validator rejects duplicated, unsupported, or append-only memory states.

The memory operation is derived by comparing the previous and updated memory:

$$a_t^{\text{mem}} \in \{\texttt{INIT}, \texttt{RETAIN}, \texttt{UPDATE_DYNAMIC}, \texttt{UPDATE_STABLE}, \texttt{UPDATE_BOTH}\}. \tag{20}$$

A five-step reasoning trace is then rendered from the validated fields, covering current-event recognition, memory comparison, memory evolution, memory routing, and response decision.

A.4 Answer Supervision and Format-Specific Verification Without Leakage

The dataset gold answer is retained privately by the validator and is never included in the annotation prompt. Candidate trajectories are verified through separate branches for multiple-choice and free-form records.

Multiple-choice answers. The teacher must first produce a free-form answer using only the visible query and causal evidence.

Let $T(\hat{y})$ denote the normalized content tokens in the teacher answer and $T(y_j)$ the tokens of option j. The answer is accepted only when

$$|T(\hat{y}) \cap T(y^*)| \geq \min(2, |T(y^*)|) \quad \wedge \quad |T(\hat{y}) \cap T(y^*)| > \max_{j \neq *} |T(\hat{y}) \cap T(y_j)|. \tag{21}$$

This requires the generated answer to support the correct option more strongly than every distractor. After successful verification, the free-form response is normalized to a canonical option letter and

retained only if that letter exactly matches the dataset gold label. Consequently, the teacher controls evidence grounding and response timing, while the final answer label remains deterministic.

Free-form answers. For open-ended records, the generated response and the dataset gold answer are submitted to a separate teacher-model call for semantic-equivalence judgment. The verifier accepts paraphrases and differences in wording only when the two answers express the same task-relevant meaning; responses that contradict, omit, or add unsupported answer-critical information are rejected. Only semantically matched candidates are retained, so the trajectory corpus includes verified free-form supervision in addition to multiple-choice supervision.

A.5 Independent Evidence Verification

Memory-augmented LTM and BOTH records undergo a separate verification stage. For each candidate, Qwen3.5-35B-A3B independently inspects the historical and current video windows using the following prompt:

> You are an independent video-evidence verifier. Inspect only the attached video frames for the requested evidence side. Candidate labels and captions are claims to test, not visual evidence.
>
> For `HISTORICAL`, describe question-relevant evidence in the historical frames and determine whether those frames alone identify the complete answer. For `CURRENT`, perform the same test using only the query-time frames. Describe only visible objects, actions, attributes, counts, OCR strings, spatial relations, or states. Never infer intention, emotion, likely cause, or unseen events. Return one JSON object and no Markdown.

The output schema is:

```
{
  "side": "HISTORICAL | CURRENT",
  "fact_visible": <boolean>,
  "visible_evidence": <short visual fact>,
  "side_sufficient": <boolean>,
  "supported_option": "A | B | C | D | empty",
  "reason": <short justification>
}
```

An LTM candidate is retained only if the historical window supports the gold answer and the current window cannot independently answer the query. A BOTH candidate is retained only if the required historical and current facts are independently visible while neither side alone determines the complete answer.

A.6 Final Serialization and Validation

After verification, annotations are serialized into two target types:

$$z_t^{\text{silent}} = \langle \texttt{MEM_OPERATION} \rangle \langle \texttt{MEM_UPDATED} \rangle \langle \texttt{silent} \rangle, \tag{22}$$

$$z_t^{\text{response}} = \langle \texttt{MEM_OPERATION} \rangle \langle \texttt{MEM_UPDATED} \rangle \langle \texttt{MEM:STM|LTM|BOTH} \rangle \langle \texttt{think} \rangle \langle \texttt{response} \rangle. \tag{23}$$

Silent turns contain neither a route token nor a reasoning trace because routing is defined as an answer-time decision. Response turns contain exactly one route. LTM and BOTH responses must be accompanied by causal historical retrieval windows, while STM responses cannot contain unnecessary retrieval.

Finally, a fail-closed validator checks JSON structure, query availability, temporal causality, route–evidence consistency, memory grammar, response lifecycle, answer grounding, unsupported names or numbers, and reasoning-step completeness. Invalid annotations are rejected and regenerated with validation feedback; irreparable or intrinsically inconsistent source records are quarantined.

B Related Works

B.1 Video Understanding in the Streaming Regime

Conventional video-language models typically assume that a complete video, or a uniformly sampled subset of its frames, is available before inference (Cheng et al., 2024; Li et al., 2024a; Wang et al., 2024; Chen et al., 2024c; Li et al., 2025a). Although effective for offline video question answering, this batch-processing paradigm incurs growing memory and computation as video duration increases and cannot naturally support questions interleaved with a live stream. Memory-augmented streaming interaction addresses this gap by propagating compact historical knowledge across an unfolding video (Xiong et al., 2025), while VideoLLM-online formalizes online video dialogue with temporally aligned video–text sequences and supports responses before the complete video is observed (Chen et al., 2024a). These works establish streaming video understanding as a distinct problem in which perception, memory, and interaction must operate causally rather than over a fully observed video.

Model development has also explored compact, scalable backbones such as MiniCPM (Hu et al., 2024); our work is complementary and focuses on temporal memory and causal streaming behavior rather than reducing backbone size.

Recent systems further improve real-time perception and interaction. TimeChat-Online removes temporally redundant visual tokens using differential token dropping and constructs streaming instructions for backward tracing, current perception, and future responding (Yao et al., 2025). StreamBridge equips offline Video-LLMs with a round-decayed memory buffer and a lightweight activation model for proactive interaction (Wang et al., 2025a). Dispider instead disentangles perception, decision, and reaction into asynchronous modules, allowing video monitoring to continue while a response is generated (Qian et al., 2025). StreamAgent further studies anticipatory behavior in streaming agents (Yang et al., 2025a). While these methods improve efficiency or interaction timing, they do not explicitly learn coordinated short-term and long-term memory evolution as part of a unified generative policy.

