DUALSTREAMBENCH: A BENCHMARK FOR DUAL-SYSTEM STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

A streaming video assistant must decide when a question warrants more computation. We propose DUALSTREAMBENCH, a benchmark that evaluates this decision through paired fast and slow answers to the same timestamped query. It covers 5,619 questions over 1,392 videos, 2,190 new human-authored open-answer questions, and audited evidence-demand labels across diverse streaming tasks. Our central finding is that paired diagnosis changes resource and routing decisions that aggregate accuracy alone cannot support. Across closed- and open-source systems, including open models up to 397B parameters, slow paths overturn correct fast answers on 3.7–14.5% of questions, including 6.7–13.1% in the six full-core configurations where they raise average accuracy. Training on paired outcomes improves allocation: on a 35B model, a rescue router beats random selection given twice as many slow calls and comes within 0.02 points of always escalating with 25% fewer. The same supervision lifts a vision-language gate from random level to 2.2 points above random across six backbones, although recent frames add no reliable gain over a matched text gate. On the same checkpoint, a longer visual window adds 6.57 points while 40.8$\times$ more thinking tokens add none, and complete-stream replay attributes 91.6% of routed service cost to memory maintenance. We believe DUALSTREAMBENCH makes beneficial escalation, transferable allocation, and state maintenance measurable targets for streaming video systems.

1 INTRODUCTION

A streaming video assistant must perceive current content, recover earlier events, localize evidence, and decide when to respond. For example, in Figure 1, identifying a shirt's color, recalling a previous action, and waiting for sugar to melt place very different demands on the same evolving stream. The first can be answered from the latest frames, while the others need history, reasoning, or patience. Under a finite computation budget, an assistant must therefore act like a dual-process system: use a fast path (System 1, S_1) and escalate only when a slow path (System 2, S_2) can help.

Video benchmarks evaluate timestamped perception, memory, and proactive interaction (Fu et al., 2025; Lin et al., 2024; Huang et al., 2025b; Niu et al., 2025). Adaptive deployment adds another requirement: *knowing when additional computation is useful*. Longer context or a stronger model can raise average accuracy while replacing some correct answers with errors. An assistant must find the queries that benefit, keep correct fast answers, and account for the cost of history.

We introduce DUALSTREAMBENCH to measure this capability. Running fast and slow configurations on the *same* timestamped query reveals whether escalation *rescues* a wrong answer, *harms* a correct one, or changes neither outcome. Our central claim is that paired diagnosis changes decisions: it identifies which resource to supply and which questions to escalate, rather than merely ranking final accuracies. The benchmark covers 5,619 timestamped questions over 1,392 videos, five operating modes, twenty task types, human-audited demand labels, and 2,190 new open-answer questions.

We evaluate eight closed and open reference systems and six open-model configurations up to 397B on all questions. Table 2 and Figure 4 expose opposing task-level returns: history can help temporal retrieval while damaging immediate perception. The paired view yields three findings:

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.

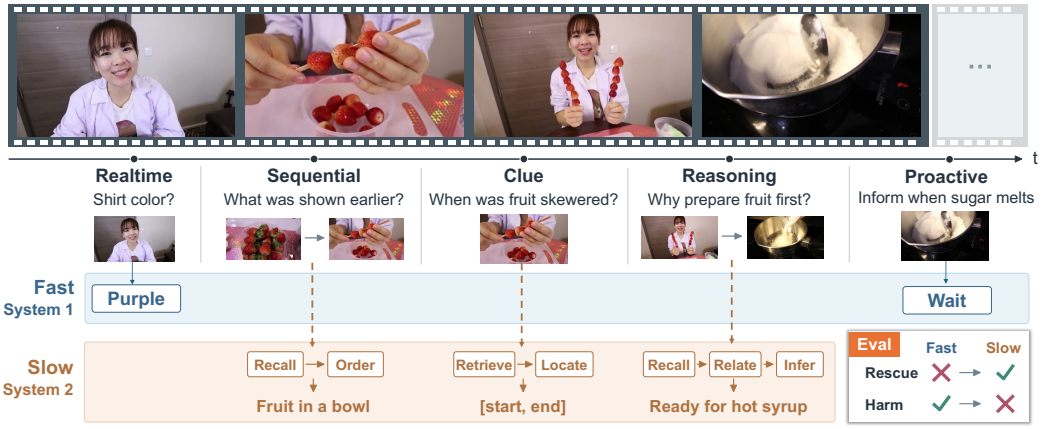


Figure 1: DUALSTREAMBENCH measures adaptive streaming competence. Timestamped questions access an observed video prefix. Paired fast and slow execution exposes the value of context, thinking, and memory through rescue and harm. The benchmark jointly measures answer quality, budgeted allocation, temporal behavior, and stream-state maintenance.

- **Escalation harms as well as helps.** In the reference tier, slow paths overturn correct fast answers on 3.68–14.47% of questions. In the full-core tier, every slow path raises average accuracy yet still harms 6.74–13.10%. Task-level effects vary sharply across systems.
- **Paired supervision improves allocation.** On a 35B checkpoint, a rescue router at a 25% budget beats random allocation at 50%. Across six backbones, model-specific targets outperform a target shared across backbones at a fixed budget. Training a vision-language gate on paired gain lifts it from random level to 2.2 points above random, while frames add little over text.
- **Resources and deployment have distinct costs.** Longer visual context helps where more deliberation does not. Routing thresholds shift across sources, and memory maintenance dominates complete-stream service time in replay.

Our contributions are: (1) a unified streaming benchmark with five operating modes, twenty task types, traceable provenance, human-audited demand labels, and a human-authored open-answer extension; (2) a paired protocol that decomposes accuracy into rescue and harm and measures temporal behavior, budgeted allocation, and complete-stream cost; and (3) a broad evaluation of closed and open systems, including models up to 397B, that reveals task-specific resource returns and persistent gaps in beneficial escalation, transfer, and state maintenance.

2 RELATED WORK

Video and streaming benchmarks. Video-MME, VideoVista, and VideoEval organize video understanding by domain, duration, and task; CG-Bench adds evidence grounding (Fu et al., 2025; Li et al., 2024b;a; Chen et al., 2025a). StreamingBench, OVBench, and OVO-Bench extend evaluation to timestamped perception, memory, and proactive tasks (Lin et al., 2024; Huang et al., 2025b; Niu et al., 2025). SVBench studies temporal dialogue, OmniMMI multimodal interaction, and StreamReady evidence readiness (Yang et al., 2025b; Wang et al., 2025c; Azad et al., 2026). DUALSTREAMBENCH adds paired resource interventions and budgeted allocation to this coverage, linking each extra computation to its answer-level benefit and harm on every query.

Fast and slow thinking in models. Dual-process cognition distinguishes fast judgments from deliberate reasoning (Kahneman, 2011). Language-model studies learn when to plan or reason longer (Saha et al., 2024; Su et al., 2025; Zhang et al., 2025), scale test-time computation (Snell et al., 2025; Muennighoff et al., 2025), and diagnose overthinking (Chen et al., 2024b). Streaming video adds resource choices over visual history and memory; DUALSTREAMBENCH measures their returns alongside deliberation, and when each helps or harms a given model.



Figure 2: Representative benchmark examples. Triangles mark queries, diamonds mark evidence, and circles mark responses. Blue and orange indicate fast/slow demand (D_1/D_2). Questions within the same stream require different observations and operations.

Adaptive inference and routing. FrameExit learns frame-level early exits, while SlowFast combines temporal resolutions (Ghodrati et al., 2021; Feichtenhofer et al., 2019). Language-model exits and cascades use confidence for selective computation (Schuster et al., 2022; Chen et al., 2023; Geifman & El-Yaniv, 2017). RouteLLM and BEST-Route learn model or budget selection; RouterEval, VL-RouterBench, and MMR-Bench evaluate routing over language and multimodal tasks (Ong et al., 2025; Ding et al., 2025; Huang et al., 2025c;a; Ma et al., 2026). DUALSTREAMBENCH evaluates allocation over evolving video prefixes together with evidence timing and state-maintenance cost.

Streaming video models and memory. Streaming dialogue, perception–response modules, and live commentary support online video interaction (Chen et al., 2024a; Qian et al., 2025; Chen et al., 2025b; Wang et al., 2025a). Memory banks, sparse tokens, and compressed or retrieved KV caches preserve history (Song et al., 2024; Zhang et al., 2024; Huang et al., 2025b; Yang et al., 2025a; Kim et al., 2025; Di et al., 2025; Xu et al., 2026). Our paired evaluation measures what history recovers, which queries need it, and the complete-stream work required to maintain it.

3 THE DUALSTREAMBENCH DATASET

3.1 TASK DESIGN AND COVERAGE

The benchmark crosses *streaming operating mode* with *semantic capability*. Five task sets determine how a query relates to the observed stream (Figure 2). **Realtime** tests perception of current or recent content. **Sequential** requires recovering or accumulating earlier observations. **Clue** pairs a question with supporting temporal intervals. **Reasoning** tests relations, verification, and prediction from visible evidence. **Proactive** tests whether to answer at successive timestamps and how accumulated event counts change. Seven categories and twenty task types further specify the operation, from object recognition to temporal localization and causal inference (Appendix A).

Table 1: Task-set × task-category composition of DualStreamBench. The bottom rows show the adjudicated demand-label split within each task set.

Category \ Task set	Realtime	Sequential	Clue	Reasoning	Proactive	Total
Visual Perception	442	90	183	0	0	715
Action & Event Recognition	432	0	276	0	698	1,406
Temporal Memory	0	783	0	0	629	1,412
Temporal Localization	0	491	60	0	0	551
Hallucination Verification	0	333	98	346	0	777
Causal & Reasoning	0	0	202	13	0	215
Predictive Reasoning	0	116	0	187	240	543
Total	**874**	**1,813**	**819**	**546**	**1,567**	**5,619**
D_1**-labeled**	839	397	1	322	582	2,141
D_2**-labeled**	35	1,416	817	224	985	3,477
Uncertain	0	0	1	0	0	1

Table 1 crosses these axes: a shared capability can require different temporal access and inference. Immediate object recognition and recovery of an earlier object state thus fall in different modes.

3.2 Construction and Annotation

Source integration. We integrate StreamingBench, OVBench, CG-Bench, and OVO-Bench, preserving source identities while standardizing formats, timestamps, semantic tasks, and demand annotations. Each question has a video, query, timestamp, and reference answer. Proactive questions occur at successive annotated times and require an answer/silence decision or cumulative count for the observed prefix. A shared instruction can thus yield different answers over time.

