An Efficient Streaming Video Understanding Framework with Agentic Control

Jinming Liu[1,2,*] **Jianguo Huang**[1,2,*] **Zhaoyang Jia**[3] **Jiahao Li**[3]
Xiaoyi Zhang[3] **Zongyu Guo**[3] **Bin Li**[3] **Wenjun Zeng**[2] **Yan Lu**[3] **Xin Jin**[2]

[1]Shanghai Jiao Tong University
[2]Eastern Institute of Technology, Ningbo, China [3]Microsoft Research Asia

Abstract

Streaming video requires handling dynamic information density under strict latency budgets. Yet, existing methods typically employ static strategies, like fixed memory compression or single-model reliance, forcing a trade-off: fast models fail on complex queries, while 'always-on' heavy models violate real-time constraints and overcomplicate some simple queries. Rather than fixing these decisions upfront, we propose R3-Streaming (Remember, Respond, Reason), which formulates streaming video understanding as a cascaded control problem: for each query, the system compresses memory, judges response readiness, and routes computation sequentially, so that each downstream decision builds on progressively refined information states. To optimize this pipeline, we introduce an age-aware forgetting policy for memory compression, as aggressively compressing historical frames can yield substantial performance gains. For compute routing, we propose TB-GRPO, a target-balanced reinforcement learning (RL) objective that routes hard queries to a stronger model while preventing mode collapse. Extensive evaluations demonstrate that R3-Streaming achieves state-of-the-art results among streaming MLLMs, reaching 57.92 on OVO-Bench and 76.36 on Streaming-Bench, all while reducing visual token usage by 95%–96%.

1 Introduction

Recent video-language models perform well on offline video understanding, including long-context tasks such as captioning, summarization, and question answering (Li et al., 2024b; Zhou et al., 2024; Wu et al., 2024; Shen et al., 2025). However, most of these systems process full clips and are hard to deploy in real-time streams, where frames arrive continuously and each decision must satisfy strict latency constraints.

Existing streaming methods usually optimize one part of the pipeline at a time, such as response triggering or token/frame compression (Chen et al., 2024; Yao et al., 2025; Jin et al., 2025; Wang et al., 2025b; Xu et al., 2025). While these methods improve efficiency, they often hit a performance bottleneck where answer quality degrades sharply under strict latency constraints. This suboptimal trade-off arises because memory retention, response timing, and reasoning depth are not dynamically coordinated, and therefore fail to adapt to the varying informational density of real-time streams.

Therefore, we reframe streaming video understanding not as a passive, feed-forward task, but as an agentic control problem. A streaming agent can autonomously manage its own state and dynamically decide its compute path through three sequentially coupled decisions per turn: (i) Remember, how much historical evidence should remain in memory; (ii) Respond, whether the current context supports a reliable answer; and (iii) Reason, whether the query should be solved by an advanced slow model.

This formulation is motivated by two empirical findings (Sec. 3), as illustrated in Fig. 1. First, in Fig. 1a, we attempt to analyze the impact of removing historical tokens or nearby tokens on the final output distribution (measured by Jensen–Shannon divergence, JSD (Lin, 1991; Kim et al., 2020)), and find that historical tokens receive most of the visual attention, yet removing them has far less impact on the output token distribution than removing nearby tokens. This suggests that historical tokens contain substantial redundancy, which can induce misleading attention allocation away from more informative nearby tokens. Fig. 2 further illustrates this point — we find that aggressively compressing historical tokens can actually improve benchmark performance. Second, model-scale gains are non-monotonic: small models may outperform

[*]Equal contribution.

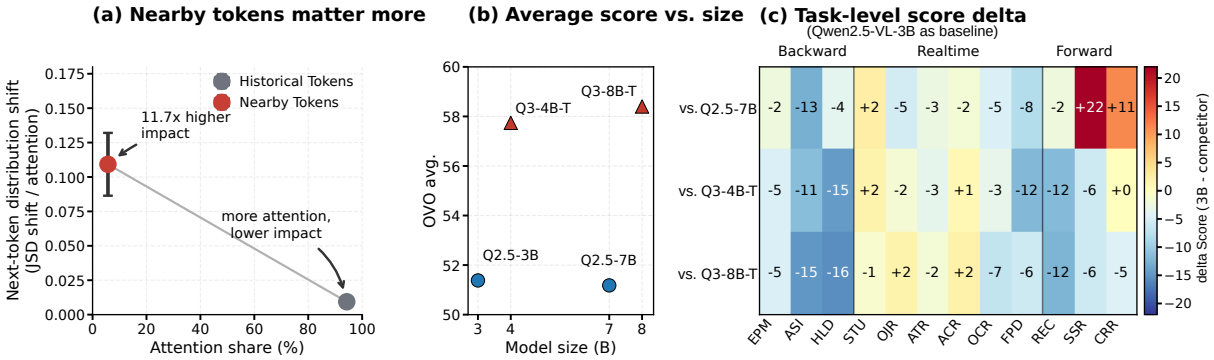


Figure 1: Empirical motivations for R3-Streaming. (a) Historical tokens receive most of the visual attention, yet deleting them has only a minor impact on the output token distribution. In contrast, removing nearby tokens induces much larger shifts in the next-token distribution. (b) Large models are not always better, where Q2.5/Q3 = Qwen2.5/3-VL, T = Thinking. (c) Per-task performance differences on OVO-Bench between Qwen2.5-VL-3B and larger models. Positive values indicate tasks where the 3B model performs better.

large thinking models on specific streaming tasks (Fig. 1b,c), so always invoking heavy "thinking" inference is not optimal.

Based on these findings and the formulation, we propose **R3-Streaming** (**R**emember, **R**espond, **R**eason), a three-decision policy system with: (i) Active Forgetting, a training-free age-aware memory compressor that reduces stale context while preserving recent frames at high fidelity; (ii) Proactive Response, a readiness head that emits `<Routine>` and defers answering when evidence is insufficient; and (iii) Adaptive Thinking, a routing policy where the fast model either answers directly or emits `<Escalate>` to invoke a heavier reasoning model for compute routing. The resulting framework is summarized in Fig. 3. Among the three decisions, learning the binary routing policy in Adaptive Thinking poses the greatest optimization challenge: standard reinforcement learning (GRPO (Shao et al., 2024)) and AutoThink methods (Tu et al., 2025) often struggle with stability, frequently leading to uncontrollable or excessively high escalation rates. The resulting policy may still invoke the heavy reasoning path at a frequency that exceeds the available compute budget. We address this with TB-GRPO, a target-balanced RL objective that stabilizes routing and keeps escalation rates within deployable compute budgets. Our contributions are:

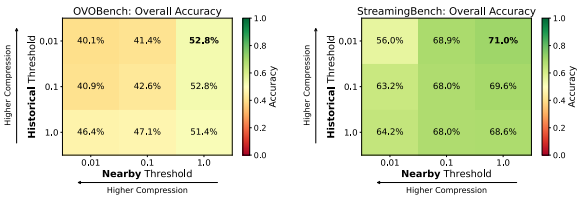
1. We establish two findings for streaming VLMs: important signal is strongly recent-focused, while historical tokens are often redundant and can lead to misleading attention; model-scale gains are non-monotonic across streaming tasks, motivating selective reasoning instead of always-on thinking.

2. We introduce R3-Streaming, a cascaded control framework whose core technical contributions are an age-aware forgetting policy for memory compression and TB-GRPO, a target-balanced RL objective that stabilizes compute routing within deployable escalation budgets.

3. We validate the framework on streaming benchmarks, reaching 57.92 on OVO-Bench and 76.36 on StreamingBench with 95%–96% visual-token reduction. On StreamingBench, our adaptive routing outperforms direct slow-only inference, achieving a better efficiency–accuracy trade-off.

2 Related Works

Online and long-video VLMs. Streaming VLMs process continuously arriving frames with state updates, token pruning, cache maintenance, or decision-reaction modules (Chen et al., 2024; Yao et al., 2025; Ning et al., 2025; Qian et al., 2025; Xu et al., 2025). In parallel, long-video VLMs and benchmarks study temporal reasoning and memory/compression over extended clips (Li et al., 2024b; Zhou et al., 2024; Wu et al., 2024; Wang et al., 2024b; Shen et al., 2025). Recent online benchmarks such as StreamingBench, OVBench, and OVO-Bench further expose timestamped understanding and reasoning challenges (Lin et al., 2024b; Huang et al., 2025; Li et al., 2025). However, most prior systems optimize memory retention, response timing, or answer quality separately. R3 instead treats streaming understanding as a joint control problem over what to remember, when to



(b) Compression ablation

Figure 2: Compression threshold ablations on OVO-Bench and StreamingBench. The results show that preserving nearby context while compressing history gives the best performance. Refer to Appendix B.2 for results across additional models and benchmarks.

respond, and when to invoke stronger reasoning.

Adaptive reasoning and routing. Adaptive decision methods study when models should deliberate versus answer directly. AutoThink uses multi-stage RL to learn when additional thinking is useful (Tu et al., 2025), and AdaptThink trains dynamic switching policies between thinking and no-thinking modes (Zhang et al., 2025b). These methods are not designed around strict streaming budgets, where excessive slow-model calls directly harm deployability. Our TB-GRPO routing objective adds target-band control over escalation frequency, preserving answer quality while keeping slow-path usage predictable.