In parallel, long-video systems reduce visual computation through token pruning, dynamic compression, adaptive hierarchies, memory augmentation, and sparse or linear attention (Chen et al., 2024b; Tao et al., 2025b; Yang et al., 2025b; Wang et al., 2025b;d; Song et al., 2024; He et al., 2024; Liu et al., 2025; Xu et al., 2025; Tao et al., 2025a). These techniques improve scalability for long inputs, but they do not by themselves define how semantic memory should evolve or when different temporal evidence should be used in a causal stream.

B.2 Memory and KV-Cache Management for Streaming Video

A major line of research represents video history using external visual memories. Flash-VStream maintains complementary context and augmentation memories to summarize long-range temporal content while retaining retrievable spatial details (Zhang et al., 2025). ReKV stores previously computed video KV caches outside GPU memory and retrieves query-relevant caches on demand, providing a training-free interface to existing Video-LLMs (Di et al., 2025). LiveVLM similarly combines a streaming-oriented cache with retrieval (Ning et al., 2025), while InfiniPot-V and Streaming-TOM compress cached visual history under bounded memory (Kim et al., 2025; Chen et al., 2025). StreamMem performs query-agnostic KV-cache compression with generic query tokens, thereby maintaining a fixed-size memory before the actual user question is known (Yang et al., 2025c). Although efficient, retrieval-based approaches generally expose the reasoning model to a sparse collection of frames or cached tokens; the resulting evidence may be fragmented and may not explicitly encode event evolution or temporal dependencies.

More recent methods impose hierarchical structure on streaming memory. HERMES interprets layer-wise KV caches as sensory, working, and long-term memory, and applies training-free hierarchical cache management, cross-layer smoothing, and position re-indexing to achieve low-latency inference without query-time retrieval (Zhang et al., 2026). FreshMem develops a frequency-space hybrid memory for streaming video (Li et al., 2026). WAT separates query-independent watching from query-triggered thinking: a FIFO short-term memory preserves recent frames, a redundancy-aware long-term memory retains semantically diverse history, and context-aware retrieval is trained with an auxiliary contrastive objective (Han et al., 2026). These designs demonstrate the importance

of separating recent and historical evidence, but their memory updates are largely governed by predefined buffering, eviction, compression, or retrieval rules. In contrast, StreamingMEM-R1 trains the model itself to summarize and update both memory levels, and to route each query to short-term, long-term, or joint evidence according to the evolving stream.

B.3 Learning to Reason and Respond over Video Streams

Beyond visual memory management, another direction introduces explicit reasoning and response-control mechanisms. ThinkStream formulates streaming understanding as a Watch–Think–Speak loop, retains intermediate reasoning traces as compressed semantic memory, and applies streaming reinforcement learning with verifiable rewards to learn answer timing and correctness (Liu et al., 2026c). Agentic and proactive approaches further learn control policies for deciding when to perceive, reason, or respond (Liu et al., 2026a; Zheng et al., 2026; Wen et al., 2026). This line of work shows that language-space reasoning can provide a compact alternative to retaining dense historical visual tokens. Its primary emphasis, however, is on reasoning and response control rather than explicit supervision of complementary semantic-memory transitions and answer-time evidence routing. We next distinguish this control-oriented direction from methods that adapt the memory representation itself as the stream evolves.

B.4 Memory Evolution in Streaming Video Understanding

Whereas the preceding subsection focuses on learned reasoning and response control, this subsection organizes methods by how the memory representation itself changes over time. Memory evolution encompasses adaptive retention of visual evidence, recurrent refinement of latent states, and consolidation of semantic knowledge. We distinguish these methods by what representation evolves, how its updates are determined, and whether memory management is jointly supervised with response generation.

Adaptive retention and retrieval. FluxMem (Xie et al., 2026) maintains hierarchical visual memory through adaptive temporal token selection and spatial consolidation. Its compression thresholds depend on input statistics, and its response trigger reuses scene-change signals. MemStream (Agarwal et al., 2026) addresses retrieval bias in dense video streams through adaptive key selection for sparse sliding-window attention and a mixture of internal and external retrieval experts. SAVEMem (Wu et al., 2026) incorporates a pseudo-question semantic prior into visual retention and adapts the retrieval scope through a query-dependent recency gate. Beyond visual caches, StreamMeCo (Wang et al., 2026) compresses agent memory graphs through representative-node sampling and entity-aware pruning, followed by time-decay retrieval. These approaches improve the preservation and accessibility of historical evidence. However, their retention and retrieval mechanisms do not explicitly train a generative policy to update persistent facts and ongoing states together with silence-or-response decisions.

Recurrent and self-evolving representations. VideoLLaMB (Wang et al., 2025c) preserves temporal context through recurrent memory bridges and retrieval over cached memory states. StreamEMS (Liu et al., 2026b) explicitly studies self-evolving memory: its Semantic Evolution Module progressively propagates information among related memory entities, while its Prior-informed Evolution Module uses historical similarity distributions to refine the current representation. These modules are trained through supervised fine-tuning. LatentStream (Qu et al., 2026) further develops query-conditioned latent memory evolution, iteratively retrieving and internalizing evidence from progressively broader temporal scopes. It uses a confidence-based reward and REINFORCE-style test-time optimization to refine latent states while keeping the backbone frozen. These methods demonstrate that streaming memory can already evolve beyond static storage. Their principal focus is recurrent or iterative refinement of internal representations, rather than explicit supervision of serialized semantic-state transitions and their associated response decisions.