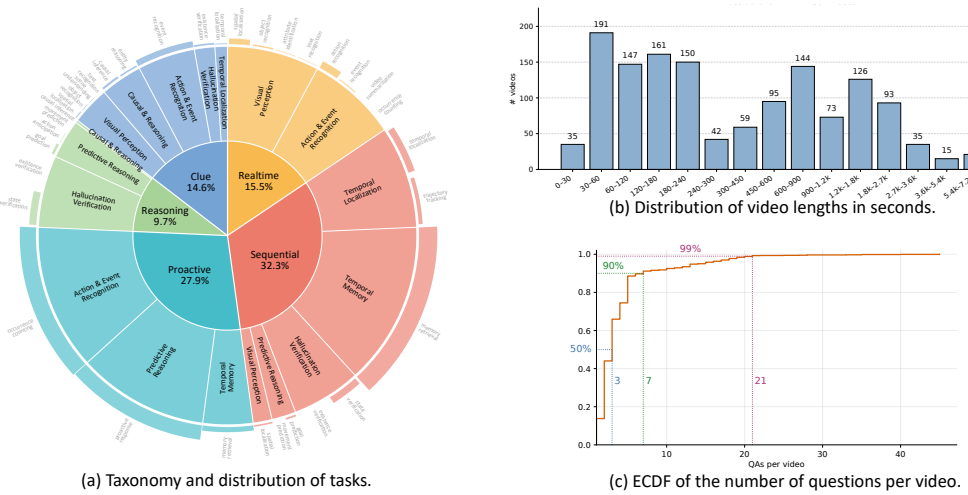
Demand annotation. Each question is manually assigned a demand label describing the evidence it requires, independent of any model. Questions answerable by immediate perception are labeled D_1; questions requiring multi-observation composition, historical retrieval, ordered recall, or inference beyond direct perception are labeled D_2. A blind audit by three new annotators covers 800 annotation groups with 1,198 questions (Fleiss' $\kappa = 0.888$). Adjudication resolves 799 groups and corrects 71 labels; one temporally ambiguous question remains uncertain (Appendix A.1). The resulting labels differ from task-set defaults on 31.0% of the 5,618 labeled questions, including 59.0% of the reasoning set, so a question's operating mode alone does not determine its demand.

Open-answer extension. Human annotators write 2,190 new questions and reference answers on 960 existing videos, covering all seven semantic categories and both demand levels (930 D_1, 1,260 D_2). Without candidate options, a model must recover and state the answer itself, which removes option-elimination shortcuts. A shared semantic rubric and two independent model judges, calibrated against human scoring, support cross-model evaluation (Appendix G).

3.3 Dataset Statistics

The core contains 5,619 timestamped questions over 1,392 videos (Table 1): 4,052 multiple-choice questions (72.1%) and 1,567 proactive questions (27.9%). Proactive questions comprise 869 answer/silence decisions and 698 counts. The open-answer extension adds 2,190 separately evaluated questions. Every question carries its query timestamp and reference answer, and video identities are preserved so that data splits and bootstrap resamples keep all questions from a video together.

Figure 3 summarizes task coverage, video duration, and query density. The demand split is 2,141 D_1, 3,477 D_2, and one uncertain question. Videos range from approximately fifteen seconds to three hours and contain a median of three questions, with 90th and 99th percentiles of seven and twenty-one. Video-level partitioning and paired video bootstrap retain these sparse and dense interactions together.



(a) Taxonomy and distribution of tasks.

(b) Distribution of video lengths in seconds.

(c) ECDF of the number of questions per video.

Figure 3: Dataset statistics. (a) Five operating modes, seven semantic categories, and twenty task types. (b) Source-recorded video durations. (c) Timestamped questions per video, shown as an empirical cumulative distribution. The median, 90th, and 99th percentile counts are 3, 7, and 21.

4 Evaluation Protocol

The protocol addresses four linked questions: (i) how accurately a model answers, (ii) which resources improve its answers, (iii) whether a policy selects the escalations that help, and (iv) what the complete stream costs over time. Reference configurations instantiate these measurements for existing open and closed backbones and a range of routing baselines.

At query time t_i, the assistant observes $x_i = (V_{\leq t_i}, q_i, h_{<i})$, containing the video prefix, question, and available interaction history. A resource action $a = (m, w, b, \mu)$ specifies a model m, context window w, generation budget b, and memory access μ. It produces an answer $\hat{y}_i(a)$, correctness $c_i(a)$, and cost $C_i(a)$. A policy π selects an action using information available at the decision time. The allocation problem is

$$\max_{\pi} \ \mathbb{E}[c_i(\pi(x_i))] \quad \text{subject to} \quad \mathbb{E}[C_i(\pi(x_i))] \leq B. \tag{1}$$

Different choices of C express an invocation, token, or wall-clock budget. We report each cost in its own unit. Two-path experiments instantiate this action space with the fast and slow configurations; single-checkpoint experiments vary inference resources while keeping model weights fixed.

Paired outcomes. Running both configurations on each question places it in one of four cells: both correct, both wrong, fixed by the slow path (*rescue*), or broken by the slow path (*harm*). For binary correctness $c_{i,\text{fast}}, c_{i,\text{slow}}$, intrinsic rescue and harm are

$$R = \mathbb{P}(c_{\text{fast}} = 0, c_{\text{slow}} = 1), \qquad H = \mathbb{P}(c_{\text{fast}} = 1, c_{\text{slow}} = 0). \tag{2}$$

They satisfy $\text{Acc}_{\text{slow}} - \text{Acc}_{\text{fast}} = R - H$ and $\text{Acc}_{\text{oracle}} = \text{Acc}_{\text{fast}} + R$, where the answer oracle selects a correct pathway whenever one exists. Two systems with equal accuracy gains can thus differ greatly in how many answers they break. For a deployed router, rescue and harm count only escalated questions, $\pi(x) = S_2$, yielding $\text{Acc}_{\pi} - \text{Acc}_{\text{fast}} = R_{\pi} - H_{\pi}$. Together, the identities separate backbone capability (Acc_{slow}), available complementarity (R), and the share of that complementarity a routing policy realizes (R_{π} relative to R) at a given budget.

Budget and information access. At invocation budget ρ, policies may escalate a fraction ρ of questions. We compare learned policies with random selection at the same rate and with a budget-matched answer oracle. A question-only pre-router decides before either model runs. A cascade router can use the fast answer or confidence and therefore includes fast inference in its cost. Offline budget curves rank held-out queries by routing score; online operating points use thresholds selected on validation videos. This separates ranking quality from the calibration a live stream needs.

Temporal state and output quality. Video and memory inputs are bounded by the query timestamp. Temporal grounding is scored by intersection-over-union with the reference evidence interval; counting uses mean absolute error. Proactive response is scored by Trigger-F1 *together with* false-alarm rate, because Trigger-F1 alone rewards a system that always answers (Appendix H). Full-video replay maintains visual memory independently of answer selection and records query and memory-update service separately over each complete observed prefix.

5 EXPERIMENTS

Experimental setup. The evaluation has two tiers. The reference tier compares four closed and four open systems on a shared set of 4,915 non-counting questions covered by all reference runs (Table 2), so that cross-system comparisons use identical questions. Its Routed column uses a question-only MiniLM–MLP that selects the slow path on the same 2,324 questions (47.28%) for every system. The full-core tier evaluates six open-model configurations from 7B to 397B (Bai et al., 2025; Wang et al., 2025b; Qwen Team, 2026) on all 5,619 questions, including the 704 occurrence-counting questions outside the API reference protocol: fast uses five recent seconds, long-direct uses fifty, and memory-enabled slow also retrieves earlier observations. Qwen2.5-VL-7B appears in both tiers: the full-core tier reruns it under the controlled protocol on every question, so its two rows are not directly comparable. Table 2 and Figure 4 report intrinsic fast-to-slow rescue and harm; deployed variants compare fast with the routed output (Appendix C). Seventeen open models also answer the 2,190 open-answer questions under both paths, scored by two calibrated model judges.

Within the full-core tier, router comparisons use three video-disjoint partitions and fixed fast/long-direct answers, with memory off (Table 3); reference Routed scores instead describe the memory-enabled pipeline. Question-only MiniLM routers (Wang et al., 2020) learn a gain score shared across backbones (shared gain), a model-specific gain, or model-specific rescue; hashed rescue adds fast confidence, and Confidence uses that signal alone. Qwen3-VL-2B-Instruct gates read the question (Text-2B) or add five recent frames (Visual-2B), either zero-shot (frozen) or with a small per-backbone head trained on paired gain (learned). Validation videos set thresholds, and 5,000 paired video resamples quantify uncertainty at the video level.

5.1 MAIN RESULTS

Closed and open systems have distinct pathway profiles. Table 2 compares pathway accuracy and complementarity in both tiers. Gemini-3-flash leads the reference systems at 67.3% routed accuracy and 71.7% oracle accuracy; InternVL3.5-8B reaches 57.4%, above GPT-4o's 56.0%. Routing improves on the fast path by 0.37–7.02 points (Table 11). Although the reference router shares one question-only score across all systems, it keeps Gemini-3-flash within 0.68 points of always-slow (67.30 vs. 67.98%) and GPT-5.2 within 1.28 (59.33 vs. 60.61%) while escalating fewer than half of the questions; Section 5.2 shows that model-specific targets improve on such shared scores. Forced slow execution helps GPT-4o, GPT-5.2, Gemini-3-flash, and Qwen2.5-VL-7B but hurts the other four reference systems. All six full-core configurations gain from memory-enabled slow execution, by 2.51–12.36 points: stronger models still show model-dependent resource returns.

Escalation harms as well as helps. All six backbones improve on average with memory-enabled slow inference, yet the slow path replaces a correct fast answer with an error on 6.74–13.10% of questions (Table 2). For Qwen3.5-397B FP8, rescue and harm nearly cancel (14.84% vs. 12.33%), and its answer oracle lies 12.3 points above its slow path. The memory-enabled path combines a longer window, retrieval, and model-specific thinking (Table 10); Section 5.3 isolates context and thinking. Harm persists even with a far stronger slow model: pairing Qwen2.5-VL-7B fast answers with the 397B slow model gains 17.30 points but still harms 10.22% of questions (Table 14).

Slow execution helps history-dependent tasks but can damage immediate decisions. Figure 4 separates the opposing transitions behind each task's net change. On realtime perception, MiniCPM-2.6-8B rescues 6.2% of questions but harms 21.4%, losing 15.2 points; InternVL3.5-8B loses 12.7 points. In contrast, Qwen2.5-VL-7B gains 10.3 points on clue questions and 14.3 on proactive tasks, where 21.3% rescue outweighs 7.0% harm. On reasoning, five of the seven displayed systems decline, while GPT-5.2 and Gemini-3-flash improve. Gemini-3-flash gains across all five task sets. Figure 6 (Appendix C) reports the full per-task accuracy profiles.

Table 2: Headline results in two evaluation tiers. Scores are percentages and Net is in percentage points. Reference systems share 4,915 non-counting questions; the full-core tier uses all 5,619. Rescue/Harm are intrinsic fast-to-slow transitions in every block, as in Figure 4; Net=Slow−Fast. Reference Routed uses a question-only MiniLM–MLP at 47.28% slow calls. Deployed rescue/harm compare fast with Routed (Appendix C). Qwen2.5-VL-7B is independently evaluated under each tier's coverage and execution protocol. Bold: best accuracy or lowest harm in each block.