3 Preliminary Findings

Before introducing module details, we summarize two empirical findings that motivate our design.

Finding 1: Important signal is strongly concentrated in recent context, while historical tokens often contain harmful redundancy and can induce misleading attention allocation. Fig. 1a and Fig. 2 summarize the evidence for Finding 1. The deletion analysis on OVO-Bench in Fig. 1a shows that Nearby Tokens cause much larger next-token distribution shifts than Historical Tokens, even though Historical Tokens receive most of the attention mass. This indicates that important signal is concentrated in nearby context, whereas historical memory may contribute noisy signals.

As a downstream benchmark check in Fig. 2, we observe a clear asymmetry: preserving **Nearby** context is critical, while **Historical** memory can be compressed more aggressively. On StreamingBench, the best setting (Nearby= 1.0, Historical= 0.01) reaches 71.0, which is +2.4 over no compression. In contrast, once nearby context is heavily compressed, performance drops much. This indicates that compressing historical tokens while retaining nearby tokens can mitigate the misleading attention and improve model performance.

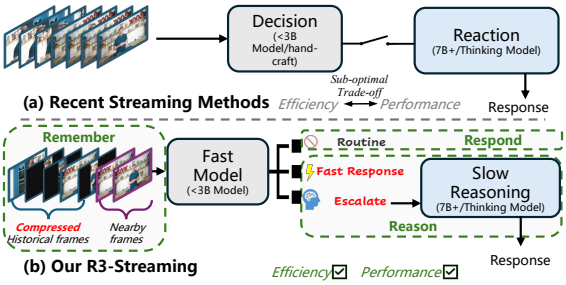


Figure 3: Framework comparison. (a) Recent decision-reaction streaming methods rely on a single reaction model, leading to a sub-optimal efficiency-performance trade-off. (b) R3-Streaming decomposes each streaming step into three cascaded decisions—memory compression (Remember), readiness estimation (Respond), and compute routing (Reason).

Finding 2: Model-scale gains are non-monotonic across streaming tasks. As shown in Fig. 1b, performance on streaming tasks does not scale monotonically with parameter size. Small fast models can match or outperform larger models on specific task groups. Notably, the Qwen2.5-VL-3B model exceeds the 7B model, while the Qwen3-VL-4B-Thinking model is comparable to Qwen3-VL-8B-Thinking. This counter-intuitive trend is further elucidated by the task-level deltas in the same panel. The 3B instruct model demonstrates superior performance over larger, thinking-enabled models specifically within the Realtime and Forward task groups. Such strong task heterogeneity indicates that 'always-on' heavy inference is not only compute-intensive but can also be sub-optimal for specific streaming perception tasks.

Implication for R3 design. These findings map directly to the R3 design. **Finding 1** motivates **Remember**: Active Forgetting keeps recent context at high fidelity and compresses old memory aggressively. **Finding 2** motivates **Reason**: Adaptive Thinking routes only hard queries to slow/thinking inference instead of using an always-on large model. **Respond** is inspired by recent streaming decision-reaction frameworks (Qian et al., 2025; Yang et al., 2025a), and is implemented here as a lightweight readiness head on the fast model to decide whether to answer now or defer.

4 Method

4.1 Overview and Problem Formulation

At each streaming step t, the system observes stream history $x_{1:t}$, receives query q_t, maintains a memory state M_t, and faces a chain of increasingly refined decisions (Fig. 3). First, the arriving frame must be assimilated into a compact memory state M_t without drowning subsequent reasoning in stale context (**Remember**). Given the updated memory, the system then evaluates whether the accumulated evidence already grounds a reliable answer or whether deferral is more prudent (**Respond**). Only when readiness is confirmed does the system commit compute—choosing between a lightweight direct path and a heavier reasoning path based on estimated query difficulty (**Reason**).

Because each downstream decision consumes the output of the preceding one, errors compound: a noisy memory misleads readiness estimation, which in turn triggers unnecessary escalation. This cascaded dependency motivates a coordinated system design rather than treating memory, readiness, and routing as unrelated decisions.

Formally, we model decision-making with the action space $A = \{$`<Answer>`, `<Escalate>`, `<Routine>`$\}$, where `<Answer>` returns grounded response y_t, `<Escalate>` delegates the query to a slow/thinking model, and `<Routine>` defers output when current evidence is insufficient. The objective is to maximize answer quality while minimizing latency and unnecessary slow-model calls under streaming causality. The following subsections detail each decision stage in the order it executes.

4.2 Remember (Active Forgetting)

Streaming history grows linearly, yet the signal that actually matters for the next answer concentrates overwhelmingly in recent frames (Sec. 3). This asymmetry makes a uniform compression schedule wasteful: it either discards recent context or retains irrelevant history that introduces attentional noise. We therefore partition the streaming history $x_{1:t}$ into two segments based on a temporal window W:

$$\begin{aligned} M_t = \text{Compress}(x_{t-W+1:t}, \tau_{near}) \\ \cup \text{Compress}(x_{1:t-W}, \tau_{hist}), \end{aligned} \tag{1}$$

where τ_{near} and τ_{hist} denote the compression thresholds for nearby and historical zones, respectively.

Guided by the observation that nearby context is critical for accuracy while stale history introduces noise and misleading attention (Sec. 3), we enforce $\tau_{near} \gg \tau_{hist}$: near-range memory keeps fine-grained tokens, while far-range memory is consolidated into compact episodic slots. Active Forgetting is training-free and introduces no additional trainable parameters. Tunable controls include recent-window size, ratio schedule, and compression operator; implementation details and ablations are analyzed in Sec. 5 and Sec. B.1, B.3.

4.3 Respond (Proactive Response)

With a compact memory state in hand, the system must next decide *whether* to answer at all. In a continuous stream, queries often arrive before critical visual evidence has appeared; answering prematurely leads to hallucination. Inspired by recent decision-reaction frameworks for real-time interaction (Qian et al., 2025; Wang et al., 2025b), we place a lightweight readiness gate before expensive reasoning. A readiness head h estimates whether available evidence supports grounded answering:

$$p_{\text{ready}} = h(q_t, M_t). \tag{2}$$

Here $p_{\text{ready}} \in [0, 1]$ is interpreted as the probability that the current query can be reliably answered by the fast path using the current memory state. The action rule is threshold-based:

$$a_t = \begin{cases} \text{emit } \texttt{<Routine>}, & p_{\text{ready}} < 0.5, \\ \text{continue to Reason}, & \text{otherwise.} \end{cases} \tag{3}$$

For optimization, Respond is trained after Reason SFT cold-start and TB-GRPO are completed: we freeze the fast VLM and train only a lightweight readiness head with SFT labels. Detailed Respond dataset construction (sample collection, labeling, and filtering) is provided in Appendix C. This decoupled stage keeps additional parameters minimal and avoids perturbing the optimized fast/slow router.

4.4 Reason (Adaptive Thinking)

Once a query passes the readiness gate, the system must decide *how much* computation to invest. Not all queries benefit from heavy reasoning—on perception-oriented streaming tasks, a 3B model can match or exceed a 7B thinking model (Sec. 3, Fig. 1b). The question is how to identify such queries at decision time without oracle knowledge

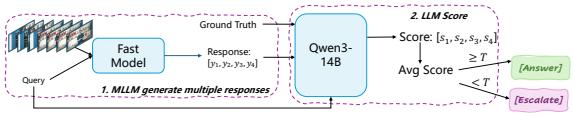


Figure 4: SFT cold-start data pipeline for routing targets. For each query, the fast model samples $K{=}4$ responses. Open-ended responses are scored by an external LLM, and objective responses are checked by exact correctness. The averaged score $\bar{s}$ is thresholded by T to assign <Answer> or <Escalate>. In this figure, y_i denotes the i-th sampled response from the fast model.

of their difficulty. We let the fast model either return a direct answer or emit <Escalate> to invoke the slow model. Learning this binary routing policy requires two stages: an SFT cold-start to establish the output format, followed by the proposed TB-GRPO to refine decision quality.

Stage 1: SFT cold-start. Before the model can learn *when* to escalate, it must first learn *how*—i.e., reliably produce the routing tokens <Answer> and <Escalate>. We construct answerability supervision via a multi-response pipeline (Fig. 4). For each training query, the fast model samples $K = 4$ responses $\{y^{(k)}\}_{k=1}^{K}$. For open-ended questions, an external LLM assigns quality scores $\{s^{(k)}\}$; for objective questions, we compute binary correctness against ground truth ($s^{(k)} \in \{0, 1\}$). We then aggregate $\bar{s} = \frac{1}{K}\sum_{k=1}^{K} s^{(k)}$, and assign routing ground truth by

$$\begin{cases} \texttt{<Answer>}, & \bar{s} \geq T, \\ \texttt{<Escalate>}, & \bar{s} < T. \end{cases} \tag{4}$$

where T is a quality threshold. The fast model is then fine-tuned on these mixed targets. This stage solves the format problem but not the decision-quality problem: SFT teaches the model to "speak the routing language" yet provides no reward signal for making *correct* routing choices.