Joint trajectory learning of semantic memory and response decisions. StreamingMEM-R1 focuses on learning an explicit memory-management policy over complete streaming trajectories. Stable memory represents persistent facts and completed events, while dynamic memory represents ongoing events and recent state changes. Memory maintenance continues at silent steps; response steps additionally specify whether short-term, long-term, or joint evidence is required. Our supervision directly covers memory operations, the resulting textual states, response decisions, and answer-time

routes. Memory-oriented RLVR then optimizes self-conditioned trajectories with verifiable rewards for operations, routing, timing, and answer correctness. This differs from both feature-level restructuring and confidence-driven test-time latent optimization: the key distinction lies in jointly learning interpretable semantic-state transitions and grounded response behavior, rather than in adaptive retention or retrieval alone. Historical visual replay remains available to recover details omitted by semantic compression.

C Datasets and Baselines

Datasets. Unlike offline video benchmarks such as Video-MME, EgoSchema, and MVBench (Fu et al., 2025; Mangalam et al., 2023; Li et al., 2024b), streaming benchmarks reveal video incrementally and require causal decisions. We evaluate StreamingMEM-R1 on three benchmark families for this setting.

OVOBench (Li et al., 2025b) evaluates online perception and reasoning under both real-time and backward-looking settings. Its Real-Time Visual Perception track consists of Optical Character Recognition (OCR), Action Recognition (ACR), Attribute Recognition (ATR), Spatial Understanding (STU), Future Prediction (FPD), and Object Recognition (OJR). Its Backward Tracing track consists of Episodic Memory (EPM), Action Sequence Identification (ASI), and Hallucination Detection (HLD). These abbreviations correspond to the columns reported in Table 1.

StreamingBench (Lin et al., 2026) presents videos progressively and issues questions at intermediate timestamps, allowing a model to use only the content observed up to each query time. Its Real-Time Visual Understanding tasks consist of Object Perception (OP), Causal Reasoning (CR), Clip Summarization (CS), Attribute Perception (ATP), Event Understanding (EU), Text-Rich Understanding (TR), Prospective Reasoning (PR), Spatial Understanding (SU), Action Perception (ACP), and Counting (CT). These abbreviations correspond to the columns reported in Table 2.

RVS-EGO and **RVS-MOVIE** evaluate retrospective visual search over long videos following the RVS protocol (Zhang et al., 2026). RVS-EGO uses egocentric streams and emphasizes recovering past first-person observations, whereas RVS-MOVIE contains diverse scenes, characters, and narrative transitions characteristic of movie understanding (Huang et al., 2020), placing greater demands on long-horizon memory and retrieval. Together, these benchmarks cover current-event perception, historical evidence recovery, and reasoning that combines recent and persistent context.

Baselines. We compare with the baseline groups reported under the official protocols of each benchmark. First, *general-purpose offline Video-LLMs* include LLaVA-Video-7B (Zhang et al., 2024b), Qwen2-VL-7B (Wang et al., 2024), LongVU-7B, Qwen2.5-VL at 3B, 7B, and 32B scales (Bai et al., 2025b), VILA-1.5-8B (Lin et al., 2024), LongVA-7B (Zhang et al., 2024a), LLaVA-NeXT-Video-32B (Liu et al., 2024), and LLaVA-OV-7B (Li et al., 2024a). These methods process a complete video or a sampled set of frames and provide references for conventional video understanding without an explicit online-memory policy.

Second, on StreamingBench, we include the *proprietary MLLMs* Gemini 1.5 Pro (Team et al., 2024), GPT-4o (Hurst et al., 2024), and Claude 3.5 Sonnet (Anthropic, 2024). Recent proprietary systems such as Gemini 2.5 further illustrate the continuing growth of multimodal long-context capabilities (Comanici et al., 2025).

Third, we compare with *online and streaming Video-LLMs*, including Flash-VStream (Zhang et al., 2025), VideoLLM-online (Chen et al., 2024a), Dispider (Qian et al., 2025), TimeChat-Online (Yao et al., 2025), StreamForest (Zeng et al., 2025), Streamo, WAT (Han et al., 2026), and ThinkStream (Liu et al., 2026c). These methods respectively study external visual memory, online interaction, asynchronous perception and response, visual-token reduction, streaming instruction tuning, or reasoning-based semantic memory. They constitute the most relevant comparison group because they operate causally over an unfolding video rather than assuming access to the complete sequence.

Finally, for RVS-EGO and RVS-MOVIE, we additionally compare with memory-oriented long-video methods: ReKV (Di et al., 2025), InfiniPot-V (Kim et al., 2025), StreamMem (Yang et al., 2025c), StreamingTOM (Chen et al., 2025), and HERMES (Zhang et al., 2026), together with Flash-VStream (Zhang et al., 2025). These baselines cover query-conditioned KV-cache retrieval, bounded

visual-memory compression, temporal online memory, and hierarchical cache management. In contrast, StreamingMEM-R1 explicitly learns the evolution of complementary short- and long-term memories, routes each response to STM, LTM, or both, and jointly optimizes memory transitions, response timing, and answer correctness through supervised trajectory learning and memory-oriented RLVR.

D Experimental Details

D.1 Dataset Information

Video sources. We construct the training corpus from the video sources released with ThinkStream (Liu et al., 2026c). These videos are drawn from curated subsets of LLaVA-Video-178K (Zhang et al., 2024b) and Tarsier2-Recap-585K, covering diverse scenes, activities, objects, and temporal structures. The source annotations span real-time dialogue, event-triggered interaction, and continuous output; they include past-, current-, and future-oriented questions in open-ended, multiple-choice, binary, and counting formats. Their content covers entity perception, actions, spatial relations, causal reasoning, procedural evolution, global context, and optical character recognition. From these sources, we first construct a large candidate corpus of approximately 800K chunk-level trajectory records. This deliberately over-generated pool provides broad coverage before quality filtering. Time-grounded captions and instructions are used only to construct supervision; the student model receives video frames as visual input.