Model	Size	Fast	Slow	Routed	Oracle	Rescue	Harm	Net
Closed-Source Models — reference tier, 4,915 questions								
Claude-Sonnet-4.5	–	44.50	43.23	48.30	53.90	9.40	10.66	−1.26
GPT-4o	–	51.39	55.12	55.99	61.49	10.09	6.37	+3.72
GPT-5.2	–	56.64	60.61	59.33	66.06	9.42	5.45	+3.97
Gemini-3-flash	–	**62.85**	**67.98**	**67.30**	**71.66**	8.81	**3.68**	+5.13
Open-Source Models — reference tier, 4,915 questions								
MiniCPM-2.6	8B	49.93	45.11	51.25	59.57	9.64	14.47	−4.82
Qwen2.5-VL	7B	40.85	46.84	47.87	60.90	20.04	14.06	+5.98
InternVL3.5	8B	**54.77**	**52.70**	**57.38**	**63.60**	8.83	10.91	−2.08
LLaVA-NeXT-Video	7B	40.94	39.74	41.30	46.94	6.00	**7.20**	−1.20
Open-Source Models — full-core tier, 5,619 questions								
Qwen2.5-VL	7B	38.99	42.41	–	53.19	14.20	10.78	+3.42
InternVL3.5	38B	45.19	47.84	–	60.31	15.13	12.48	+2.65
Qwen3.5	35B-A3B	49.17	54.69	–	67.79	18.62	13.10	+5.52
Qwen2.5-VL	72B	41.64	54.00	–	60.74	19.10	**6.74**	+12.36
Qwen3.5	122B-A10B	47.87	55.90	–	67.81	19.93	11.91	+8.03
Qwen3.5 (FP8)	397B-A17B	**53.78**	**56.29**	–	**68.62**	14.84	12.33	+2.51

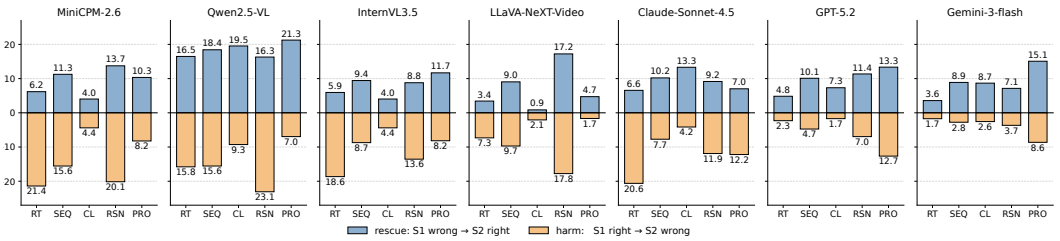


Figure 4: Intrinsic fast-to-slow rescue and harm on the 4,915 non-counting reference questions for seven systems (all except GPT-4o). Blue: wrong-to-correct; orange: correct-to-wrong, as percentages of each task set. Rescue minus harm equals the slow-minus-fast accuracy change. RT/SEQ/CL/RSN/PRO: realtime, sequential, clue, reasoning, proactive.

Open answers reveal a greater dependence on history. Across seventeen models, slow-path gains on the 2,190 human-authored questions range from 6.39 to 31.51 points. Across all models, fast/slow accuracy is 21.97/37.98%, with 18.18% rescue and 2.18% harm. Gains are larger on D_2 than D_1 questions (19.93 versus 10.69 points), linking demand labels to the value of history in answer generation. Appendix G gives every model, both judges, and paired intervals.

5.2 Routing

Similar aggregate accuracy can hide different escalation risks. In the full-core tier, Qwen3.5-35B and Qwen2.5-VL-72B have similar slow accuracy (54.69/54.00%) but harm rates of 13.10/6.74% (Table 2). The 35B model offers more oracle headroom, yet exploiting it requires protecting twice as many correct fast answers from being overturned by escalation.

Model-specific targets improve allocation at fixed endpoints. On Qwen3.5-35B at 25% escalation, shared gain scores 50.15% against 50.28% for random. With the same MiniLM question features, model-specific gain and rescue reach 51.70% and 51.49%; hashed rescue adds fast confidence and reaches 53.32%. This exceeds random at twice the budget (52.56% at 50%). At 75%, hashed rescue reaches 57.10%, within 0.02 points of always escalating, using a quarter fewer slow calls. Across all

Table 3: Router choice at 25% slow calls: accuracy (%) over three video splits, 5-second fast vs. 50-second direct, for the six full-core backbones. Frozen 2B gates score context need zero-shot; learned gates train a per-backbone head on paired gain. †: needs fast inference. Bold: best deployable; top rows lie within 95% paired intervals (Appendix E).

Router	7B	38B	35B	72B	122B	397B	Mean
Random	40.60	45.77	50.28	43.85	50.27	55.67	47.74
MiniLM shared gain	41.00	47.92	50.15	44.06	50.87	57.44	48.57
MiniLM model-specific gain	42.22	**48.05**	51.70	46.17	52.20	57.82	49.69
MiniLM model-specific rescue	41.99	47.30	51.49	46.18	51.22	57.09	49.21
Text-2B, frozen	40.82	46.08	50.45	44.50	50.91	56.79	48.26
Visual-2B, frozen	40.52	45.92	50.27	44.03	50.45	55.06	47.71
Text-2B, learned	42.80	47.38	52.10	46.61	**52.39**	58.19	49.91
Visual-2B, learned	**43.16**	47.73	51.85	**46.70**	52.02	**58.32**	**49.96**
Hashed rescue†	41.75	47.26	**53.32**	46.05	51.93	57.38	49.62
Confidence†	41.08	46.64	52.20	43.03	51.26	56.88	48.51
Oracle	54.97	59.33	66.73	59.97	66.75	68.99	62.79

six backbones, model-specific gain exceeds shared gain and hashed rescue exceeds matched random by 1.15–3.04 points (Table 3). The 35B router captures 18.5% of the available random-to-oracle improvement. Figure 5 shows the six-model gain over random; Appendix D gives full curves.

Paired supervision, not visual input, closes the gap. Used zero-shot, frozen Visual-2B, frozen Text-2B, and random reach 47.71%, 48.26%, and 47.74% (Table 3); visual minus random is -0.03 points, with a paired interval spanning zero. Training a per-backbone head on paired gain over the same frozen encoder changes this. Learned Visual-2B reaches 49.96%, $+2.22$ points over random (95% CI $[1.79, 2.63]$) and $+2.26$ over its frozen counterpart, raising rescue from 4.14% to 5.80% while lowering harm from 2.44% to 1.84%. It beats random on all six backbones and keeps its gain under a frozen validation threshold (49.99% at 25.00% slow calls). Frames, however, add little: the matched learned text gate reaches 49.91%, and visual minus text is $+0.05$ points ($[-0.40, 0.47]$), ranging from -0.76 to $+0.65$ across splits. What makes these routers useful is supervision from paired outcomes, which aggregate accuracy cannot provide (Appendix E).

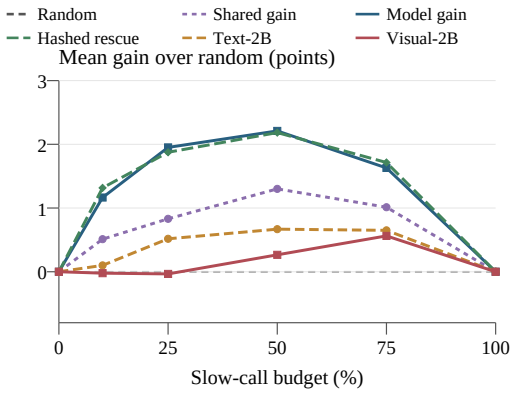


Figure 5: Offline gain over rate-matched random, averaged over six backbones and three video splits. Answers are fixed across policies.

5.3 Resource Returns

Context and thinking have different returns. On Qwen3.5-35B with memory off, extending the visible window from five to fifty seconds raises accuracy from 49.17% to 55.74%. Thinking at that same context reaches 55.51% with a 4,096-token cap, while mean generated tokens rise from 49.8 to 2,032. Thus 40.8× more generation does not reproduce the 6.57-point gain from visual history. Thinking raises 122B accuracy from 54.71% to 56.10% but lowers 397B FP8 from 59.30% to 57.13% (Figure 9). Appendix D reports caps and completion rates; Table 34 isolates answer-format effects.

Memory returns depend on the answering model. In the 1,323-question retrieval ablation, retrieval helps existence verification by 4.1 points for Gemini-3-flash but hurts it by 10.2 for InternVL3.5-8B. Under adjudicated demand labels, the memory-path gain on D_2 relative to D_1 is $+19.87$ points for Qwen2.5-VL-72B and -14.63 for Qwen3.5-122B. Recording evidence demand and realized return separately exposes how effectively each model uses the available history (Table 8).

5.4 TRANSFER AND COST

Allocation rules transfer worse than resource gains. The 35B shared-gain router changes accuracy by -0.30 points against matched random on a held-out source, with escalation from 1.6% to 56.2% against a 25% target. Frozen visual gating shifts to 16.23–50.53% across sources and remains within 0.50 points of matched random. Text-2B has positive excess accuracy on OVO-Bench ($+2.67$ points) and OVBench ($+0.43$), with paired intervals above zero (Table 21). On 7,734 SVBench questions, longer context improves 35B/InternVL3.5-38B accuracy by 2.07/2.40 points, while the shared-gain router gains only 0.03/0.20 points over matched random and escalates 4.37–16.49%. Resource gains transfer across datasets; operating thresholds must transfer too (Appendix H).

Temporal metrics reorder models and expose response control. Temporal-localization accuracy reverses 7 of 15 model-pair orderings among the six backbones, and clue-interval IoU reverses 12. Gemini-3-flash reaches Trigger-F1 0.788 with a 0.166 false-alarm rate; always answering gives similar F1 (0.790) but a false-alarm rate of 1.000. Joint metrics reveal both evidence localization and control over whether and when to answer (Appendix H).

Maintaining stream state dominates complete-stream cost. A full-stream replay executes 1,654 questions on 326 held-out videos while updating visual memory every sixty seconds, recording query and memory-update service separately (Table 29). Relative to always-memory inference, the router cuts slow calls by 86.82% and query service by 61.00%. Total service, however, falls by only 12.74%, from 9.63 to 8.40 seconds per query, because memory maintenance accounts for 91.59% of routed service time. With accuracy nearly flat (53.53–54.15%; Appendix F), state maintenance is a separate allocation target from deciding how to answer each query.

5.5 VALIDITY

Demand labels are reliable. The blind three-annotator audit (Section 3) reaches Fleiss' $\kappa = 0.888$ over 800 annotation groups, and 92.75% of groups are unanimous before adjudication. The demand-conditioned memory-gain contrasts keep their signs for every model under the original labels, the adjudicated labels, and either assignment of the ambiguous question (Appendix A.1).

Automated open-answer scoring agrees with humans and is conservative. Two annotators blindly score 6,000 outputs, i.e., 3,000 fast/slow pairs across all seventeen models, with inter-rater $\kappa = 0.809$. Against the adjudicated human reference, the InternVL3.5-38B and Qwen2.5-VL-72B judges agree on 91.00% and 90.67% of outputs. Their errors are mostly false rejections, yet they preserve the pathway effect: humans measure a 14.67-point slow-path gain (33.33% to 48.00%), and both judges find positive gains for every model on the full open set (Appendix G.1).