Stage 2: Target-Balanced GRPO (TB-GRPO). The natural next step is RL to refine routing decisions with reward feedback. However, binary routing presents a problem: vanilla GRPO (Shao et al., 2024) rapidly collapses to a single mode—almost always <Escalate>—because the sparse reward signal makes it easier to "always escalate" than to discriminate. Prior adaptive-reasoning RL such as AutoThink (Tu et al., 2025) mitigates early collapse but does not explicitly control escalation frequency, leaving the ratio unpredictable under streaming latency budgets.

TB-GRPO addresses this by introducing *target-band control*: a piecewise penalty mechanism that anchors the escalation ratio around a user-specified operating point (η, γ), as illustrated in Fig. 5. Intuitively, when the current escalation ratio ρ drifts above $\eta{+}\gamma$, TB-GRPO suppresses the positive reward for escalation; when ρ drops below $\eta{-}\gamma$, it penalizes non-escalation—effectively applying proportional feedback control to routing frequency.

We now formalize this mechanism. For each query x, the fast routing policy $\pi_{\theta_{\text{policy}}}$ samples a response group $\{y_i\}_{i=1}^{G}$, where each y_i is either direct answer text or <Escalate>. Let $e_i = \mathbb{I}[y_i = \texttt{<Escalate>}]$ denote the escalation indicator, c_i the action-dependent correctness, and $\rho = \frac{1}{G}\sum_{i=1}^{G} e_i$ the group escalation ratio.

We first define a base reward that encodes deployment priorities:

$$r_i^{\text{naive}} = \begin{cases} 2, & e_i = 0,\ c_i = 1, \\ -1, & e_i = 0,\ c_i = 0, \\ 1, & e_i = 1,\ c_i = 1, \\ 0, & e_i = 1,\ c_i = 0. \end{cases} \tag{5}$$

A correct direct answer receives a higher reward (2) than a correct escalation (1), because it achieves quality with lower latency on the fast path. This asymmetry encodes the "answer when capable, escalate when necessary" principle.

However, r_i^{naive} alone often leads to early mode collapse. The key idea of TB-GRPO is to modulate rewards based on how far the current escalation ratio ρ deviates from the target band $[\eta{-}\gamma,\ \eta{+}\gamma]$:

$$\begin{aligned} \delta_{\text{esc}} &= \text{clip}(\rho - (\eta + \gamma), 0, 1)\,, \\ \delta_{\text{ans}} &= \text{clip}((\eta - \gamma) - \rho, 0, 1)\,. \end{aligned} \tag{6}$$

As Fig. 5 (right) illustrates, δ_{esc} activates only when escalation is excessive ($\rho > \eta{+}\gamma$), δ_{ans} activates only when escalation is insufficient ($\rho < \eta{-}\gamma$), and both remain zero inside the target band.

These deviations then modulate the base reward:

$$r_i = \begin{cases} (1 - \delta_{\text{esc}})\, r_i^{\text{naive}}, & e_i = 1,\ c_i = 1, \\ (1 - \delta_{\text{esc}})\, r_i^{\text{naive}} - \delta_{\text{esc}}, & e_i = 1,\ c_i = 0, \\ (1 - \delta_{\text{ans}})\, r_i^{\text{naive}}, & e_i = 0,\ c_i = 1, \\ (1 - \delta_{\text{ans}})\, r_i^{\text{naive}} - 2\delta_{\text{ans}}, & e_i = 0,\ c_i = 0. \end{cases} \tag{7}$$

The effect is automatic correction: over-escalation compresses escalation rewards, while under-escalation penalizes direct answers, steering ρ back toward the target band.

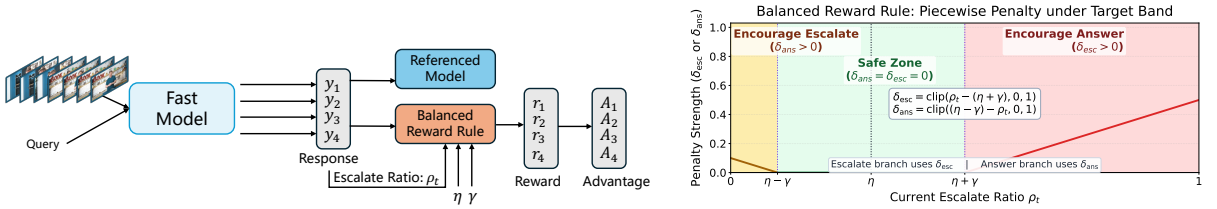


Figure 5: TB-GRPO for adaptive routing. **Left:** training pipeline where the policy samples grouped routing outputs, computes ratio-aware rewards under target-band control (η, γ), normalizes advantages, and updates with clipped GRPO plus KL regularization. **Right:** piecewise penalties versus escalation ratio ρ: when $\rho < \eta - \gamma$, non-escalation is penalized ($\delta_{\text{ans}} > 0$); when $\eta - \gamma \le \rho \le \eta + \gamma$, both penalties are inactive; when $\rho > \eta + \gamma$, escalation is penalized ($\delta_{\text{esc}} > 0$).

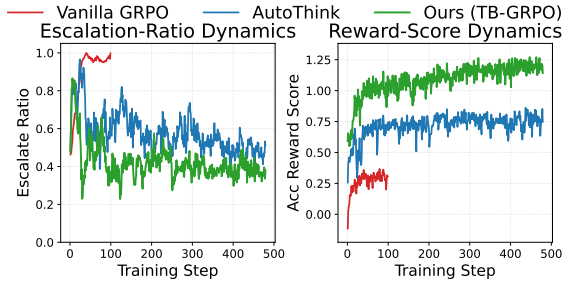


Figure 6: Training dynamics during routing optimization. Vanilla GRPO saturates toward the all-<Escalate> mode, while AutoThink and TB-GRPO avoid collapse. TB-GRPO further maintains a lower escalation ratio and converges to a higher reward level.

Finally, group-normalized advantages $A_i = (r_i - \bar{r})/(\text{std}(\{r_j\}) + \epsilon)$ are computed and optimized with a clipped policy objective:

$$\begin{aligned}\mathcal{L}_{\text{TB-GRPO}} = \mathbb{E}[&\min(w_i A_i, \\ &\text{clip}(w_i, 1-\epsilon_c, 1+\epsilon_c) A_i)] \\ &- \beta_{\text{KL}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}),\end{aligned} \tag{8}$$

where $w_i = \pi_\theta(y_i|x)/\pi_{\theta_{\text{policy}}}(y_i|x)$. The two-parameter control (η, γ) makes escalation frequency directly tunable for compute budgets.

Fig. 6 confirms this empirically. Vanilla GRPO collapses to ρ=1.0 by step 40. AutoThink (Tu et al., 2025) avoids early collapse but still settles at a high escalation ratio. TB-GRPO converges to both a lower, more stable ratio and a higher reward level—properties essential for meeting streaming latency constraints. Additional η and γ ablations appear in Sec. 5 (Table 8), and the reward-surface visualization is in Appendix D.2.

5 Experiments

5.1 Experiment Settings

Base Settings. We use Qwen2.5-VL-3B or Qwen2.5-VL-7B as fast models (Bai et al., 2025b). The candidate slow models are Qwen3-VL-4/8B-Thinking (Bai et al., 2025a), and Qwen2.5-VL-32B (Bai et al., 2025b). In our evaluations, we denote specific pipeline configurations using the format R3-Streaming-[Fast Model]|[Slow Model]. For example, R3-Streaming-3B|4B-Thinking indicates the use of the 3B model for the fast response/routing path and the 4B-Thinking model for the escalated reasoning path. For *Remember*, we adopt the DTD compression operator from TimeChat-Online (Yao et al., 2025). Unless otherwise stated, the Nearby threshold is 1.0 (no compression), the Historical threshold is 0.01, and the Nearby window is 3 frames. For policy learning, we build the *Respond* and *Reason* training sets from TimeChat-Online-139K (Yao et al., 2025). In TB-GRPO, we set the target escalation ratio to $\eta = 0.3$ and the tolerance parameter to $\gamma = 0.2$.

Benchmarks. We evaluate streaming understanding on StreamingBench (Lin et al., 2024b) and OVO-Bench (Li et al., 2025), which focus on online video QA. We also conduct long-context evaluation on MLVU (Zhou et al., 2024) and Video-MME (Fu et al., 2024).

Baselines. We compare against representative online-video methods, including TimeChat-Online (Yao et al., 2025), Streamo (Xia et al., 2025), Dispider (Qian et al., 2025), and StreamingVLM (Xu et al., 2025). We also include Qwen2.5-VL (Bai et al., 2025b) as a foundation VLM baseline and AdaReTaKe (Wang et al., 2025a) as a long-video redundancy-reduction baseline.

5.2 Streaming Benchmarks

As shown in Table 1 and Table 2, R3-Streaming achieves SOTA performance among streaming MLLMs on two mainstream streaming benchmarks, OVO-Bench (Li et al., 2025) and StreamingBench (Lin et al., 2024b), outperforming recent

Table 1: Main comparison on OVO-Bench (Li et al., 2025). We report the average score for each benchmark category and the overall average; detailed subcategory scores are provided in Appendix Table 12. For our entries, "96% ↓" denotes the visual-token reduction ratio introduced by **Remember** (Active Forgetting) before downstream reasoning. Bold indicates the best result within the Streaming MLLMs block.