Memory-trajectory synthesis. We use Qwen3.5-35B-A3B (Team, 2026) as the multimodal teacher to convert each candidate into a causal sequence of chunk-level memory transitions. At step t, the teacher receives the current chunk, the preceding stable and dynamic memories, the causally available query, and selected historical video evidence only when the target route is LTM or BOTH. It predicts a memory operation, the updated memories, and a silence-or-response decision; response steps additionally contain a memory route, a reasoning trace, and the final answer. Future chunks and dialogue turns are never exposed. Retaining both silent and answer-bearing steps provides supervision for memory evolution, response timing, routing, reasoning, and answer generation throughout the stream.

Visual-evidence verification, ranking, and repair. The initial 800K candidates are screened using the teacher model against the actual video evidence rather than captions alone. For each trajectory, the verifier checks whether the predicted stable and dynamic memories are supported by the causally visible current and historical frames, whether the selected route is compatible with the required evidence, and whether the answer matches the source supervision. Multiple-choice responses must match the canonical option, while free-form responses are checked for semantic agreement with the source answer. We additionally validate temporal causality, output structure, memory-state consistency, and response lifecycle. Failed chunks enter a retry pool with a maximum of five attempts using the same visual evidence and accumulated validation feedback; irreparable candidates are quarantined. The surviving trajectories are ranked by their aggregate answer, visual-grounding, causality, memory-consistency, and route-compatibility checks, prioritizing examples that provide reliable supervision for memory evolution.

Final dataset. From the ranked candidate pool, we select the top 150K verified trajectory records and balance them across interaction modes, temporal scopes, answer formats, and memory routes. This high-quality subset forms the final training corpus: 100K records for SFT and 50K records for RLVR. The SFT split comprises 7,844 videos, 9,852 causal chains, and 10,104 response steps; response-centered windowing produces 10,136 training sequences. The RLVR split comprises 3,469 videos, 3,884 chains, and 4,057 response steps, yielding 4,625 rollout sequences. To strengthen historical reasoning, the RLVR response steps are intentionally enriched for long-term evidence: 2,000 use LTM, 500 use BOTH, and 1,557 use STM. Selecting only highly ranked, visually grounded trajectories ensures that the stable/dynamic memory states and their transitions provide reliable supervision. Table 5 reports the complete statistics.

Table 5: Statistics of the synthesized training data. "Records" denotes chunk-level records, "chains" denotes continuous source sequences, and "windows" denotes sequences constructed by the trajectory sampler. Route counts are measured only at answer-bearing steps.

Split	Records	Videos	Chains	Windows	Responses	STM	LTM	BOTH
SFT	100,000	7,844	9,852	10,136	10,104	7,200	2,421	483
RLVR	50,000	3,469	3,884	4,625	4,057	1,557	2,000	500

D.2 Supervised Fine-Tuning

We initialize StreamingMEM-R1 from Qwen2.5-VL-7B-Instruct (Bai et al., 2025b). The visual backbone is frozen, while the visual merger, language model, and output head are optimized. We train for one epoch with AdamW, a peak learning rate of 1×10^{-5}, zero weight decay, cosine decay, a warm-up ratio of 0.03, and gradient clipping at 1.0. With a per-device batch size of 1 and eight gradient-accumulation steps, the effective global batch size is 64 across eight GPUs.

Videos are sampled at 1 FPS with at most 32 frames per chunk. To preserve complete response supervision in long streams, we construct response-aware windows of at most 32 chunks instead of truncating each trajectory at a fixed prefix. The context length is 131,072 tokens, with a 120,000-token input budget that leaves space for multimodal packing and target generation. The streaming cache uses a four-chunk visual attention window. Training uses bfloat16 precision, language-model gradient checkpointing, and DeepSpeed ZeRO-3. Checkpoint recomputation is disabled for the frozen visual backbone but retained for trainable modules.

D.3 Memory-Oriented RLVR

We initialize the policy from the SFT checkpoint and optimize it with Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Liu et al., 2026c). Each prompt produces a group of eight rollouts. Training runs for two epochs with AdamW, a learning rate of 2×10^{-7}, cosine decay, a warm-up ratio of 0.03, and gradient clipping at 1.0. The per-device prompt batch size is 2 without gradient accumulation. The frozen SFT checkpoint serves as the reference policy, with a KL coefficient of $\beta = 0.001$ to constrain policy drift.

RLVR rollouts use 1 FPS and adaptive chunk selection, with chunk boundaries determined for each trajectory and each chunk capped at 40 sampled frames. This setting matches the adaptive chunking protocol used in evaluation while bounding the visual-token budget. Each rollout contains at most twelve causally ordered chunks and terminates at the annotated response position. At each step, the policy generates at most 192 tokens using temperature 1.05, top-$k = 50$, and top-$p = 0.98$. The rollout KV cache is limited to 16,384 tokens, and policy reconstruction uses a maximum sequence length of 32,768 tokens. The policy relies on bounded streaming context and its evolving semantic memory. The visual backbone remains frozen, while the visual merger and language model are optimized.