Answers depend on the video, and more so without options. Each of the 4,052 multiple-choice questions is rewritten as a self-contained query and answered with and without options from text only, five seconds, or fifty seconds of video (Table 34). Text-only multiple-choice accuracy is 40–42%, well above chance, but falls to 12–14% without options; fifty-second open-answer accuracy exceeds text-only by 16.83 and 13.30 points for Qwen3.5-35B and Qwen2.5-VL-72B, and open-answer context gains exceed multiple-choice gains by 2.81 and 2.02 points (Appendix G.2). The open-answer format is thus the more visually grounded of the two tests.

6 LIMITATIONS AND CONCLUSION

Limitations. Public core questions may overlap with pretraining; the open-answer extension is new. Memory-enabled slow inference bundles resources, and many thinking outputs reach their token cap. Routing covers six backbones and three splits; visual gates use one frozen 2B encoder with at most a small learned head, so end-to-end visual routers may still extract more from frames. Model judges understate absolute accuracy, and replay covers one backbone and memory design.

Conclusion. DUALSTREAMBENCH makes paired diagnosis actionable: it reveals resource returns and enables routing that improves accuracy at a fixed budget. Persistent harm, shifting operating thresholds, and dominant memory-maintenance cost show why streaming systems must optimize beneficial escalation alongside final accuracy, and why routers should learn from paired outcomes.

Reproducibility Statement

The core benchmark uses traceable source identities, source grouping IDs, and timestamped question IDs. Appendix A describes the schema and taxonomy, Appendix B specifies paired execution and routing, and Appendix G reports all open-answer model results. Model precision, context, memory, and evaluation denominators accompany each experiment. The evaluation artifacts record split manifests, per-question predictions, grader configurations, replay events, and the CPU aggregation procedures used for the reported tables.

Ethics Statement

The benchmark builds on public video resources and human-authored annotations. Source identities and usage terms are preserved, and video access follows the original releases. Streaming assistants have applications in accessibility and interactive assistance; deployment should respect the privacy and consent of people appearing in video.

AI Use Statement

Generative AI assisted manuscript restructuring, editing, and preparation of the evaluation analysis. The authors take responsibility for the final text, analysis, and released artifacts.

References

Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. StreamReady: Learning what to answer and when in long streaming videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40494–40504, 2026. URL https://arxiv.org/abs/2603.08620.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. CG-Bench: Clue-grounded question answering benchmark for long video understanding. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/70fa5df8e3300dc30bf19bee44a56155-Abstract-Conference.html.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a. URL https://arxiv.org/abs/2406.11816.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. LiveCC: Learning video LLM with streaming speech transcription at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b. URL https://arxiv.org/abs/2504.16030.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance, 2023. URL https://arxiv.org/abs/2305.05176.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? On the overthinking of o1-like LLMs, 2024b. URL https://arxiv.org/abs/2412.21187.

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

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. BEST-Route: Adaptive LLM routing with test-time optimal compute. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13870–13884. PMLR, 2025. URL https://proceedings.mlr.press/v267/ding25d.html.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6202–6211, October 2019. doi: 10.1109/ICCV.2019.00630. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Feichtenhofer_SlowFast_Networks_for_Video_Recognition_ICCV_2019_paper.html.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, June 2025. doi: 10.1109/CVPR52734.2025.02245. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1705.08500.

Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. FrameExit: Conditional early exiting for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15608–15618, June 2021. doi: 10.1109/CVPR46437.2021.01535. URL https://openaccess.thecvf.com/content/CVPR2021/html/Ghodrati_FrameExit_Conditional_Early_Exiting_for_Efficient_Video_Recognition_CVPR_2021_paper.html.

Zhehao Huang, Baijiong Lin, Jingyuan Zhang, Jingying Wang, Yuhang Liu, Ning Lu, Tao Li, and Xiaolin Huang. VL-RouterBench: A benchmark for vision-language model routing, 2025a. URL https://arxiv.org/abs/2512.23562.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: OVBench and VideoChat-Online. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3328–3338, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Huang_Online_Video_Understanding_OVBench_and_VideoChat-Online_CVPR_2025_paper.html.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 3860–3887, Suzhou, China, November 2025c. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.208. URL https://aclanthology.org/2025.findings-emnlp.208/.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot-V: Memory-constrained KV cache compression for streaming video understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=hFxOZjHyTg.

Xinhao Li, Zhenpeng Huang, Jing Wang, Kunchang Li, and Limin Wang. VideoEval: Comprehensive benchmark suite for low-cost evaluation of video foundation model, 2024a. URL https://arxiv.org/abs/2407.06491.

Yunxin Li, Xinyu Chen, Baotian Hu, Longyue Wang, Haoyuan Shi, and Min Zhang. VideoVista: A versatile benchmark for video understanding and reasoning, 2024b. URL https://arxiv.org/abs/2406.11303.

Junming Lin, Zheng Fang, Chi Chen, Zihao Wan, Fuwen Luo, Peng Li, Yang Liu, and Maosong Sun. StreamingBench: Assessing the gap for MLLMs to achieve streaming video understanding, 2024. URL https://arxiv.org/abs/2411.03628.

Haoxuan Ma, Guannan Lai, and Han-Jia Ye. MMR-Bench: A comprehensive benchmark for multimodal LLM routing, 2026. URL https://arxiv.org/abs/2601.17814.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How far is your Video-LLMs from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Niu_OVO-Bench_How_Far_is_Your_Video-LLMs_from_Real-World_Online_Video_CVPR_2025_paper.html.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2501.03218.

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

Swarnadeep Saha, Archiki Prasad, Justin Chih-Yao Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. System-1.x: Learning to balance fast and slow planning with language models, 2024. URL https://arxiv.org/abs/2407.14414.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2207.07061.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.03314.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. MovieChat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2307.16449.

DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, and Qinqing Zheng. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.09918.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. StreamBridge: Turning your offline video large language model into a proactive streaming assistant. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-4406. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/bf6939f9058a391c47014731b2486e2a-Abstract-Conference.html.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b. URL https://arxiv.org/abs/2508.18265.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. OmniMMI: A comprehensive multi-modal interaction benchmark in streaming video contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18925–18935, June 2025c. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_OmniMMI_A_Comprehensive_Multi-modal_Interaction_Benchmark_in_Streaming_Video_Contexts_CVPR_2025_paper.html.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-time understanding for infinite video streams. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6445dd88ebb9a6a3afa0b126ad87fe41-Abstract-Conference.html.

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

Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. SVBench: A benchmark with temporal multi-turn dialogues for streaming video understanding. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/693ce820ea7e0f0c70a2a833cbdf7ccc-Abstract-Conference.html.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-VStream: Memory-based real-time understanding for long video streams, 2024. URL https://arxiv.org/abs/2406.08085.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think, 2025. URL https://arxiv.org/abs/2505.13417.

A Data Schema, Taxonomy, and Annotation

Timestamped questions and source provenance. The core contains 5,619 questions on 1,392 videos: 4,052 multiple-choice questions, 869 answer/silence questions, and 698 counting questions. A question is a video–query–timestamp instance with its own reference answer. Proactive questions are instantiated at annotated times within a shared instruction; their targets track whether an event has occurred or how many occurrences have accumulated. Source grouping IDs link these related questions for provenance and annotation. All questions from one video stay together in data splits and video-bootstrap resamples. The open-answer extension contains 2,190 additional question/reference pairs on 960 existing videos.

Table 4: Fields in the evaluation schema. Demand labels, reference answers, and source identities support evaluation and analysis; routing inputs are specified separately for each policy.

Field	Meaning
`video_id`, `source`	Original video identity and contributing benchmark
`qid`, `probe_id`	Source grouping ID and unique timestamped question ID
`ask_timestamp`	Last observable video time when the query is evaluated
`question`, `options`	Query text and answer candidates where applicable
`answer`, `answer_type`	Reference answer and response format
`task_set`, `category`, `type`	Operating mode, semantic category, and diagnostic operation
`demand`, `criteria`	D_1/D_2 annotation and triggering rubric criteria
`clue_timestamps`	Reference intervals supporting the answer
`meta.test_info`	Proactive answer/silence or counting supervision at each question

Source roles. StreamingBench contributes timestamped perception and contextual queries; OVBench contributes online perception, memory, and prediction; CG-Bench contributes evidence-grounded questions; OVO-Bench contributes online temporal and proactive behavior. The construction preserves source references, standardizes timestamps and response formats, maps semantic operations, and checks evidence consistency. The new open-answer questions are written on videos selected from OVBench, CG-Bench, and StreamingBench with coverage across video-duration buckets.

Table 5: Core questions by source.

Source	Videos	Questions
StreamingBench	171	855
OVBench	705	2,360
CG-Bench	382	819
OVO-Bench	134	1,585
Total	1,392	5,619

Table 6: The seven semantic categories and all twenty task types. Task sets specify when and how the assistant operates; categories and types specify the capability being evaluated.

Category	Task types
Visual Perception	Object recognition; attribute identification; text recognition; spatial localization; scene understanding
Action & Event Recognition	Action recognition; event recognition; video summarization; occurrence counting
Temporal Memory	Memory retrieval
Temporal Localization	Temporal localization; trajectory tracking
Hallucination Verification	State verification; existence verification
Causal & Reasoning	Causal inference; entity reasoning
Predictive Reasoning	Goal prediction; movement prediction; action anticipation; proactive response

Annotation procedure. The four criteria are multi-observation composition, historical retrieval, ordered recall, and inference beyond direct observation. The original rubric was piloted on 200 items. In the independent audit, three annotators independently review 800 source-linked annotation groups spanning 1,198 timestamped questions and 568 videos. Sampling uses source, task type, and original demand strata followed by random fill, with seed 20260918 and no model outcomes. The interface hides original demand labels and model results, presents every timestamped question in each group, and records required operations and five-second sufficiency separately. Agreement is computed over the 800 group-level annotation decisions; each resulting label is attached to all questions in that group.

Table 7: Definitions of the twenty semantic task types. A task type can be instantiated in several streaming operating modes.

Type	Operation evaluated
Object recognition	Identify an object or entity from visible appearance.
Attribute identification	Identify an object's color, material, or other visible property.
Text recognition	Read text appearing in the observed video.
Spatial localization	Locate an object or action spatially in a frame.
Scene understanding	Recognize the scene and its relevant visual context.
Action recognition	Identify a person's or object's ongoing action.
Event recognition	Identify a temporally organized event.
Video summarization	Summarize events in the observed interval.
Occurrence counting	Count occurrences or instances defined by the query.
Memory retrieval	Recover information from an earlier observed interval.
Temporal Localization	Identify the interval in which relevant evidence occurs.
Trajectory tracking	Recover an entity's change in position over time.
State verification	Verify a specified state or its temporal persistence.
Existence verification	Verify whether a specified entity is present at the queried time.
Causal inference	Relate an observed outcome to its supporting events or conditions.
Entity reasoning	Combine observations to infer an entity's role or relation.
Goal prediction	Infer a likely goal or subsequent procedural step.
Movement prediction	Predict a short-term change in an entity's position.
Action anticipation	Predict the next action from the observed context.
Proactive response	Decide when the observed evidence warrants an answer.