Model	Real-Time Avg.	Backward Avg.	Forward Avg.	Overall Avg.
Offline Video MLLMs				
Qwen2-VL-72B (Bai et al., 2025b)	61.92	56.95	49.30	56.27
Qwen3-VL-4B-Thinking (Bai et al., 2025a)	62.93	51.56	58.74	57.74
Qwen2.5-VL-7B (Bai et al., 2025b)	64.70	44.68	41.80	50.39
LLaVA-OneVision-7B (Li et al., 2024a)	64.02	43.71	50.50	52.74
Qwen2-VL-7B (Bai et al., 2025b)	55.98	46.46	48.74	50.39
LongVU-7B (Shen et al., 2025)	57.61	35.01	47.50	46.71
Streaming MLLMs				
Flash-VStream-7B (Zhang et al., 2025a)	28.37	27.38	45.09	33.61
VideoLLM-online-8B (Chen et al., 2024)	20.79	17.73	-	-
Dispider-7B (Qian et al., 2025)	54.55	36.06	34.72	41.78
ET-Instruct-3B (Liu et al., 2024)	46.47	28.52	46.62	40.54
Streamo-3B (Xia et al., 2025)	61.51	41.76	53.72	52.33
Streamo-7B (Xia et al., 2025)	65.98	46.10	54.77	55.61
TimeChat-Online-7B (85%↓) (Yao et al., 2025)	58.60	42.00	36.40	45.60
StreamAgent-7B (Yang et al., 2025a)	61.30	41.70	45.40	49.40
FluxMem-7B (Xie et al., 2026)	-	-	-	53.40
R3-Streaming-3B|7B-Instruct (Ours, 96%↓)	65.12	44.64	52.44	54.07
R3-Streaming-3B|4B-Thinking (Ours, 96%↓)	63.58	50.23	**55.83**	56.55
R3-Streaming-7B|4B-Thinking (Ours, 96%↓)	**71.89**	**51.27**	50.60	**57.92**

streaming methods. While reducing more than 95% of visual tokens, our method also surpasses both the slow baseline Qwen3-VL-4B-Thinking and the fast baseline Qwen2.5-VL-7B. This indicates that the performance gain does not simply come from a fixed trade-off between slow and fast models. Instead, R3-Streaming uses agentic control to adaptively select a more suitable action according to the query type. The finer-grained subtask analysis in Fig. 11 further corroborates this behavior.

5.3 Long Video Understanding Benchmarks

Though designed for streaming, R3-Streaming generalizes to offline long-video tasks (Table 3). Considering the substantial differences between offline and online video settings, we reset the Historical threshold to 0.5 for offline long-video evaluation, corresponding to about 45% token drop. A more detailed hyperparameter analysis is provided in Appendix B.2 (Fig. 9) and Sec. B.2.2. This hyperparameter difference reflects the offline/online setting shift, and we use the same offline configuration across long-video benchmarks. On both benchmarks, R3-Streaming also outperforms representative long-video methods such as AdaReTaKe (Wang et al., 2025a) and VideoAgent (Wang et al., 2024a). These results confirm the robustness of our method: although R3-Streaming is designed for online/streaming scenarios, it still achieves strong performance on offline scenarios.

Table 2: Main comparison on StreamingBench (Lin et al., 2024b). We report the official "All" score in the main paper; detailed subtask scores and frame settings are provided in Appendix Table 13. Bold indicates the best result within the Streaming MLLMs block.

Model	Params	All
Human		
Human	-	91.46
Proprietary MLLMs		
GPT-4o (OpenAI, 2024)	-	73.28
Claude 3.5 Sonnet (Anthropic, 2024)	-	72.44
Offline Video MLLMs		
LLaVA-OneVision (Li et al., 2024a)	7B	71.12
Qwen2-VL (Bai et al., 2025b)	7B	69.04
Qwen3-VL-4B-Thinking (Bai et al., 2025a)	4B	73.16
Qwen2.5-VL-7B (Bai et al., 2025b)	7B	73.28
MiniCPM-V 2.6 (Yao et al., 2024)	8B	67.44
LLaVA-NeXT-Video (Zhang et al., 2024)	32B	66.96
VILA-1.5 (Lin et al., 2024a)	8B	52.32
Video-CCAM (Fei et al., 2024)	14B	53.96
Video-LLaMA2 (Cheng et al., 2024)	7B	49.52
Streaming MLLMs		
Flash-VStream (Zhang et al., 2025a)	7B	23.23
VideoLLM-online (Chen et al., 2024)	8B	35.99
Dispider (Qian et al., 2025)	7B	67.63
TimeChat-Online-7B (83%↓) (Yao et al., 2025)	7B	73.64
StreamAgent (Yang et al., 2025a)	7B	74.28
R3-Streaming-3B|7B-Instruct (Ours, 95%↓)	3B/7B	73.84
R3-Streaming-3B|4B-Thinking (Ours, 95%↓)	3B/4B	74.36
R3-Streaming-7B|4B-Thinking (Ours, 95%↓)	7B/4B	**76.36**

Table 3: Long-video understanding results on MLVU (Zhou et al., 2024) and Video-MME (Fu et al., 2024). For Video-MME, we report the overall score without subtitles. Bold indicates the best value per metric.

Model	MLVU	Video-MME Overall
Baseline (Qwen2.5-VL-3B (Bai et al., 2025b))	66.5	60.4
Qwen2.5-VL-7B (Bai et al., 2025b)	66.9	63.6
VideoAgent (Wang et al., 2024a)	57.8	56.0
TimeChat-Online-7B (Yao et al., 2025)	62.9	63.3
StreamAgent-7B (Yang et al., 2025a)	67.2	62.9
AdaReTaKe (Wang et al., 2025a)	-	63.1
R3-Streaming (Ours, 3B|4B)	**70.6**	**65.5**

5.4 Ablations

5.4.1 Base Module Ablation

We first evaluate the individual and combined effects of the two main technical components, Remember and Reason, in Table 4. Respond only targets proactive forward tasks rather than all tasks accuracy, and is therefore evaluated separately on the StreamingBench Proactive split in Sec. 5.4.3. Applying either the Remember (memory compression) or Reason (adaptive routing) module independently improves upon the Qwen2.5-VL-3B baseline (Bai et al., 2025b). Crucially, integrating both modules yields the best performance on StreamingBench (Lin et al., 2024b) and OVO-Bench (Li et al.,

Table 4: Base-module ablation on StreamingBench (Lin et al., 2024b) and OVO-Bench (Li et al., 2025). Integrating both **Remember** and **Reason** yields the best performance.

Method	Streaming-Bench ↑	OVO-Bench ↑
Baseline (Qwen2.5-VL-3B (Bai et al., 2025b))	68.6	51.4
w/ Remember	71.0	52.8
w/ Reason	73.4	55.9
w/ Both	**74.4**	**56.6**

Table 5: Remember compression comparison on Qwen2.5-VL-7B (Bai et al., 2025b). Our lightweight DTD-based variant achieves the best accuracy and highest drop ratio.

Method	StreamingBench ↑	Drop Ratio ↑
DivPrune (Alvar et al., 2025)	68.48	92.8%
VisionZip (Yang et al., 2025b)	68.32	92.8%
DTD (Yao et al., 2025)	65.16	92.7%
Ours w/ DTD	**75.90**	**95.0%**

Table 6: Respond ablation on the StreamingBench (Lin et al., 2024b) Proactive split. The dedicated readiness head substantially improves proactive output quality.

Method	Proactive Output ↑
Qwen2.5-VL-7B (Bai et al., 2025b)	0.204
Qwen2.5-VL-3B (Bai et al., 2025b)	0.216
Ours	**0.328**

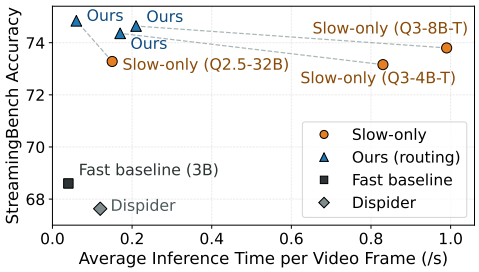
Table 7: Comparison of Reason routing policies. TB-GRPO avoids collapse and achieves the strongest performance-efficiency trade-off.

Method	Streaming-Bench ↑	Escalate Ratio ↓
SFT-only	73.16	100.0%
Vanilla GRPO (Shao et al., 2024)	73.16	100.0%
AutoThink (Tu et al., 2025)	70.00	53.2%
Ours (TB-GRPO)	**74.36**	**24.0%**

Table 8: TB-GRPO hyperparameter ablation (*Acc. / Esc. ratio*). (η, γ) controls the operating point between accuracy and escalation frequency.

$\eta \backslash \gamma$	0.1	0.2	0.3
0.2	71.1 / 15%	73.2 / 19%	74.2 / 26%
0.3	73.8 / 21%	74.4 / 24%	74.1 / 32%
0.4	74.5 / 32%	73.5 / 35%	73.7 / 41%

Figure 7: Efficiency vs Performance. Adaptive routing consistently outperforms direct slow-only inference across all tested slow models.



2025). This demonstrates that active memory control and adaptive compute routing are not isolated optimizations, but rather highly complementary components within our agentic framework.