For reproducibility, the implementation combines the five verifiable reward components as

$$\mathcal{R} = 0.2\mathcal{R}_{\text{route}} + 0.2\mathcal{R}_{\text{mem}} + 0.10\mathcal{R}_{\text{time}} + 0.50\mathcal{R}_{\text{ans}}, \tag{24}$$

Here, $\mathcal{R}_{\text{route}}$ evaluates routing and output structure; $\mathcal{R}_{\text{mem}}$ evaluates the memory operation and resulting stable/dynamic states; $\mathcal{R}_{\text{time}}$ evaluates the response-versus-silence decision; $\mathcal{R}_{\text{ans}}$ evaluates answer correctness. The larger answer weight keeps memory optimization grounded in downstream question answering.

D.4 Compute Environment

All experiments are conducted on one node with eight NVIDIA H20 GPUs, each providing 96 GB of memory. SFT and RLVR use bfloat16 mixed precision and DeepSpeed ZeRO-3. The same eight-GPU configuration is used for rollout generation and policy updates. Data loading is distributed across ranks, while fused GRPO loss computation and FlashInfer sampling reduce RLVR rollout and optimization overhead.

Table 6: **Ablation results on OVOBench.** We report performance on perception and reasoning tasks. The best result in each column is highlighted in bold.

Variant	Perception						Reasoning				Overall
	OCR	ACR	ATR	STU	FPD	Avg.	EPM	ASI	HLD	Avg.	
StreamingMEM-R1	**82.82**	**73.21**	**72.93**	**63.99**	**72.47**	**72.35**	**56.54**	**65.49**	**55.91**	**59.31**	**65.83**
w/o Memory Router Reward	76.51	70.64	70.69	60.67	65.35	68.36	45.79	61.49	51.36	52.88	60.62
w/o Memory Operation Reward	77.18	71.56	71.55	61.80	66.83	69.20	47.47	63.18	52.13	54.26	61.73
w/o Streaming KV Memory	74.50	68.00	68.50	55.50	62.00	65.00	39.49	55.49	43.80	46.26	55.63
w/o GRPO	75.00	69.50	69.80	58.20	65.00	66.90	43.49	59.49	49.90	50.96	58.93

Table 7: **Ablation results on StreamingBench.** The results demonstrate the contribution of memory routing, memory operation supervision, streaming KV memory, and GRPO optimization. The best result in each column is highlighted in bold.

Variant	OP	CR	CS	ATP	EU	PR	SU	ACP	CT	Avg.
StreamingMEM-R1	**86.73**	**79.17**	**88.03**	**87.24**	**74.37**	**86.23**	**85.31**	**75.14**	**63.63**	**80.71**
w/o Memory Router Reward	84.50	76.00	82.50	83.50	72.00	80.00	81.00	72.00	59.00	76.75
w/o Memory Operation Reward	84.80	76.80	84.00	84.50	72.50	81.50	82.50	72.80	55.80	77.37
w/o Streaming KV Memory	78.00	70.00	75.00	76.00	66.00	72.00	73.00	65.00	55.30	70.03
w/o GRPO	80.50	72.50	78.00	79.00	67.50	75.00	76.00	67.50	57.70	72.62

E Supplementary Experiments

E.1 Full Results of Ablation Studies

Tables 6 and 7 present the task-level results of the ablation studies summarized in Section 3.3. Removing any component consistently reduces performance on both benchmarks.

Memory routing is particularly important for tasks requiring temporally displaced evidence. Removing its reward causes the largest task-level drops on EPM (-10.75) and FPD (-7.12) in OVOBench, and on PR (-6.23) and CS (-5.53) in StreamingBench. Memory-operation supervision complements routing by controlling how stable and dynamic memories evolve. Its removal notably reduces EPM by 9.07 points and CT by 7.83 points, indicating its importance for historical recall and state accumulation.

Streaming KV memory has the largest overall contribution. Removing it decreases the aggregate score by 10.20 points on OVOBench and 10.68 points on StreamingBench. The largest reductions occur on EPM (-17.05), PR (-14.23), CS (-13.03), and SU (-12.31), confirming the importance of preserving coherent visual and semantic context across chunks.

Finally, removing GRPO lowers performance by 6.90 points on OVOBench and 8.09 points on StreamingBench, with pronounced degradation on EPM (-13.05), PR (-11.23), and CS (-10.03). This shows that supervised initialization alone is insufficient and that trajectory-level optimization improves long-horizon memory use and reasoning. Overall, the results demonstrate complementary roles: memory operations determine what information is retained, memory routing selects the evidence used for answering, streaming KV memory preserves context, and GRPO jointly optimizes these decisions over complete trajectories.

E.2 Supplementary Efficiency Analysis

Table 8 compares the efficiency of different streaming-inference implementations under the same 8.29B-parameter model. StreamingMEM-R1 achieves the lowest average and P95 latency, the highest throughput, and the lowest peak memory consumption. Compared with full-history recomputation, it reduces per-chunk latency from 8.74 to 1.43 seconds and peak memory from 70.8 to 27.6 GB, corresponding to reductions of 83.6% and 61.0%, respectively, while increasing throughput from 0.11 to 0.70 chunks per second. The intermediate variants attribute these efficiency gains to complementary systems components: replacing the full KV cache with a sliding cache substantially

Table 8: **Overall streaming-inference efficiency.** Latency is measured per video chunk. P95 denotes the 95th-percentile latency, and peak memory denotes the maximum GPU memory consumption.

Method	Params	Latency/Chunk ↓	P95 Latency ↓	Throughput ↑	Peak Memory ↓	StreamingBench ↑
Full History Recompute	8.29B	8.74 s	13.92 s	0.11 chunk/s	70.8 GB	80.23
Standard Full KV Cache	8.29B	3.46 s	4.83 s	0.29 chunk/s	61.7 GB	80.37
Sliding KV, w/o CUDA Graph	8.29B	2.38 s	3.17 s	0.42 chunk/s	34.9 GB	80.56
w/o In-place KV Shift	8.29B	2.09 s	2.91 s	0.48 chunk/s	32.8 GB	80.64
StreamingMEM-R1	**8.29B**	**1.43 s**	**1.92 s**	**0.70 chunk/s**	**27.6 GB**	**80.71**

Table 9: **Latency scaling with processed video length.** We report the per-chunk latency after processing different amounts of video history. Lower values indicate better temporal scalability.