A.1 Independent Demand Audit

The audit comprises 2,400 independent judgments, three per annotation group. Before adjudication, 476 groups receive three D_2 votes and 266 receive three D_1 votes, so 92.75% of groups are unanimous. Across all 800 groups, pairwise Cohen's κ is 0.859/0.949/0.856 and Fleiss' κ is 0.888, counting an uncertain vote as a third category. On the 784 groups with three binary votes, Fleiss' κ is 0.924.

All 58 non-unanimous groups are adjudicated. The resulting audit contains 275 D_1 and 524 D_2 annotation groups, covering 1,197 timestamped questions across 568 videos. One question keeps an uncertain label because its text refers to second 554 while its query time is 365 seconds; it is omitted from binary demand analyses. Final and original labels agree on 728 of 799 groups (91.11%): 70 change from D_1 to D_2 and one changes in the reverse direction. These corrections propagate to every associated question in the full-core label overlay, while predictions and task identities remain fixed.

Demand and realized resource returns. To test whether demand predicts benefit, we compare each resource's gain on D_2 and D_1 groups. Each annotation group has equal weight: the outcomes of its timestamped questions are averaged first. A paired bootstrap draws 5,000 video samples while retaining model and pathway pairing. Table 8 reports $\Delta_{D_2} - \Delta_{D_1}$ under the final labels. The context contrast ranges from -2.07 to $+19.21$ points, and the memory contrast is positive for three models and negative for three: $+19.87$ points for Qwen2.5-VL-72B but -14.63 for Qwen3.5-122B. The same evidence-demand rubric thus reveals very different abilities to turn an added resource into correct answers. Demand describes what a question needs; the paired outcome describes whether a given model can use what it is given.

Table 8: Demand-conditioned gain contrasts after adjudication (275 D_1 and 524 D_2 annotation groups). Each entry is the resource gain on D_2 minus the gain on D_1, relative to the same fast baseline, in points with paired 95% video-bootstrap intervals. Positive values mean the resource helps more on questions that require more than immediate perception.

Model	Long direct: $\Delta_{D_2} - \Delta_{D_1}$	Memory: $\Delta_{D_2} - \Delta_{D_1}$
Qwen2.5-VL-7B	$+3.01$ [-4.83, 10.62]	$+7.90$ [-0.03, 15.57]
InternVL3.5-38B	$+10.10$ [2.36, 18.00]	$+8.48$ [0.42, 16.63]
Qwen2.5-VL-72B	$+19.21$ [12.48, 25.97]	$+19.87$ [12.83, 26.60]
Qwen3.5-35B	-2.07 [-9.51, 5.68]	-6.70 [-14.39, 1.27]
Qwen3.5-122B	-1.78 [-9.20, 6.02]	-14.63 [-21.83, -7.42]
Qwen3.5-397B-FP8	$+1.94$ [-4.33, 8.43]	-9.78 [-16.81, -2.85]

Label sensitivity. Table 9 recomputes the memory contrast under the original and adjudicated labels on the same 799 annotation groups, and then with the single uncertain question assigned to either demand level. The sign of the contrast is unchanged for every model across all variants, and removing the separately flagged short-media record also preserves all six signs. Five-second sufficiency is recorded independently of demand: the original annotators mark 29–33 D_2 groups as answerable from five seconds, because a question can require inference even when its evidence is recent.

Table 9: Memory gain contrast $\Delta_{D_2} - \Delta_{D_1}$ under label-sensitivity variants, in points. Original and Adjudicated use the same 799 annotation groups; the last two columns include all 800 groups and assign the one uncertain question to D_1 or D_2. Every model keeps its sign across all four variants.

Model	Original	Adjudicated	Uncertain to D_1	Uncertain to D_2
Qwen2.5-VL-7B	$+10.60$	$+7.90$	$+7.90$	$+7.88$
InternVL3.5-38B	$+11.04$	$+8.48$	$+8.47$	$+8.47$
Qwen2.5-VL-72B	$+18.63$	$+19.87$	$+19.51$	$+20.03$
Qwen3.5-35B	-5.98	-6.70	-6.32	-6.89
Qwen3.5-122B	-13.75	-14.63	-14.58	-14.63
Qwen3.5-397B-FP8	-10.23	-9.78	-9.75	-9.78

B Paired Execution, Routing, and Statistical Evaluation

Table 10: Resource interventions. Each paired comparison fixes the model and query time. Direct inference uses a 256-token cap; thinking caps are 1,024 or 4,096 tokens. Memory contains prior observations available at the query time.

Configuration	Window	Thinking	Memory	Primary comparison
Fast direct	5 s	Off	Off	Recent perception
Long direct	50 s	Off	Off	Context return
Long thinking	50 s	On	Off	Thinking return
Memory-enabled	50 s	Model-specific	On	History access

Resource control. The fast condition uses a recent five-second window, direct answering, and no external memory. The long-direct condition extends the visible window to fifty seconds. Same-checkpoint thinking comparisons hold that fifty-second input and memory-off setting fixed while changing native thinking and the generation cap. All windows terminate at the query timestamp. The memory-enabled condition retrieves earlier summaries and keyframes. Same-model comparisons use the same backbone; heterogeneous comparisons pair saved fast answers with independently produced slow answers.

Response contract. Multiple-choice scoring extracts the requested answer option. Typed proactive questions specify whether an answer or silence is appropriate at the query time. Counting predictions

are parsed as numeric answers. Open answers are evaluated semantically against a reference, allowing equivalent descriptions of the same visible event. Invalid or missing answers remain in the accuracy denominator and receive zero correctness. Recomputed grounding diagnostics assign zero IoU to missing predicted intervals. For localization, the evaluation reports both answer correctness and temporal overlap; for proactive tasks, it reports Trigger-F1, recall, and false-alarm rate jointly.

Reference memory. Short-term memory contains factual summaries of earlier interactions. A long-term index stores segment summaries and selected keyframes. The coordinator retrieves prior observations for the slow prompt. The reference implementation writes the round's fast answer into memory after answering, and its long-term-memory ablation disables retrieval while fixing routing decisions. This yields paired measurements of the contribution of the read path.

Intrinsic and deployed outcomes. Each paired record falls into both-correct, fast-only-correct, slow-only-correct, or both-wrong. Intrinsic rescue is the slow-only-correct share, and intrinsic harm is the fast-only-correct share. For a routed system,

$$R_\pi = N^{-1} \sum_i \mathbf{1}[\pi(x_i) = S_2](1 - c_{i,\text{fast}})c_{i,\text{slow}}, \tag{3}$$

$$H_\pi = N^{-1} \sum_i \mathbf{1}[\pi(x_i) = S_2]c_{i,\text{fast}}(1 - c_{i,\text{slow}}). \tag{4}$$

The identity $\text{Acc}_\pi = \text{Acc}_{\text{fast}} + R_\pi - H_\pi$ provides an internal check on each table. At an at-most-ρ budget, the answer oracle chooses beneficial slow replacements up to that budget, retaining fast predictions elsewhere. A cheapest-correct oracle over several actions selects the least expensive correct configuration whenever a correct configuration exists.

Rescue-MLP. The video-disjoint baseline uses hashed question unigrams/bigrams, question length, response format, option count, query timestamp, and fast confidence. Its target is whether the slow path rescues the fast answer. Source identity, task-set/type, manual demand label, and correctness are excluded from runtime inputs. Videos are partitioned 65/15/20 for training, validation, and testing with seeds 17/29/43. Early stopping uses validation performance. At a fixed budget, scores rank the test queries and select the prescribed number of escalations. Source-held-out evaluation freezes the learned score and threshold before applying them to the fourth source.

Question-only shared-gain MLP. Frozen all-MiniLM-L6-v2 embeddings feed an MLP with hidden widths 128 and 32. Within a video split, let I_q contain all questions with question text q, and let $d_{mi} = c_{i,\text{slow},m} - c_{i,\text{fast},m}$. The target and training weight are

$$y(q) = \sum_m \sum_{i \in I_q} d_{mi}, \qquad w(q) = \text{clip}\left(|y(q)| \log\left(1 + \sum_m \sum_{i \in I_q} |d_{mi}| \right), 0.1, 20 \right). \tag{5}$$

Because the target sums gains over all backbones m, the router outputs one score per question that every backbone shares. Targets and weights are constructed separately for training and validation videos. Training normalizes embeddings and targets using training-set moments, divides weights by their training mean, and minimizes weighted Smooth-L1 loss. Validation loss controls early stopping. Model-specific MiniLM baselines predict signed gain with Smooth-L1 or rescue with binary cross-entropy. At inference, these three MiniLM policies use question text alone. Offline fixed-budget evaluation selects the prescribed top-k scores; deployment thresholds are selected on validation videos and then frozen.

Uncertainty and grouping. Routing and uniform open-answer comparisons use 5,000 paired video resamples. Each selected video contributes all of its questions, proactive timestamps, and evaluated model/path configurations. Routing results average three video splits. Joint frozen-gate intervals use shared video resampling weights across overlapping splits; each replicate retains all six backbones together. Open-answer video resampling likewise retains all models' answers to a question. Binary trigger and count metrics use separate question sets; invalid binary outputs remain silent for emission metrics and incorrect for strict accuracy, while count MAE is reported with valid-output coverage.

C Reference-System Coverage and Task-Level Diagnostics

Evaluation tiers and paired quantities. The reference tier compares four API systems and four local systems on the same 4,915 questions, the common coverage of the saved reference runs: 4,046 multiple-choice and 869 answer/silence questions. The four API reference runs did not execute the 704 occurrence-counting questions (698 proactive counting and 6 realtime counting questions), so the tier excludes them for all eight systems and every cross-system comparison uses identical questions. The full-core tier independently evaluates six open-model configurations on all 5,619 questions, including these 704 counting questions. Table 2 uses intrinsic rescue/harm throughout, as does Figure 4: rescue is a fast error corrected by slow execution, harm is a correct fast answer lost under slow execution, and Net equals Slow minus Fast. Deployed rescue/harm compare the fast answer with the routed answer, giving Routed minus Fast (Table 11). These are two summaries of the same paired records, rather than additional model runs.

Reference router and operating point. Routed uses the saved question-only weighted-net MLP: frozen all-MiniLM-L6-v2 embeddings feed hidden layers of width 128 and 32, and the model selects slow when its score exceeds 0.109948. The training checkpoint groups identical question text, uses a 70/15/15 question-group split with seed 42, and selects its threshold on validation groups. The same saved decisions apply to every reference backbone: 2,324 of 4,915 questions select slow (47.28%). This reference operating point describes pipeline behavior on the shared subset. Generalization comparisons in Section 5.2 train separate routers on video-disjoint partitions and use independently saved fast/long-direct answers with memory disabled.