5.4.2 Remember Compression Ablation

In Table 5, we compare R3-Streaming with prior methods that perform token compression alone. The results show that age-aware policy brings a substantial performance gain over standalone token compression. Furthermore, Fig. 8 in Appendix B.1 studies the effect of different compression operators. We find that different operators achieve similar peak performance under the age-aware policy, indicating that the gain comes more from the age-aware policy than from the specific operator design. This also demonstrates the robustness of our age-aware memory strategy. Appendix D.3.1 provides a finer-grained view of this complementarity, showing that better nearby-memory preservation simultaneously improves accuracy and reduces the escalation ratio.

5.4.3 Respond Proactive Ablation

Table 6 shows that the Respond head improves proactive output performance on StreamingBench (Lin et al., 2024b) (0.328 vs. 0.216 with the same fast backbone). This suggests that readiness control is learned by the response head instead of coming from base-model scaling.

5.4.4 Reason Ablation

In Table 7, we isolate the effect of the proposed TB-GRPO objective. Directly optimizing agentic control with Vanilla GRPO (Shao et al., 2024) or SFT-only leads to mode collapse, where all queries trigger `<Escalate>` (invoking the slow model), causing large latency as shown in Fig. 7. Also, AutoThink (Tu et al., 2025) still invokes escalation for a large fraction of queries, while the performance is suboptimal. By contrast, TB-GRPO triggers only a small number of escalations, improving benchmark performance while maintaining low latency, making it better suited to streaming understanding. Appendix D.3.2 further shows that this routing policy accurately adapts to intrinsic task difficulty.

5.5 Complexity and Cost Analysis

In Fig. 7, we analyze the complexity of R3-Streaming. Compared with the fast baseline and the prior streaming method Dispider (Qian et al., 2025), our method substantially improves benchmark performance. Compared with slow-only models, R3-Streaming greatly reduces per-frame latency. This shows that our method can preserve strong performance while remaining practical for real-time streaming scenarios.

6 Conclusion

We present R3-Streaming, a cascaded agentic framework that decomposes streaming video under-

standing into three sequential decisions, enabling adaptive coordination of memory compression, response timing, and compute routing. Stabilized by TB-GRPO, our system achieves state-of-the-art streaming performance on OVO-Bench and StreamingBench while reducing visual token usage by 95–96%, and generalizes effectively to long-video scenarios.

Limitations

While TB-GRPO successfully constrains the average escalation ratio to respect overarching compute budgets, handling worst-case latency spikes during prolonged segments of extreme information density remains an open challenge. Current streaming benchmarks primarily evaluate average metrics. We believe that addressing such localized computational peaks, where even fast models struggle, is an important next step for the streaming VLM community and calls for spike-aware benchmarks in future work.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9392–9401.

Anthropic. 2024. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`. Anthropic model announcement, June 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, and 1 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

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

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

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

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. 2025. Online video understanding: Ovbench and videochat-online. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3328–3338.

Xinqi Jin, Hanxun Yu, Bohan Yu, Kebin Liu, Jian Liu, Keda Tao, Yixuan Pei, Huan Wang, Fan Dang, Jiangchuan Liu, and Weiqiang Wang. 2025. Streamingassistant: Efficient visual token pruning for accelerating online video understanding. *arXiv preprint arXiv:2512.12560*.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167. Association for Computational Linguistics.

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

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

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. 2025. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18902–18913.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024a. Vila: On

pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26689–26699.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. 2024b. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*.

Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. 2024. E.t. bench: Towards open-ended event-level video-language understanding. *arXiv preprint arXiv:2409.18111*.

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

OpenAI. 2024. GPT-4o system card. `https://openai.com/index/gpt-4o-system-card/`. OpenAI technical report, August 2024.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

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

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *arXiv preprint arXiv:2505.10832*.

Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. 2025a. Adaretake: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5417–5432.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. 2024a. Videoagent: Long-form video understanding with large language model as agent. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yiqin Wang, Haoji Zhang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. 2024b. Hierarchical memory for long video qa. *arXiv preprint arXiv:2407.00603*. Accepted to CVPR 2024 Workshop.

Yueqian Wang, Xiaojun Meng, Yifan Wang, Huishuai Zhang, and Dongyan Zhao. 2025b. Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models. *arXiv preprint arXiv:2507.09313*.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. 2025. Streaming video instruction tuning. *arXiv preprint arXiv:2512.21334*.

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

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. 2025. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*.

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

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

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. 2025. Timechat-online: 80%

visual tokens are naturally redundant in streaming videos. In *ACM Multimedia 2025*, pages 10807–10816.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, and 4 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

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

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025b. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*.

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

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: Benchmarking multi-task long video understanding. *arXiv preprint arXiv:2406.04264*.

A Appendix Overview and Benchmark Task Taxonomy

This appendix provides supplementary analyses and full results organized by the three core modules of R3-Streaming. Appendix B consolidates all **Remember** (Active Forgetting) experiments: compression operator ablation (Sec. B.1), backbone and benchmark generalization grids (Sec. B.2), and nearby window-size analysis (Sec. B.3). Appendix C describes the **Respond** module's SFT data construction. Appendix D presents additional **Reason** (Adaptive Thinking) analysis: the TB-GRPO reward-surface visualization (Sec. D.2) and subtask-level routing behavior. Finally, Appendix E collects the full subtask-level benchmark tables. We begin by spelling out the subtask taxonomies used throughout our evaluation tables so that the abbreviated column headers remain easy to read.

OVO-Bench. OVO-Bench (Li et al., 2025) evaluates online video understanding under three causal settings: *Real-Time Visual Perception*, *Backward Tracing*, and *Forward Active Responding*. Table 9 lists the fine-grained task names. The official project page uses FTP for *Future Prediction*, while several reproduced result tables, including ours, inherit the shorthand FPD. We keep FPD in the tables for consistency with the main paper and interpret it as *Future Prediction*.

Table 9: **OVO-Bench Subtask Taxonomy.** Detailed full names for the subtask abbreviations referenced in the main paper's evaluation tables.

Group	Abbrev.	Full Name
Real-Time Visual Perception	OCR	Optical Character Recognition
Real-Time Visual Perception	ACR	Action Recognition
Real-Time Visual Perception	ATR	Attribute Recognition
Real-Time Visual Perception	STU	Spatial Understanding
Real-Time Visual Perception	FPD	Future Prediction
Real-Time Visual Perception	OJR	Object Recognition
Backward Tracing	EPM	Episodic Memory
Backward Tracing	ASI	Action Sequence Identification
Backward Tracing	HLD	Hallucination Detection
Forward Active Responding	REC	Repetition Event Count
Forward Active Responding	SSR	Sequential Steps Recognition
Forward Active Responding	CRR	Clues Reveal Responding

StreamingBench. In the main paper we report the *Real-Time Visual Understanding* split of StreamingBench (Lin et al., 2024b). Table 10 expands the ten abbreviated subtask names used in StreamingBench.

Table 10: **StreamingBench Subtask Taxonomy.** Detailed full names for the real-time visual understanding subtasks referenced in the main paper.

Abbrev.	Full Name
OP	Object Perception
CR	Causal Reasoning
CS	Clip Summarization
ATP	Attribute Perception
EU	Event Understanding
TR	Text-Rich Understanding
PR	Prospective Reasoning
SU	Spatial Understanding
ACP	Action Perception
CT	Counting

B Remember: Memory Compression Details

This section consolidates all ablation and generalization experiments related to the Remember (Active Forgetting) module.

B.1 Compression Operator Ablation

Figure 8 shows that Remember does not rely on a single compression operator. Across DivPrune (Alvar et al., 2025), VisionZip (Yang et al., 2025b), Average Pooling, and DTD (Yao et al., 2025), the strongest cells consistently appear when nearby evidence is preserved and historical memory is aggressively compressed. This supports the main-paper conclusion that the age-aware compression policy matters more than the particular operator used to instantiate it.

B.2 Compression Grids Across Backbones and Benchmarks

B.2.1 Consistency Across Model Architectures

For streaming video understanding (StreamingBench), the compression heatmaps reveal a highly consistent pattern regardless of the underlying model. Whether using a standard fast model (Qwen2.5-VL-7B) or a reasoning-heavy model (Qwen3-VL-4B-Thinking), the optimal performance invariably lies in the top-right region of the grid. Specifically, maintaining recent evidence at high fidelity (Nearby Threshold = 1.0) while aggressively compressing stale history (Historical Threshold = 0.01) yields the highest accuracy: 75.9% for the 7B model and 75.6% for the 4B-Thinking model. Moving horizontally to the left (compressing nearby frames) causes a catastrophic performance drop—for instance, dropping from

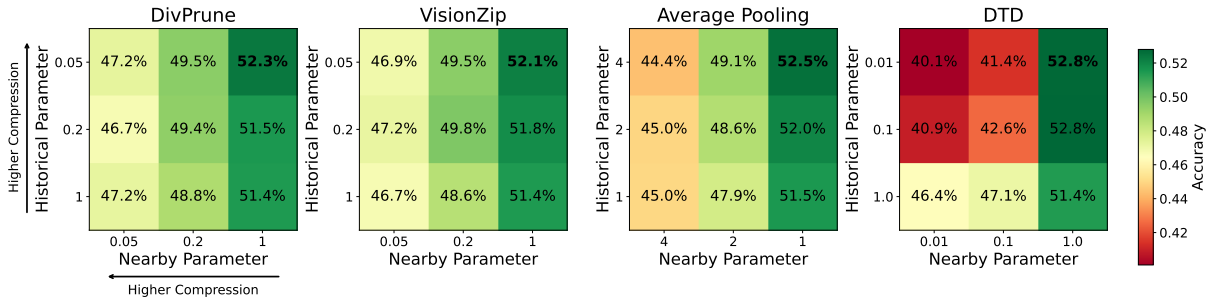


Figure 8: **Remember ablation with four compression operators on OVO-Bench (Li et al., 2025).** Each panel shows a grid search over operator-specific hyperparameters, and each cell reports overall accuracy. For Pooling, Parameter indicates the pooling kernel size. In the top-right region (aggressive historical compression with nearby evidence preserved), all operators outperform the no-compression baseline.