Video Length	Chunks	Full Recompute	Full KV Cache	Sliding KV	StreamingMEM-R1
30 s	2	1.52 s	1.47 s	1.41 s	**1.28 s**
60 s	3	1.84 s	1.55 s	1.43 s	**1.29 s**
120 s	6	2.73 s	1.76 s	1.47 s	**1.31 s**
240 s	12	4.38 s	2.14 s	1.53 s	**1.34 s**
480 s	24	7.21 s	2.91 s	1.67 s	**1.39 s**
1200 s	60	16.84 s	5.26 s	2.03 s	**1.51 s**

reduces latency and memory, while in-place KV shifting and CUDA Graph execution further reduce cache-maintenance and kernel-launch overhead. Removing these optimizations does not cause a substantial accuracy degradation. StreamingBench scores vary only from 80.23 to 80.71 across all implementations, and the sliding-cache variants remain within 0.15 points of the complete system. These results indicate that the components evaluated here primarily improve computational efficiency rather than constitute the source of the model's predictive gains.

Table 9 further evaluates how per-chunk latency changes as the processed video history grows. Full recomputation scales poorly, increasing from 1.52 seconds at 30 seconds of video to 16.84 seconds at 1,200 seconds. A full KV cache reduces repeated computation but still grows to 5.26 seconds because its retained context expands with video length. Sliding KV bounds this growth more effectively, but its latency still increases from 1.41 to 2.03 seconds.

In contrast, StreamingMEM-R1 maintains nearly constant latency, increasing only from 1.28 to 1.51 seconds across the same range. At 1,200 seconds, it is approximately $11.2\times$ faster than full recomputation, $3.5\times$ faster than the full KV cache, and $1.3\times$ faster than standard sliding KV. These results demonstrate that bounded visual context, in-place cache maintenance, and graph-based decoding jointly enable stable long-horizon streaming inference.

F Case Studies

We present two qualitative comparisons with ThinkStream to examine how explicit memory evolution affects streaming decisions. The first case requires summarizing an action distributed across multiple chunks, whereas the second requires recovering a transient textual fact long after it disappears from the current view.

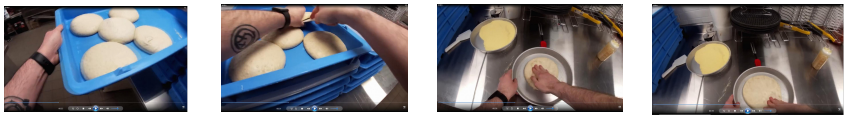
Case 1: Cross-chunk action summarization. Figures 7 and 8 show an 88-second video in which a worker progressively presses and stretches pizza dough into a baking pan. StreamingMEM-R1 remains silent while the action unfolds, continuously updating dynamic memory with the current state and stable memory with persistent action history. At the final chunk, it routes the response to `BOTH`, combines the earlier preparation sequence with the completed untopped state, and correctly selects option A. By contrast, ThinkStream retains only a coarse description of pizza preparation; it can no longer distinguish dough stretching from topping application and incorrectly selects option D. This comparison illustrates how coordinated stable–dynamic memory evolution preserves both temporal progression and fine-grained action details needed for grounded summarization.

Question: Which of the following options best summarizes the content of the video just now?
"A": "The individual pressed and stretched pizza dough into a baking pan, ensuring it was evenly spread.",
"B": "The individual prepared a salad, adding various toppings and dressing.",
"C": "The individual assembled a sandwich, layering different meats and cheeses.",
"D": "The individual applied sauce and added cheese to a cooked pizza crust."



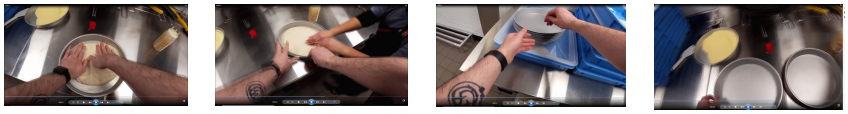
Chunk 0-20s

<MEM_OPERATION>INIT</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: A kitchen worker washes hands and retrieves trays containing round pizza-dough balls.</MEM_UPDATED><silent>



Chunk 20-40s

<MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION><MEM_UPDATED>Stable Memory: The worker is preparing pizza dough from blue storage trays. Dynamic Memory: Round baking pans and a yellow coating are arranged on the work surface.</MEM_UPDATED><silent>



Chunk 40-60s

<MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION><MEM_UPDATED>Stable Memory: The worker coats a round baking pan, places pizza dough inside, and begins pressing it flat. Dynamic Memory: Both hands press the dough outward from the center.</MEM_UPDATED><silent>



Chunk 60-80s

<MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION><MEM_UPDATED>Stable Memory: The worker placed pizza dough in a coated round pan and stretched it outward toward the rim. Dynamic Memory: The dough is being pressed repeatedly to create an even layer.</MEM_UPDATED><silent>



Chunk 80-88s

<MEM:BOTH><MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION><MEM_UPDATED>Stable Memory: The worker pressed and stretched pizza dough into a coated baking pan. Dynamic Memory: Final adjustments leave the dough evenly spread to the rim; no toppings are added.</MEM_UPDATED><think>Step 1. Current Event Recognition: The worker is finishing an even layer of dough in a round baking pan, with no toppings present. Step 2. Memory Comparison: Stable Memory supplies the earlier steps of placing, pressing, and stretching the pizza dough; current Dynamic Memory supplies the completed even coverage. Step 3. Memory Evolution: UPDATE_BOTH records the completed action sequence and the final untopped state. Step 4. Memory Routing: Use BOTH because the full summary requires the historical preparation sequence and the current completion state. Step 5. Response Decision: Only option A describes pressing and stretching dough evenly in a baking pan.</think><response>A</response>

Figure 7: Case 1 with StreamingMEM-R1. Explicit memory updates preserve the dough-preparation sequence, enabling the correct response through joint stable and dynamic memory.