Why the repeated backbone differs. The reference Qwen2.5-VL-7B row uses the reference streaming prompts and memory pipeline on 4,915 questions. Its full-core row comes from independent inference under the controlled five-second fast and fifty-second memory-enabled slow protocol on 5,619 questions. Thus both question coverage and execution settings differ; the two rows characterize these two configurations of the same backbone. Table 10 defines the controlled interventions, and Table 12 gives the full-core paired quantities.

Table 11: Deployed reference routing on the 4,915 shared non-counting questions. The question-only MiniLM–MLP selects slow on 47.28% for every backbone. Rescue/Harm compare fast with the routed output; Net equals Routed minus Fast. All scores are percentages; Net is in points.

Model	Routed	Deployed rescue	Deployed harm	Net
Claude-Sonnet-4.5	48.30	6.41	2.60	+3.80
GPT-4o	55.99	7.12	2.52	+4.60
GPT-5.2	59.33	5.27	2.58	+2.69
Gemini-3-flash	67.30	6.08	1.63	+4.46
MiniCPM-2.6-8B	51.25	6.08	4.76	+1.32
Qwen2.5-VL-7B	47.87	11.82	4.80	+7.02
InternVL3.5-8B	57.38	5.70	3.09	+2.60
LLaVA-NeXT-Video-7B	41.30	2.56	2.20	+0.37

Resource value changes across operating modes. Figure 6 complements the main-text rescue/harm decomposition (Figure 4) with absolute accuracies under fast, slow, routed, and oracle execution. On realtime queries, MiniCPM-2.6-8B has 6.2% rescue and 21.4% harm, yielding a 15.2-point loss; InternVL3.5-8B has 5.9% rescue and 18.6% harm, yielding a 12.7-point loss. On proactive queries, Qwen2.5-VL-7B has 21.3% rescue and 7.0% harm, yielding a 14.3-point gain. Gemini-3-flash has positive net changes in all five task sets. The taxonomy thus separates immediate perception from temporal evidence use, and the transition counts distinguish limited benefit from substantial rescue cancelled by substantial harm.

Task-specific memory ablation. Retrieval is disabled while keeping the backbone and the remaining reference pipeline fixed, across 1,323 questions from four memory-relevant task types. Figure 7 reports the task-wise comparison. Removing retrieval lowers Gemini-3-flash's aggregate accuracy by 3.10 points and raises InternVL3.5-8B's by 0.87 points. The different effects on retrieval, tracking, localization, and verification show why memory should be evaluated per capability and model.

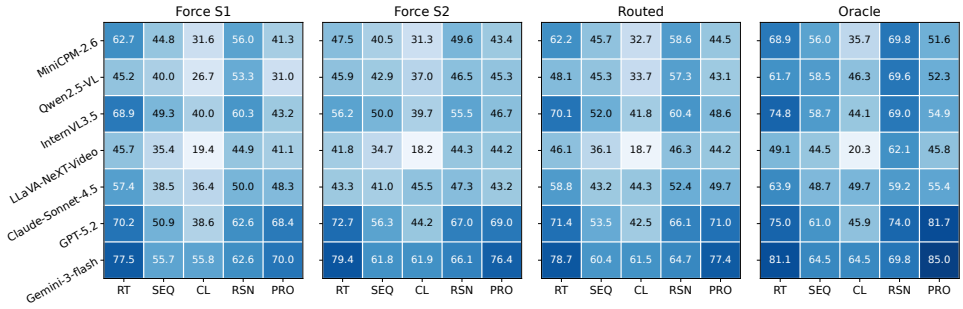


Figure 6: Per-task-set accuracy of seven reference systems on the same 4,915 questions as Table 2, under fast, slow, routed, and oracle execution. RT/SEQ/CL/RSN/PRO denote realtime, sequential, clue, reasoning, and proactive tasks. Extra computation changes different capabilities in different directions, while the oracle exposes complementary answers.

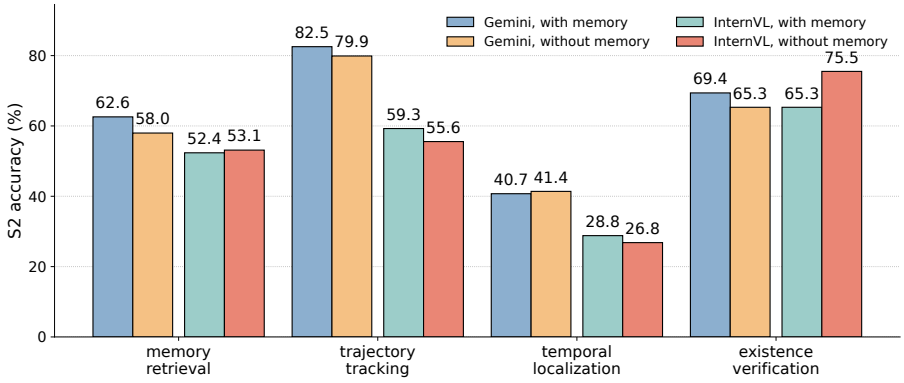


Figure 7: Reference memory ablation by task type for Gemini-3-flash and InternVL3.5-8B. Retrieval is disabled with the model, routing decisions, and remaining memory pipeline held fixed on 1,323 questions.

Table 12: Independent fast and memory-enabled slow paths on all 5,619 core questions. All entries are percentages; rescue and harm are intrinsic paired events. Oracle equals fast accuracy plus rescue, up to rounding.

Model	Fast	Slow	Rescue	Harm	Oracle
Qwen2.5-VL-7B	38.99	42.41	14.20	10.78	53.19
InternVL3.5-38B	45.19	47.84	15.13	12.48	60.31
Qwen3.5-35B-A3B	49.17	54.69	18.62	13.10	67.79
Qwen2.5-VL-72B	41.64	54.00	19.10	6.74	60.74
Qwen3.5-122B-A10B	47.87	55.90	19.93	11.91	67.81
Qwen3.5-397B-A17B-FP8	53.78	56.29	14.84	12.33	68.62

D Additional Resource and Model Results

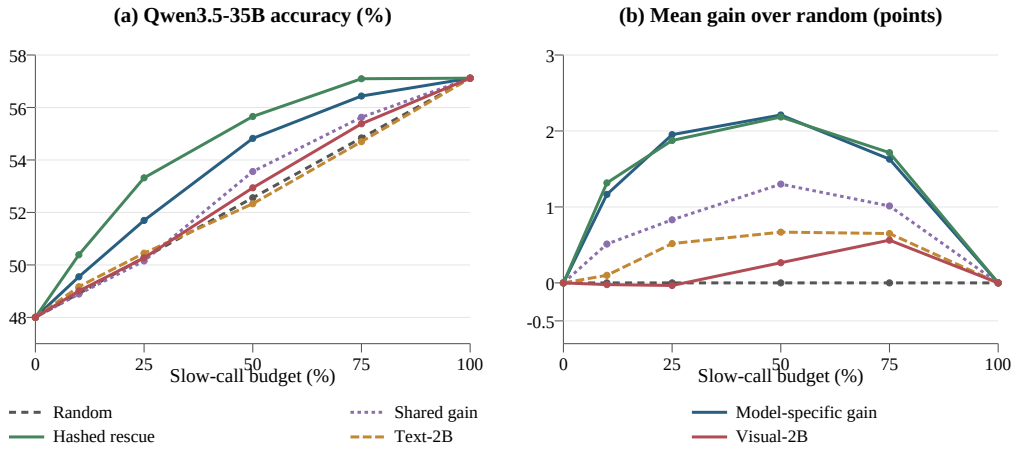


Figure 8: Routing budget curves averaged over three video splits. (a) Qwen3.5-35B accuracy, with values in Table 16. (b) Six-backbone mean improvement over rate-matched random. All policies share paired answers; frozen gates and learned routers differ in their information and supervision.

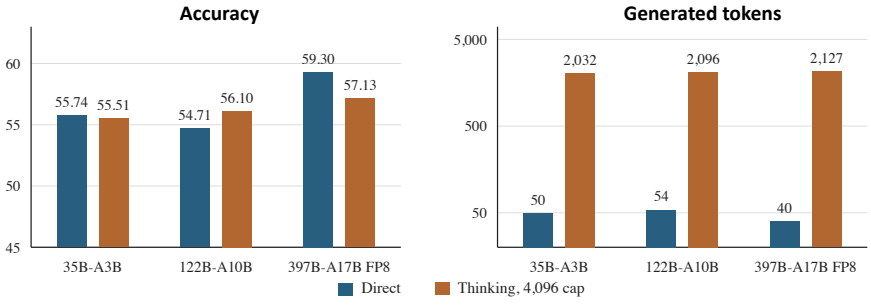


Figure 9: Same-checkpoint thinking at a fixed 50-second context without memory, over all 5,619 core questions. Left: direct and 4,096-cap thinking accuracy. Right: actual mean generated tokens on a logarithmic scale. The 397B checkpoint uses FP8.

Generation-budget utilization. Budget utilization helps explain the weak 35B result at a 1,024-token cap: accuracy falls to 47.23%, and 56.4% of outputs hit the cap. At 4,096 tokens, the cap-hit rate is still 31.4%. These observations connect the compute setting to answer completion and support recording realized output length alongside the nominal cap. A confidence policy that upgrades half

Table 13: Same-checkpoint compute control on 5,619 questions. All rows use fifty-second context and no external memory. Tokens are actual mean generated tokens. Cap-hit rates for 35B are 56.4% at 1,024 tokens and 31.4% at 4,096 tokens.

Model	Mode	Accuracy (%)	Tokens
Qwen3.5-35B-A3B	Direct	55.74	49.80
	Thinking, 1,024 cap	47.23	792.10
	Thinking, 4,096 cap	55.51	2,032.00
Qwen3.5-122B-A10B	Direct	54.71	54.17
	Thinking, 4,096 cap	56.10	2,096.00
Qwen3.5-397B-A17B-FP8	Direct	59.30	40.06
	Thinking, 4,096 cap	57.13	2,127.00

Table 14: Heterogeneous pairing with Qwen2.5-VL-7B fast predictions, which score 38.99% on the 5,619 core questions. Independent slow inference uses memory. Entries are percentages except net gain in points.

Slow model	Accuracy	Net	Rescue	Harm	Oracle
Qwen3.5-35B-A3B	54.69	+15.70	26.57	10.87	65.56
Qwen2.5-VL-72B	54.00	+15.00	22.80	7.80	61.79
Qwen3.5-122B-A10B	55.90	+16.91	27.16	10.25	66.15
Qwen3.5-397B-A17B-FP8	56.29	+17.30	27.51	10.22	66.51

of long-direct queries to thinking reaches 55.86%, compared with 55.53% for matched random allocation. The 71.67% cheapest-correct oracle demonstrates that paired configurations contain useful alternatives that confidence only partially identifies.

Task-specific memory return. The memory ablation covers memory retrieval, trajectory tracking, temporal localization, and existence verification. For Gemini-3-flash, memory-enabled versus retrieval-disabled accuracies are 62.6/58.0, 82.5/79.9, 40.7/41.4, and 69.4/65.3. For InternVL3.5-8B they are 52.4/53.1, 59.3/55.6, 28.8/26.8, and 65.3/75.5. These paired values show why a memory component should be evaluated by the task and model combination on which it is used.