75.9% to 56.9% on the 7B model at Historical = 0.01. Conversely, moving vertically downward (retaining more history) provides no meaningful accuracy gain and often degrades performance by introducing noise. This definitively proves that the age-aware Active Forgetting policy is not backbone-specific; the optimal strategy is universally recent-focused.

B.2.2 Generalization to Offline Long-Video Tasks

The bottom panels of Figure 9 extend this analysis to offline long-video benchmarks (MLVU (Zhou et al., 2024) and Video-MME (Fu et al., 2024)) using the Qwen2.5-VL-3B backbone and an expanded threshold sweep $\{0.2, 0.5, 0.8, 1.0\}$. Although our method is designed for streaming, it remains robust in offline long-video settings. Because offline videos distribute useful evidence more evenly across the full clip, the best operating point shifts from extreme historical compression to a moderate historical threshold (Historical Threshold = 0.5): 68.6% on MLVU and 62.1% on Video-MME when Nearby Threshold = 1.0. This adjustment is not a video-specific hyperparameter search, but a deterministic scenario toggle: in deployment, the system knows a priori whether it is processing a continuous live stream or a pre-recorded offline file, so this binary configuration adapts R3 to the distinct evidence distributions of online and offline settings without requiring complex adaptive tuning. The added 0.2 threshold confirms this distinction: aggressive historical compression remains competitive on MLVU (68.5% at Historical = 0.2, Nearby = 1.0) but is less reliable on Video-MME (60.3%), where retaining moderate history better supports long-range dependency modeling. Taken together, these grids show that our dual-zone memory policy robustly transfers beyond streaming while allowing the historical threshold to be selected by the known deployment scenario.

B.3 Effect of Nearby Window Size

Table 11: **Effect of Nearby window size on StreamingBench performance and compression ratio.** A window of 3 frames rigorously enforces the extreme 95% token drop ratio utilized as the default in our main experiments. While expanding the window to 5–10 frames yields peak accuracy, excessive expansion (e.g., 20 or 40 frames) significantly degrades performance despite retaining more tokens. This confirms that *Active Forgetting* crucially filters out informational noise from stale history.

Nearby Window	StreamingBench Score ↑	Drop Ratio ↑
3	71.0	95%
5	72.24	92%
10	72.08	87%
20	70.40	77%
40	69.88	56%

Table 11 quantifies the performance-efficiency trade-off controlled by the Nearby window size.

B.3.1 Balancing Compression and Accuracy

The size of the Nearby window serves as a flexible tuning knob depending on deployment constraints. A strict window of 3 frames achieves an extreme 95% token drop ratio, making it highly effective for strictly constrained streaming environments. However, expanding the window to 5 or 10 frames unlocks the highest accuracy regime (peaking at 72.24) while still maintaining a highly efficient 87%-92% drop ratio. This demonstrates that our framework easily adapts to scenarios where a slight relaxation of compute budgets can be traded for peak performance.

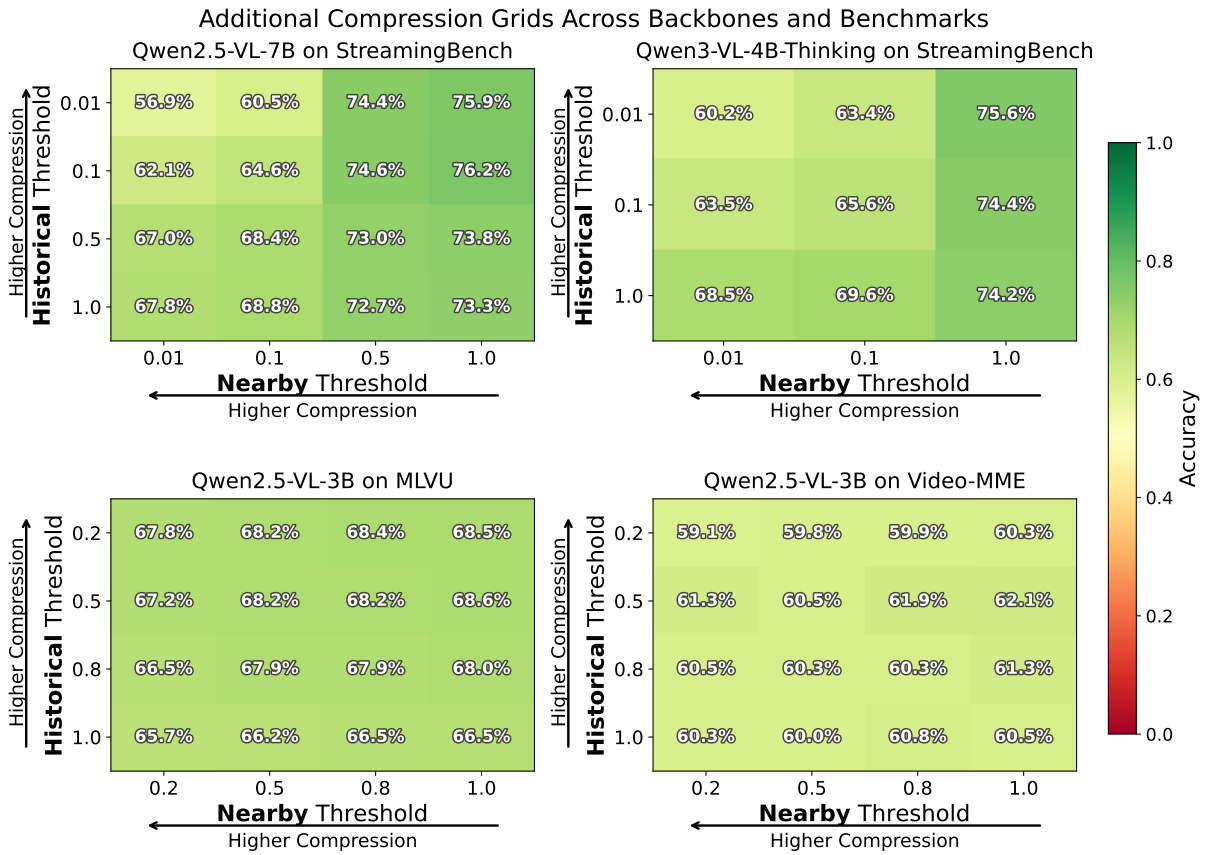


Figure 9: **Additional memory compression grids across backbones and benchmarks.** The heatmaps illustrate the effect of varying the Historical and Nearby thresholds on overall accuracy. For streaming tasks (top row), the optimal operating region consistently lies in the top-right (Historical=0.01, Nearby=1.0), validating that our recent-focused *Active Forgetting* policy is universally effective across both fast models (Qwen2.5-VL-7B) and reasoning models (Qwen3-VL-4B-Thinking). For offline long-video tasks (bottom row), we expand the sweep with threshold 0.2; the core robustness remains, but moderate historical compression is preferred because useful evidence is more evenly distributed across the video.

B.3.2 The Window Size of Active Forgetting

Most importantly, Table 11 shows that indiscriminately increasing the window size eventually harms performance. When expanded to 40 frames, the accuracy drops to 69.88, even though the model retains significantly more visual tokens (only a 56% drop ratio). This counter-intuitive degradation directly corroborates Finding 1 from the main text: preserving too much uncompressed history introduces informational noise that dilutes critical recent evidence. Thus, proper window sizing in Active Forgetting acts not merely as a latency-saving heuristic, but as an essential noise-filtering mechanism that actively improves model perception.

C Respond: Proactive Response SFT Data Construction

As introduced in the main text, the *Respond* module uses a lightweight readiness head to emit `<Routine>` when evidence is insufficient, deferring the answer until a reliable response can be grounded. We construct the training set for this proactive behavior using streaming video data from TimeChat-Online-139K (Yao et al., 2025) and COIN (Tang et al., 2019).

Decision-Boundary-Focused Sampling. Rather than randomly sampling positive and negative frames across an entire video, we focus exclusively on temporal segments where answerability strictly hinges on the arrival of a specific visual clue. For each training instance, we identify the exact clue timestamp t_c at which the question first becomes answerable. We then convert the immediate temporal neighborhood into binary readiness supervision:

- **Insufficient Evidence (Target: `<Routine>`):** The three frames immediately preceding the clue emergence, namely $\{t_c-3, t_c-2, t_c-1\}$, are explicitly labeled as unready.