Case 2: Long-horizon historical OCR retrieval. Figures 9 and 10 consider a query issued at 198.7 seconds, when the current racecourse view contains no readable date. StreamingMEM-R1 consolidates "SEPT 16, 2023" into stable memory when the dated broadcast strip first appears. At query time, it preserves this fact, selects the `LTM` route, and correctly returns option B. ThinkStream instead retains only that an information strip appeared, without its exact content, and chooses the superficially plausible but incorrect option A. The result demonstrates the value of storing transient yet query-relevant evidence in stable memory and accessing it selectively when current observations are insufficient.

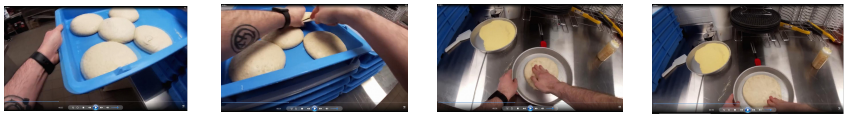
Together, these cases show that StreamingMEM-R1 supports complementary forms of temporal reasoning: dynamic memory tracks evolving local states, stable memory preserves durable or easily missed evidence, and response-conditional routing selects the appropriate source only when an answer is required.

Question: Which of the following options best summarizes the content of the video just now?
"A": "The individual pressed and stretched pizza dough into a baking pan, ensuring it was evenly spread.",
"B": "The individual prepared a salad, adding various toppings and dressing.",
"C": "The individual assembled a sandwich, layering different meats and cheeses.",
"D": "The individual applied sauce and added cheese to a cooked pizza crust."



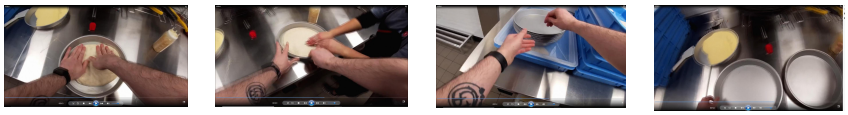
Chunk 0-20s

<think>A kitchen worker washes their hands and handles food-storage trays.</think><silent>



Chunk 20-40s

<think>The worker arranges trays and several round pans for food preparation.</think><silent>



Chunk 40-60s

<think>The worker puts food into a round pan and presses it by hand.</think>



Chunk 60-80s

<think>The worker continues preparing a round pizza base in the pan.</think><silent>

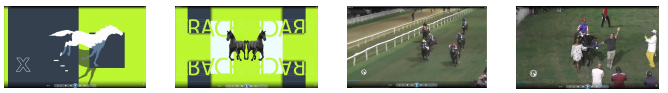


Chunk 80-88s

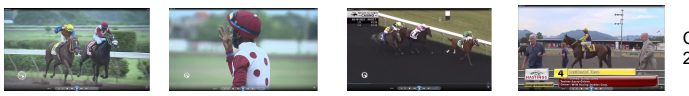
<think>The current close-up shows a round pizza base being finished. The compressed history does not retain whether the key sequence was dough stretching or topping application. I map the generic pizza-preparation caption to option D.</think><response>**D**

Figure 8: Case 1 with ThinkStream. The compressed history loses the distinction between dough stretching and topping application, resulting in an incorrect response.

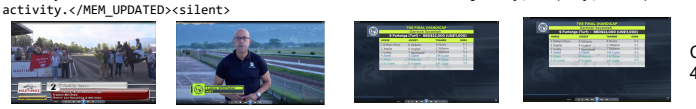
"What date is the event taking place?","A": "09-18-2023","B": "SEPT 16, 2023", "C": "Sep-16-2023", "D": "08-29-2023"
query time seconds: 198.7



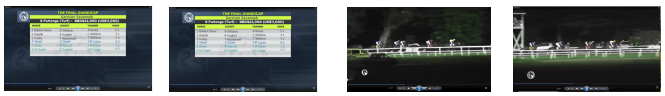
<MEM_OPERATION>INIT</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: A daytime horse race is underway on a grass track.</MEM_UPDATED><silent>



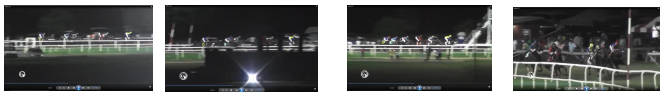
<MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: The daytime race finishes and the broadcast shows the jockey, replay, and post-race activity.</MEM_UPDATED><silent>



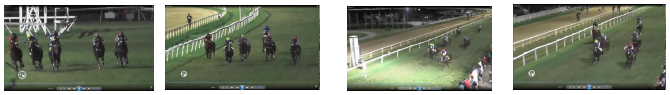
<MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: A trackside presenter introduces a final-handicap standings table.</MEM_UPDATED><silent>



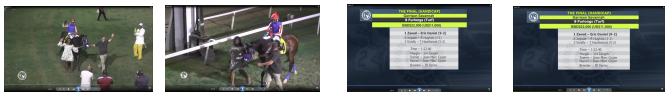
<MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: Horses race at night under bright track lights.</MEM_UPDATED><silent>



<MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: Spectators watch another daytime horse race.</MEM_UPDATED><silent>



<MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: A race finishes and the broadcast displays final-handicap results.</MEM_UPDATED><silent>



<MEM_OPERATION>RETAIN</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: A race finishes and the broadcast displays final-handicap results.</MEM_UPDATED><silent>



<MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Empty. Dynamic Memory: A second full-screen race-results table lists placements and names.</MEM_UPDATED><silent>



<MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION><MEM_UPDATED>Stable Memory: Broadcast event date: SEPT 16, 2023. Dynamic Memory: The horse-racing broadcast continues with a dated lower information strip.</MEM_UPDATED><silent>



<MEM:LTM><MEM_OPERATION>UPDATE_DYNAMIC</MEM_OPERATION><MEM_UPDATED>Stable Memory: Broadcast event date: SEPT 16, 2023. Dynamic Memory: Current view is a wide shot of the racecourse without readable date text.</MEM_UPDATED><think>Step 1. Current Event Recognition: The current frames show a wide racecourse view, but no readable date. Step 2. Memory Comparison: The question asks for a historical OCR fact recorded when the broadcast information strip was visible. Step 3. Memory Evolution: UPDATE_DYNAMIC preserves the stable date and records that the current view lacks readable date text. Step 4. Memory Routing: Use LTM and verify the stable fact against retrieved chunks 7 and 8. Step 5. Response Decision: The retrieved strip and Stable Memory both give SEPT 16, 2023, which is option B.</think><response>B</response>

Figure 9: Case 2 with StreamingMEM-R1. A transient date is consolidated into stable memory and later recalled through long-term memory routing.



Figure 10: Case 2 with ThinkStream. The compressed history records the presence of an information strip but omits the date, leading to an incorrect answer.

G EXAMPLES OF CONSTRUCTED TRAINING DATASETS

Here, we present several examples from our constructed training dataset to illustrate the structure and content of our data.

Example: Streaming Memory Training Sample

Item 1: Intermediate Streaming Step

Instruction: Your task is to process the current video chunk causally using the previous two-level memory and the available user query. First, select a memory operation from `INIT`, `RETAIN`, `UPDATE_DYNAMIC`, `UPDATE_STABLE`, and `UPDATE_BOTH`, and then update the stable and dynamic memories. If the available evidence is insufficient for answering, output the memory operation, updated memory, and `<silent>`. If the evidence is sufficient, additionally select exactly one route from `STM`, `LTM`, and `BOTH`, produce the five-step reasoning trace, and return the answer in `<response></response>`. Memory routing is permitted only when responding. Do not use information from future video chunks.

Input:
`Current Video Chunk: [40s--60s]`
A kitchen worker places pizza dough into a coated round baking pan and begins pressing it outward from the center.
`User Query:` Which option best summarizes the complete action?
`A.` The worker pressed and stretched pizza dough evenly into a pan.
`B.` The worker prepared a salad and added dressing.
`C.` The worker assembled a sandwich.
`D.` The worker added sauce and cheese to a pizza crust.
`Previous Memory:`
Stable Memory: The worker retrieved pizza dough and prepared a round baking pan.
Dynamic Memory: The worker is coating the pan before placing the dough.

Output:

```
<MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION>
<MEM_UPDATED>
Stable Memory: The worker retrieved pizza dough, coated a
round baking pan, and placed the dough inside.
Dynamic Memory: The worker is pressing the dough outward
from the center of the pan.
</MEM_UPDATED>
<silent>
```

Item 2: Answerable Streaming Step

Instruction: Your task is to process the current video chunk causally using the previous two-level memory and the available user query. First, select a memory operation from `INIT`, `RETAIN`, `UPDATE_DYNAMIC`, `UPDATE_STABLE`, and `UPDATE_BOTH`, and then update the stable and dynamic memories. If the available evidence is insufficient for answering, output the memory operation, updated memory, and `<silent>`. If the evidence is sufficient, additionally select exactly one route from `STM`, `LTM`, and `BOTH`, produce the five-step reasoning trace, and return the answer in `<response></response>`. Memory routing is permitted only when responding. Do not use information from future video chunks.

Input:
`Current Video Chunk: [80s--88s]`
The worker finishes stretching the dough to the rim of the pan. The dough is evenly spread, and no toppings have been added.
`User Query:` Which option best summarizes the complete action?
`A.` The worker pressed and stretched pizza dough evenly into a pan.
`B.` The worker prepared a salad and added dressing.
`C.` The worker assembled a sandwich.
`D.` The worker added sauce and cheese to a pizza crust.
`Previous Memory:`
Stable Memory: The worker retrieved pizza dough, coated a round baking pan, placed the dough inside, and began pressing it outward.
Dynamic Memory: The worker is repeatedly stretching the dough toward the rim.

Output:

```
<MEM:BOTH>
<MEM_OPERATION>UPDATE_BOTH</MEM_OPERATION>
<MEM_UPDATED>
Stable Memory: The worker coated a round baking pan, placed
pizza dough inside, and pressed and stretched it evenly to
the rim.
Dynamic Memory: The dough now forms an even untopped layer
across the pan.
</MEM_UPDATED>
<think>
Step 1. Current Event Recognition: The worker finishes
stretching the dough evenly to the rim, without adding toppings.
Step 2. Memory Comparison: Stable Memory records the earlier
placement and pressing steps, while the current chunk shows
the completed state.
Step 3. Memory Evolution: UPDATE_BOTH consolidates the full
action sequence and records its completion.
Step 4. Memory Routing: Use BOTH because the answer requires
the historical preparation process and the current final state.
Step 5. Response Decision: The combined evidence uniquely
matches option A.
</think>
<response>A</response>
```