Table 15: Three batch-one latency repeats on the same 516-query Qwen2.5-VL-72B sample. Accuracy is specific to this sample and execution backend. Time is in seconds.

Configuration	Accuracy (%)	P50	P90	P99
5-second direct	44.19	0.964	1.581	1.651
4,096-cap long context	54.84	5.571	7.011	7.709
Independent memory-enabled	55.23	5.774	7.161	9.472

Cost components. In the repeated latency study, video decoding/window export has approximately 0.62-second P50 and independent slow-model execution approximately 4.80-second P50. A separate CPU replay of the saved text encoder and MLP uses 341 questions, with three warmed-up P50 measurements of 12.1, 11.4, and 11.3 ms. The second pass has P90/P99 of 16.8/31.4 ms. These measurements locate the router's computational cost relative to video processing.

Table 16: Measured budget sweep underlying Figure 8. Accuracy is in percent, averaged over three held-out video splits.

Escalation rate	0%	10%	25%	50%	75%	100%
Random	48.00	48.91	50.28	52.56	54.84	57.12
Shared gain	48.00	48.89	50.15	53.56	55.63	57.12
Model-specific gain	48.00	49.55	51.70	54.82	56.44	57.12
Hashed rescue	48.00	50.39	53.32	55.66	57.10	57.12
Text-2B	48.00	49.17	50.45	52.33	54.70	57.12
Visual-2B	48.00	49.01	50.27	52.94	55.38	57.12

E VISION-LANGUAGE ROUTERS

This appendix details the two vision-language routers summarized in Section 5.2. Both use Qwen3-VL-2B-Instruct and decide between the same 5-second fast and 50-second direct paths. Appendix E.1 asks whether the pretrained model, prompted with the question and the most recent frames, can identify which questions benefit from a longer history without any paired supervision. Appendix E.2 keeps its encoder frozen but trains a small routing head on paired gain, and compares visual with text-only input under identical training.

E.1 FROZEN ZERO-SHOT GATES

Gate and paired answer reuse. We use Qwen3-VL-2B-Instruct (revision `89644892`) with frozen weights and a fixed prompt to score the usefulness of extending the recent video context from 5 to 50 seconds, i.e., of choosing the long-direct path over the fast path of Table 10. Both conditions receive the same system instruction, question, options when available, and query timestamp. The visual condition additionally receives at most five frames from the recent five seconds and their actual timestamps; the text condition receives neither frames nor frame timestamps. Frames are sampled at 1 fps at $t-4,\dots,t$ for query time t; each target time takes the last decoded frame not later than it, so no frame follows the query. Images are resized to a maximum long side of 320 pixels before processing, with processor bounds of 65,536–131,072 pixels per image. Most questions (5,581 of 5,619) receive five frames; the remaining 38 questions near the start of their videos receive two to four frames. The fixed system prompt is:

```
You allocate video context for a streaming video question. Choose between:
A: FAST, answer using only the recent 5-second video window.
B: SLOW, answer using the recent 50-second video window, including earlier events.
Choose B when earlier visual context is likely to improve the answer: an event or referent
is missing from the recent window, a history-dependent count or temporal relation is
needed, or the relevant evidence is incomplete. Choose A when the recent scene or action
provides sufficient evidence. Extra context can also distract, so do not choose B merely
because a question sounds difficult. Judge the expected usefulness of earlier context, not
the length of the answer. Do not answer the question itself.
The frames, when provided, are in chronological order and contain no future observations.
When no frames are provided, use the question and choices alone.
Return exactly one letter: A or B.
```

The user message lists the question, the choices when present, the query time in seconds, and, for the visual condition only, the frame times after the images. The decision letters A and B are unrelated to answer options.

The gate produces one token, restricted to A (fast) or B (slow), using greedy decoding and processed log probabilities. If ℓ_A and ℓ_B are the two log probabilities, the ranking score is $\sigma(\ell_B - \ell_A)$. The relative decision-token probability serves as the ranking score. Taken at face value, the generated letters would escalate 80.94% (visual) and 97.03% (text) of questions, so budgets are enforced by ranking scores. The gate receives neither reference answers, demand labels, cached fast/slow answers, nor the identity of the answering backbone, and there is no fine-tuning or prompt search. The 5,619 questions produce 11,238 routing records across the two input conditions, and a single set of scores is shared across the six answering backbones; the gate is therefore a model-agnostic judgment of context need.

The fast and long-direct paths answer queries independently and have no routing-dependent state updates. Consequently, for a fixed cached pair and routing decision r_i, the routed correctness is

exactly $c_{i,\text{fast}} + r_i(c_{i,\text{slow}} - c_{i,\text{fast}})$. All cached outcomes, including incorrect and invalid answers, remain in the denominator. This fixed-prediction design evaluates alternative routing policies and budgets by selecting the corresponding saved answers. Gate latency is measured separately below.

Fixed budgets and comparison with learned routers. For a set of N questions, the offline policy selects the top $\text{round}(0.25N)$ scores, breaking ties by a deterministic question-ID hash. Random uses the analytic expectation at the same realized rate; the budget-matched oracle ranks by true gain. All seven routing policies share identical test questions and paired answers in each of the three splits. Table 3 is the unified three-split comparison; Table 17 reports the detailed seed-17 result. At 25%, model-specific gain averages 49.69%, hashed rescue 49.62%, Text-2B 48.26%, and Visual-2B 47.71%, against 47.74% for random. The equal-split budget sweep appears in Table 18.

Table 17: Accuracy (%) at 25% escalation on the seed-17 test split (1,264 questions) for all routers. Text-2B and Visual-2B are the frozen gates; Shared, Gain, Rescue, Hashed, and Conf. are the routers of Table 3. Random is the analytic rate-matched expectation. The mean row averages the six backbones equally.

Model	Random	Text-2B	Visual-2B	Shared	Gain	Rescue	Hashed	Conf.	Oracle
Qwen2.5-VL-7B	39.18	39.24	39.16	39.72	40.74	40.43	40.98	39.87	53.16
InternVL3.5-38B	45.47	46.28	45.81	47.86	47.39	46.60	47.23	46.52	58.78
Qwen3.5-35B-A3B	50.69	50.40	49.76	50.08	52.22	51.58	53.48	53.09	66.06
Qwen2.5-VL-72B	43.67	43.35	43.67	44.30	45.57	45.49	46.28	43.51	59.34
Qwen3.5-122B-A10B	49.78	50.16	49.84	50.71	52.14	50.32	51.11	50.32	65.35
Qwen3.5-397B-A17B-FP8	56.31	56.88	55.14	58.86	57.83	58.07	58.15	57.52	69.15
Mean	47.52	47.72	47.23	48.59	49.31	48.75	49.54	48.47	61.97

Table 18: Frozen-gate budget sweep: accuracy (%) averaged equally over six backbones and the three predefined video splits. Endpoints are always-fast and always-long-direct. Differences use unrounded values.

Budget	0%	10%	25%	50%	75%	100%
Random	46.01	46.70	47.74	49.47	51.21	52.94
Text-2B	46.01	46.80	48.26	50.14	51.86	52.94
Visual-2B	46.01	46.68	47.71	49.74	51.77	52.94
Visual − Text	0.00	-0.12	-0.55	-0.40	-0.09	0.00

On the Qwen3.5-35B backbone, for example, the visual gate rescues 3.64% and harms 2.22% of test questions at 25%, a net gain of 1.42 points over always-fast (48.34% to 49.76%), whereas random escalation at the same rate reaches 50.69% in expectation. The gate thus escalates questions whose slow answers help less often than an average question's.

Uncertainty across splits. We reuse the original video partitions with seeds 17, 29, and 43. These are sensitivity analyses of one frozen score set, not independent inference runs. Their test sets have 1,264, 1,021, and 1,064 questions, respectively; together they contain 3,349 split memberships but only 2,749 distinct questions on 714 distinct videos. We use 5,000 percentile bootstrap replicates, resampling videos with replacement and retaining all questions, both routing conditions, and all six backbones together. The primary intervals keep scores fixed and reselect top-k within each resampled population; complementary intervals hold the original routing decisions fixed. Both methods match the random reference to the realized rate in each replicate. The equal-weight mean across splits uses common resampling weights over the full 1,392-video universe before restricting each replicate to each test split, preserving overlap between splits. Backbones are averaged within each replicate and are not independently resampled, and all intervals are conditional on the frozen gate, scores, and cached answer outputs (Table 19).

Full-pool reallocation selects 1,405 questions for each gate. Six-backbone means are 47.64% (random), 48.28% (text), and 47.52% (visual), with Visual-2B below Text-2B on every backbone. This supplementary analysis uses the full 5,619-question pool; the main comparison uses the three predefined test splits.

Table 19: Frozen visual gate at 25% escalation. Differences are percentage points, averaged over six answering backbones, with 95% paired video-bootstrap intervals. The equal-split mean averages the three overlapping splits using shared video resamples; the full pool overlaps all splits and is descriptive. Scores remain fixed; each replicate reselects the budgeted top-k queries.

Evaluation	Visual − Text	Visual − Random
Split 17	-0.49 $[-1.41, 0.37]$	-0.29 $[-1.01, 0.20]$
Split 29	-0.56 $[-1.59, 0.44]$	-0.23 $[-0.96, 0.51]$
Split 43	-0.61 $[-1.60, 0.28]$	$+0.41$ $[-0.34, 1.14]$
Equal split mean	-0.55 $[-1.21, 0.06]$	-0.03 $[-0.53, 0.40]$
Full 5,619-question pool	-0.75 $[-1.15, -0.35]$	-0.12 $[-0.45, 0.21]$

Tie-breaking sensitivity. The complete visual and text score sets contain 145 and 67 distinct values, respectively. We evaluate the original hash ordering, its reverse, ascending and descending question-ID order, and 200 deterministic salted hash orderings. Each ordering is applied to both conditions and all six backbones without reading correctness. The six-backbone mean visual-minus-text remains negative under all 204 rules in every split. Across the 200 salted rules, its ranges are $[-0.86, -0.20]$, $[-0.73, -0.21]$, and $[-0.91, -0.33]$ points for splits 17, 29, and 43. The ranges summarize sensitivity to tie order; Table 19 quantifies uncertainty across videos.

Frozen thresholds and realized invocation rates. For each split and condition, a threshold is set using only its validation scores: the midpoint of the scores at ranks k and $k + 1$, with $k = \mathrm{round}(0.25N_{\mathrm{val}})$. A test query is escalated when its score is at least the threshold. Ties at the boundary mean that the nominal 25% target is not attained exactly even on validation data (Table 20). Thresholds stay fixed during test-video resampling, so these intervals condition on the selected validation thresholds. Each test query is routed independently with the frozen threshold, and its saved fast or slow answer determines correctness.

Table 20: Realized invocation rates (%) of the nominal 25% frozen-threshold policy. Both thresholds are identical across the three splits: 0.9241418268 for visual and 0.9947798744 for text. The boundary scores tie in every validation set.