- **Ready (Target: Proceed to *Reason*):** The frame at the clue timestamp and the subsequent two frames, namely $\{t_c, t_c + 1, t_c + 2\}$, are labeled as ready.

Boundary cases are clipped to the valid range of the video, and frames outside this narrow local supervision window are deliberately omitted. This hard-mining approach ensures the supervision remains heavily concentrated on the exact decision boundary. By forcing the readiness head to distinguish between the frame *just before* and *just after* the evidence appears, the model learns to avoid premature hallucination while minimizing reaction latency, satisfying the strict causal constraints of streaming video understanding. Finally, as described in the main paper, we freeze the fast VLM backbone and train only the lightweight readiness head using this targeted binary data.

D Reason: Adaptive Thinking (TB-GRPO) Details

This section presents additional analysis of the Reason module, including the TB-GRPO reward-surface visualization and subtask-level routing behavior.

D.1 SFT Data Construction

As shown in Fig. 4, we construct a portion of data for the cold start prior to TB-GRPO. Specifically for the LLM Score, we employ an LLM to compare the generated responses with the ground truth (GT) and assign scores using a 5-point rating scale. The threshold T is empirically set to 2.5.

D.2 Reward-Surface Visualization

Figure 10 shows how the controllable operating point changes with the target escalation ratio η and tolerance band γ. Increasing η shifts the preferred operating band toward higher escalation frequency, while increasing γ widens the neutral region where neither branch receives a ratio penalty. This confirms that (η, γ) gives TB-GRPO an explicit control interface over routing frequency, complementing the piecewise schematic in Fig. 5.

D.3 Subtask-Level Escalation and Accuracy Analysis

Figure 11 provides a granular view of how our dynamic routing policy (Reason) behaves across different subtasks under varying memory compression states (Remember). The heatmaps show both the escalation ratio (red, top) and accuracy (green, bottom) for a comprehensive efficiency-performance analysis.

D.3.1 Memory–Routing Synergy

The clearest global pattern emerges in the "Overall" panel. Shifting from an overly aggressive compression state (Historical=0.01, Nearby=0.01) to our optimal age-aware policy (Historical=0.01, Nearby=1.0) significantly improves accuracy from 64.72% to 73.64%. Crucially, this accuracy boost is accompanied by a substantial decrease in the escalation ratio, dropping from 37.04% to 24.04%. This demonstrates a profound synergy between our modules: when the Remember module properly preserves nearby visual evidence, the fast model becomes highly capable of generating direct answers. Consequently, the optimal operating region is not achieved by blindly querying the slow model more often, but by supplying the fast path with high-fidelity recent context, thereby naturally suppressing unnecessary computational overhead.

D.3.2 Task-Driven Routing Behaviors

The subtask panels further reveal that our routing policy accurately adapts to intrinsic task difficulty. For perception-oriented tasks such as Object Perception, Attribute Perception, and Clip Summarization, restoring the Nearby Threshold to 1.0 makes the tasks both highly accurate and highly escalation-efficient. For example, Object Perception reaches 80.76% accuracy while triggering the slow model only 20.05% of the time. The fast model confidently handles these routine queries.

By contrast, cognitively demanding tasks like Counting and Prospective Reasoning inherently require deeper logic. As shown in their respective panels, even under optimal memory retention, Counting still necessitates a high escalation rate (ranging from 48.94% to 71.81%), and Prospective Reasoning maintains a baseline escalation rate above 65%. This divergence perfectly validates the design of the Adaptive Thinking module via TB-GRPO: it dynamically allocates heavy reasoning budgets to anticipatory and complex reasoning tasks, while efficiently resolving perception-heavy queries on the fast path.

E Full Benchmark Results

This section provides the complete subtask-level results for OVO-Bench and StreamingBench, expanding the category-level summaries in the main

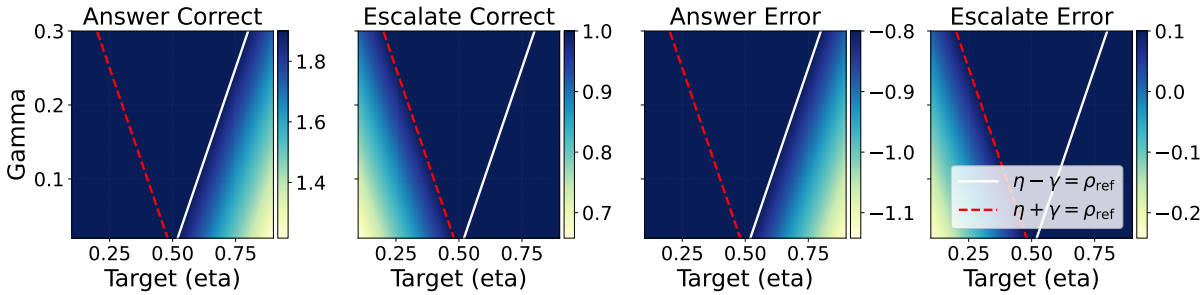


Figure 10: **Reward-surface visualization under target-band control.** With fixed ρ_{ref}=0.5 and format score = 0.1, the four panels expand the piecewise target-band rule in Fig. 5 into the *Answer/Escalate* × *Correct/Error* branches. The boundary lines indicate where answer-path or escalate-path penalties become active. This visualization serves as a sensitivity analysis for (η, γ), while the main paper focuses Fig. 6 on empirical training behavior.

paper.

E.1 Detailed OVO-Bench Results

Table 12 expands the main OVO-Bench comparison in Table 1 by reporting every subcategory score. The main paper keeps only category-level averages to make the primary comparison easier to scan.

E.2 Detailed StreamingBench Results

Table 13 expands the main StreamingBench comparison in Table 2 by reporting frame settings and all ten subtask scores. The main paper keeps only the official aggregate "All" score for compactness.

F LLM Usage

We acknowledge the use of a large language model (LLM) to assist in the preparation of this manuscript. The LLM's role was strictly limited to improving grammar and refining language. It did not contribute to any of the core research components, such as the initial ideas, experimental design, data analysis, or interpretation of the results.

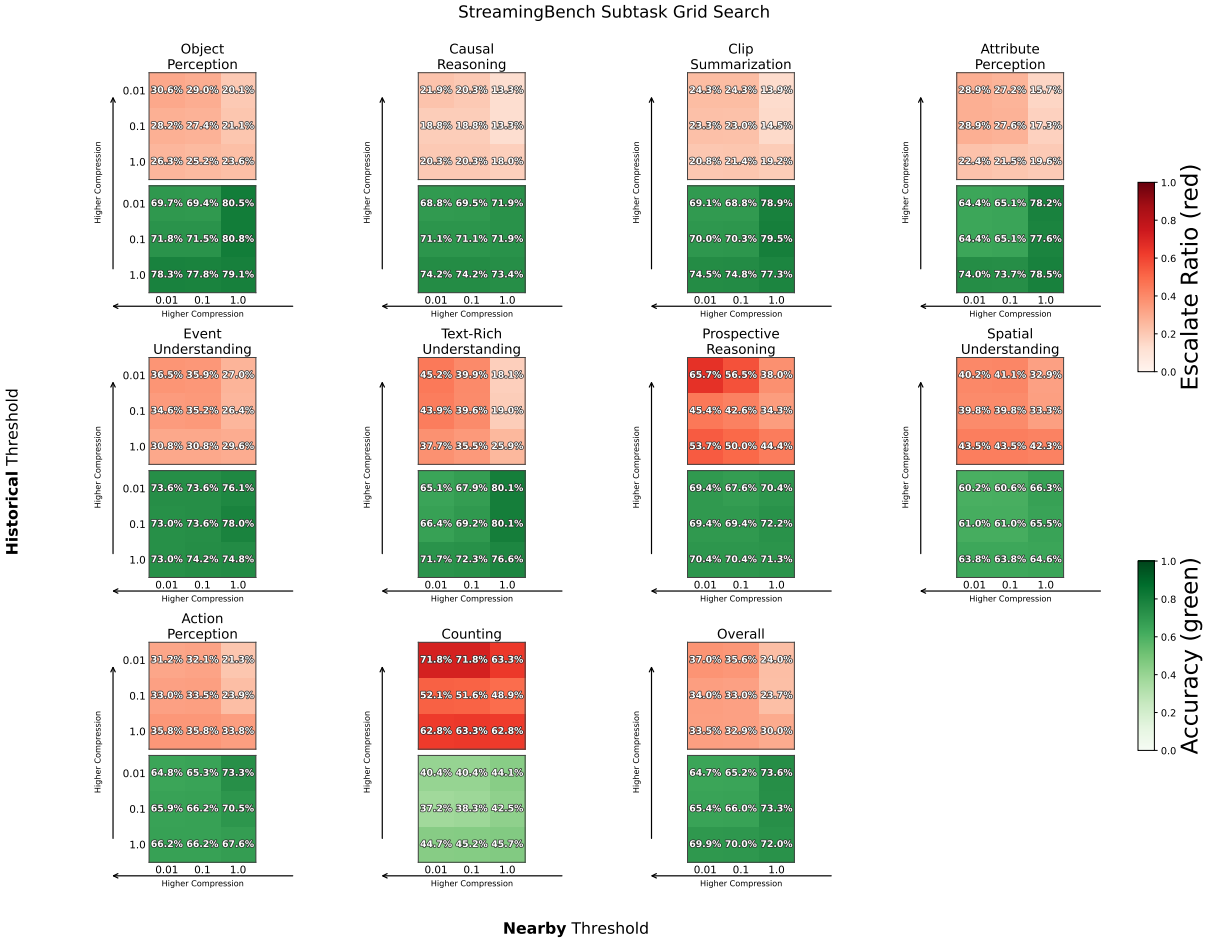


Figure 11: **StreamingBench subtask-level analysis of accuracy and escalation ratio.** Each panel displays the escalation ratio (top, red) and accuracy (bottom, green) under varying Historical and Nearby memory compression thresholds. Preserving recent evidence (Nearby=1.0) simultaneously boosts accuracy and naturally suppresses the need for slow-model escalation across most perception-oriented tasks (e.g., Object Perception). Conversely, cognitively demanding tasks like Prospective Reasoning and Counting consistently maintain higher escalation rates regardless of memory state, demonstrating the dynamic, task-driven nature of our *Adaptive Thinking* routing policy.

Table 12: **Detailed OVO-Bench comparison.** This table reports the full subcategory breakdown corresponding to Table 1. For our entries, "96% ↓" denotes the visual-token reduction ratio introduced by **Remember** (Active Forgetting) before downstream reasoning. Bold indicates the best result within the Streaming MLLMs block.

Model	#Frames	Real-Time Visual Perception							Backward Tracing				Forward Active Responding				Overall Avg.
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Overall Avg.
Offline Video MLLMs																	
Qwen2-VL-72B (Bai et al., 2025b)	64	65.77	60.55	69.83	51.69	69.31	54.35	61.92	52.53	60.81	57.53	56.95	38.83	64.07	45	49.3	56.27
Qwen3-VL-4B-Thinking (Bai et al., 2025a)	1fps	72.48	55.05	71.55	46.07	74.26	58.15	62.93	50.17	66.89	37.63	51.56	45.20	69.36	61.67	58.74	57.74
Qwen2.5-VL-7B (Bai et al., 2025b)	1fps	76.51	61.47	69.83	53.37	68.32	58.70	64.70	51.85	62.84	19.35	44.68	34.63	44.09	46.67	41.80	50.39
LLaVA-OneVision-7B (Li et al., 2024a)	64	66.44	57.8	73.28	53.37	71.29	61.96	64.02	54.21	55.41	21.51	43.71	25.64	67.09	58.75	50.5	52.74
Qwen2-VL-7B (Bai et al., 2025b)	64	60.4	50.46	56.03	47.19	66.34	55.43	55.98	47.81	35.48	56.08	46.46	31.66	65.82	48.75	48.74	50.39
LongVU-7B (Shen et al., 2025)	1fps	53.69	53.21	62.93	47.75	68.32	59.78	57.61	40.74	59.46	4.84	35.01	12.18	69.48	60.83	47.5	46.71
Streaming MLLMs																	
Flash-VStream-7B (Zhang et al., 2025a)	1fps	24.16	29.36	28.45	33.71	25.74	28.8	28.37	39.06	37.16	5.91	27.38	8.02	**67.25**	60	45.09	33.61
VideoLLM-online-8B (Chen et al., 2024)	2fps	8.05	23.85	12.07	14.04	45.54	21.2	20.79	22.22	18.8	12.18	17.73	-	-	-	-	-
Dispider-7B (Qian et al., 2025)	1fps	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.3	36.06	18.05	37.36	48.75	34.72	41.78
ET-Instruct-3B (Liu et al., 2024)	1fps	65.1	35.78	56.9	35.39	24.75	60.87	46.47	41.81	35.14	8.6	28.52	20.06	52.31	67.5	46.62	40.54
Streamo-3B (Xia et al., 2025)	1fps	78.52	52.29	67.24	44.38	55.45	**71.2**	61.51	51.18	57.43	16.67	41.76	27.94	50.72	82.5	53.72	52.33
Streamo-7B (Xia et al., 2025)	1fps	79.19	57.8	**75**	49.44	64.36	70.11	65.98	54.55	52.03	31.72	46.1	29.96	51.03	**83.33**	54.77	55.61
TimeChat-Online-7B (85%↓) (Yao et al., 2025)	1fps	69.8	48.6	64.7	44.9	68.3	55.4	58.6	53.9	62.8	9.1	42	32.5	36.5	40	36.4	45.6
StreamAgent-7B (Yang et al., 2025a)	1fps	71.2	53.2	63.6	53.9	67.3	58.7	61.3	54.8	58.1	25.8	41.7	35.9	48.4	52	45.4	49.4
FluxMem-7B (Xie et al., 2026)	1fps	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	53.4
R3-Streaming-3B|7B-Instruct (Ours, 96%↓)	1fps	83.22	58.72	63.79	55.06	66.34	63.59	65.12	57.91	64.19	11.83	44.64	34.24	66.01	57.08	52.44	54.07
R3-Streaming-3B|4B-Thinking (Ours, 96%↓)	1fps	81.21	56.88	63.79	50.56	64.36	64.67	63.58	**55.56**	**70.95**	24.19	50.23	44.41	66.01	57.08	**55.83**	56.55
R3-Streaming-7B|4B-Thinking (Ours, 96%↓)	1fps	**87.92**	**70.64**	73.28	**59.55**	**69.32**	70.65	**71.89**	51.18	65.54	**37.1**	**51.27**	**45.13**	57.09	49.58	50.60	**57.92**

Table 13: **Detailed StreamingBench comparison.** This table reports the full subtask breakdown corresponding to Table 2. Bold indicates the best result within the Streaming MLLMs block, and "All" follows the official benchmark aggregation.

Model	Params	Frames	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	All
Human													
Human	-	-	89.47	92	93.6	91.47	95.65	92.52	88	88.75	89.74	91.3	91.46
Proprietary MLLMs													
GPT-4o (OpenAI, 2024)	-	64	77.11	80.47	83.91	76.47	70.19	83.8	66.67	62.19	69.12	49.22	73.28
Claude 3.5 Sonnet (Anthropic, 2024)	-	20	80.49	77.34	82.02	81.73	72.33	75.39	61.11	61.79	69.32	43.09	72.44
Offline Video MLLMs													
LLaVA-OneVision (Li et al., 2024a)	7B	32	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12
Qwen2-VL (Bai et al., 2025b)	7B	0.2-1 fps	75.2	82.81	73.19	77.45	68.32	71.03	72.22	61.19	61.47	46.11	69.04
Qwen3-VL-4B-Thinking (Bai et al., 2025a)	4B	1 fps	77.78	74.22	76.97	80.45	73.58	78.50	74.07	63.41	67.90	57.45	73.16
Qwen2.5-VL-7B (Bai et al., 2025b)	7B	1 fps	80.22	79.69	79.18	81.41	72.96	77.26	76.85	62.20	65.06	53.19	73.28
MiniCPM-V 2.6 (Yao et al., 2024)	8B	32	71.93	71.09	77.92	75.82	64.6	65.73	70.37	56.1	62.32	53.37	67.44
LLaVA-NeXT-Video (Zhang et al., 2024)	32B	64	78.2	70.31	73.82	76.8	63.35	69.78	57.41	56.1	64.31	38.86	66.96
VILA-1.5 (Lin et al., 2024a)	8B	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32
Video-CCAM (Fei et al., 2024)	14B	96	56.4	57.81	65.3	62.75	64.6	51.4	42.59	47.97	49.58	31.61	53.96
Video-LLaMA2 (Cheng et al., 2024)	7B	32	55.86	55.47	57.41	58.17	52.8	43.61	39.81	42.68	45.61	35.23	49.52
Streaming MLLMs													
Flash-VStream (Zhang et al., 2025a)	7B	-	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.2	23.87	48.7	23.23
VideoLLM-online (Chen et al., 2024)	8B	2 fps	39.07	40.06	34.49	31.05	45.96	32.4	31.48	34.16	42.49	27.89	35.99
Dispider (Qian et al., 2025)	7B	1 fps	74.92	75.53	74.1	73.08	74.44	59.92	76.14	62.91	62.16	45.8	67.63
TimeChat-Online-7B (83%↓) (Yao et al., 2025)	7B	1 fps	79.13	**81.25**	78.86	80.77	70.44	77.26	77.78	67.07	66.19	53.72	73.64
StreamAgent (Yang et al., 2025a)	7B	1 fps	79.63	78.31	79.28	75.87	74.74	76.92	82.94	66.31	**73.69**	**55.4**	74.28
R3-Streaming-3B|7B-Instruct (Ours, 95%↓)	3B/7B	1 fps	81.03	75.78	79.18	77.88	77.36	78.82	74.07	64.23	71.31	48.40	73.84
R3-Streaming-3B|4B-Thinking (Ours, 95%↓)	3B/4B	1 fps	80.49	75.00	78.86	79.49	76.1	81.31	72.22	65.85	72.44	48.40	74.36
R3-Streaming-7B|4B-Thinking (Ours, 95%↓)	7B/4B	1 fps	**82.38**	77.34	**84.23**	**82.37**	**80.5**	**82.87**	**84.26**	**69.92**	69.03	43.62	**76.36**