	Validation		Test	
Split	Visual	Text	Visual	Text
17	25.45	30.10	26.34	26.50
29	26.33	27.42	25.76	26.44
43	29.67	30.08	26.79	25.09

Threshold-based visual-minus-text differences average -0.67, -0.57, and -0.44 points across the six backbones in the three splits, and each paired interval includes zero. Because invocation rates differ, we also subtract each policy's own rate-matched random expectation and compare the resulting excess accuracies. These contrasts are -0.66 ($[-1.53, 0.19]$), -0.52 ($[-1.48, 0.42]$), and -0.54 ($[-1.45, 0.38]$) points. The resulting contrast measures excess accuracy after accounting for each policy's realized invocation rate.

Threshold transfer to held-out sources. We calibrate each gate on validation videos from three sources and freeze its threshold before evaluating the fourth. For seeds 17, 29, and 43, validation uses other-source videos whose existing hash split value is below 0.2, matching the learned-router source-transfer protocol. We reuse the original gate scores and paired answers for all 5,619 questions. Each source is evaluated in full, with 5,000 shared video-bootstrap draws retaining both policies and all backbones; thresholds remain fixed in each draw. Table 21 averages the three calibration splits and six answering backbones.

The visual gate's mean invocation rate ranges from 16.23% on OVBench to 50.53% on Streaming-Bench. Its excess accuracy over rate-matched random ranges from -0.49 to $+0.09$ points, with every interval spanning zero. Text gating also exhibits substantial rate shifts, from 8.56% to 60.82%, while retaining positive excess accuracy on OVBench and OVO-Bench. Thus both budget stability

and selection quality depend on the target source. All thresholds are chosen without target-source questions or answers.

Table 21: Frozen-gate threshold transfer to an entire held-out source. Thresholds use validation videos from the other three sources. Entries average three calibration splits and six backbones. Rates are percentages; excess accuracy over each policy's rate-matched random reference is in points, with 95% paired video-bootstrap intervals.

Source	Slow-call rate Text	Slow-call rate Visual	Excess accuracy Text	Excess accuracy Visual
CG-Bench	52.50	36.06	-0.20 $[-1.00, 0.60]$	-0.49 $[-1.25, 0.24]$
OVBench	8.56	16.23	$+0.43$ $[0.12, 0.76]$	-0.15 $[-0.53, 0.21]$
OVO-Bench	60.82	26.88	$+2.67$ $[1.79, 3.51]$	$+0.09$ $[-0.70, 0.80]$
StreamingBench	16.49	50.53	-0.04 $[-0.71, 0.62]$	$+0.09$ $[-0.79, 0.95]$

Exploratory decomposition of changed decisions. Let $d_{im} = c_{i,\mathrm{slow},m} - c_{i,\mathrm{fast},m}$. Relative to text routing, define $\mathcal{A} = \{i : r_i^V = 1, r_i^T = 0\}$ and $\mathcal{C} = \{i : r_i^V = 0, r_i^T = 1\}$ as added and cancelled escalations. The paired accuracy difference for backbone m decomposes exactly as

$$\mathrm{Acc}_{V,m} - \mathrm{Acc}_{T,m} = \frac{1}{N}\sum_{i\in\mathcal{A}} d_{im} - \frac{1}{N}\sum_{i\in\mathcal{C}} d_{im}. \tag{6}$$

Every contribution uses the complete test-set denominator. Group intervals hold the original changed-decision sets fixed while resampling videos; they correspond to the fixed-decision analysis rather than reranking in every replicate.

Table 22: Exploratory attribution at the fixed 25% budget. Contributions are percentage points of full-test accuracy, averaged over six backbones. The counting column isolates that task's contribution to the net change. Added and cancelled counts refer to disjoint sets of questions.

Split	Added N	Cancelled N	Added	Cancelled	Net	Counting net
17	206	206	$+0.83$	-1.32	-0.49	-1.08
29	161	161	$+0.54$	-1.09	-0.56	-0.42
43	158	158	$+0.63$	-1.24	-0.61	-0.72

Added escalations yield positive net gains in all three splits, but cancelled escalations lose more correctness than those additions recover (Table 22; entries are rounded independently). Counting questions contribute negatively in all three splits. In split 17, the visual condition adds 3 counting escalations and cancels 93, yielding a net contribution of -1.08 points ($[-1.77, -0.48]$) to full-test accuracy. Other task types partly offset this contribution. This exploratory decomposition localizes the observed loss to missed useful escalations. Task intervals are pointwise, and source and task tables provide separate marginal decompositions of the same decisions.

Router overhead. We time the same frozen gate on one GPU with weights resident and prefix and multimodal caches disabled, on 256 label-independent questions from 170 test videos, using 512 measured requests per setting after 96 warm-up requests (Table 23). The visual gate adds 30.72 ms per query over the text gate, or 2.73× its latency, almost entirely in model serving (45.91 vs. 17.67 ms). These timings start from in-memory JPEG frames. Decoding the frames directly from video adds 604.74 ms per query on average (P95 1,730.75 ms), versus 0.83 ms when an application frame cache already holds them. Model initialization (26.48 s) is excluded. For comparison, the MiniLM question-only router runs in about 11–12 ms on CPU (Appendix D).

Scope and reproducibility. The evaluated gate uses frozen Qwen3-VL-2B weights, a fixed prompt, sparse recent frames, and a shared score across answering backbones. Gate inference ran in vLLM 0.19.1 in bf16 on four GPUs identified by the driver as NVIDIA L20X, with one replica per GPU. The released records specify stable question IDs, model revision, prompts, decision log probabilities, paired correctness, split and tie rules, and analysis seeds. The main results use the original frozen scores throughout.

Table 23: Frozen-gate overhead on one GPU. Batch-one latencies are per query; batch-32 figures give offline throughput. Times exclude frame decoding.

Gate input	Mean (ms)	P50 (ms)	P95 (ms)	Batch-32 (q/s)	Batch-32 (ms/q)
Text-2B	17.80	16.70	22.62	402.01	2.49
Visual-2B	48.52	47.50	56.03	59.33	16.85

E.2 Learned Gates

Architecture and training. The learned gates keep every Qwen3-VL-2B-Instruct parameter frozen (bf16) and train only a routing head; there is no LoRA or end-to-end fine-tuning. The text input is the question, options, and query time under the same prompt as the frozen gates; the visual input adds at most five causal frames from the recent five seconds with their timestamps, sampled and resized as in Appendix E.1. The representation is the 2,048-dimensional final hidden state of the last input token after the assistant generation prompt, obtained by a single prefill without generating a decision token. Features are standardized with training-split statistics (standard-deviation floor 10^{-4}). Each head is an MLP 2048→128→32→1 with ReLU and dropout 0.1 (266,433 parameters). For backbone m and question i its target is the signed paired gain $y_{im} = c_{i,\mathrm{slow},m} - c_{i,\mathrm{fast},m} \in \{-1, 0, +1\}$, so the head learns rescue, harm, and no change jointly. Training uses Smooth-L1 loss ($\beta = 1$), AdamW (learning rate 10^{-3}, weight decay 10^{-4}), batch size 256, at most 100 epochs, and early stopping after ten epochs without validation improvement, keeping the lowest-validation-loss checkpoint. There is one head per input type, backbone, and split, i.e., $2 \times 6 \times 3 = 36$ heads, with no tuning on test data. The predicted gain ranks questions for escalation; it is not a calibrated probability.

Information and evaluation. At inference the gates receive no reference answers, demand labels, cached answers, correctness, source, or task type. Training correctness comes only from training videos, and validation correctness only selects checkpoints and thresholds. Features are extracted once for all 5,619 questions and both inputs (11,238 samples), without labels; standardization and head training happen after splitting. The splits are the seed-17/29/43 video partitions of Table 3, with 3,408/947/1,264, 3,679/919/1,021, and 3,817/738/1,064 training/validation/test questions. Routed correctness is computed from the cached fast and long-direct answers as $c_{i,\mathrm{fast}} + r_i(c_{i,\mathrm{slow}} - c_{i,\mathrm{fast}})$, so no answering model is rerun, and $\mathrm{Acc}_\pi = \mathrm{Acc}_{\mathrm{fast}} + R_\pi - H_\pi$ holds exactly. Fixed-budget evaluation escalates the top $\mathrm{round}(0.25N)$ test questions per backbone, breaking ties by a label-free question hash; the realized mean rate is 24.99%. We report per-split six-backbone means and then average the three splits equally. The splits share one feature extraction and overlap in questions, so they are sensitivity analyses rather than independent datasets.

Table 24: All routers at 25% slow calls, averaged over three splits and six backbones. Rescue and harm are deployed rates over all test questions; Δ is accuracy minus rate-matched random, in points. Always-fast and always-slow reach 46.01% and 52.94%; Oracle uses test correctness and is an upper bound.

Router	Accuracy (%)	Rescue (%)	Harm (%)	Δ vs. random
Visual-2B, learned	49.96	5.80	1.84	+2.22
Text-2B, learned	49.91	5.72	1.81	+2.17
MiniLM model-specific gain	49.69	5.43	1.75	+1.95
Hashed rescue†	49.62	6.23	2.62	+1.88
MiniLM model-specific rescue	49.21	5.53	2.33	+1.47
MiniLM shared gain	48.57	4.43	1.86	+0.83
Confidence†	48.51	5.69	3.18	+0.77
Text-2B, frozen	48.26	4.08	1.83	+0.52
Visual-2B, frozen	47.71	4.14	2.44	−0.03
Random (rate-matched)	47.74	–	–	0.00
Oracle	62.79	–	–	+15.05

Results. The learned visual gate exceeds rate-matched random on every backbone (Table 3), and its six-backbone mean is 2.22 points above random and 2.26 above the frozen visual gate. Relative to the

frozen gate, it raises rescue from 4.14% to 5.80% and lowers harm from 2.44% to 1.84% (Table 24). This comparison changes several factors at once: supervision, per-backbone scores instead of one shared score, and the scoring path (a prefill feature with an MLP head instead of a generated decision token). The same-encoder learned text gate isolates the visual input, and there the difference is small and split-dependent: -0.76, $+0.65$, and $+0.27$ points in splits 17, 29, and 43 (Table 25). The learned visual gate is below the learned text gate on two of six backbones (Qwen3.5-35B and 122B).

Table 25: Learned gates by split: six-backbone mean accuracy (%) at 25% slow calls. The last column is learned visual minus learned text, in points.

Split	Random	Visual, frozen	MiniLM gain	Text, learned	Visual, learned	V−T
17	47.52	47.23	49.31	50.11	49.34	-0.76
29	48.14	47.91	49.95	49.77	50.42	$+0.65$
43	47.57	47.98	49.81	49.86	50.13	$+0.27$
Mean	47.74	47.71	49.69	49.91	49.96	$+0.05$

Uncertainty. Intervals use 5,000 paired video bootstrap replicates (seed 20260926). Each replicate draws common weights over the full 1,392-video universe and applies them to all three test splits, retaining every question of a video, all six backbones, and the overlap between splits, before averaging splits equally (Table 26). The intervals condition on the trained checkpoints and the original per-question decisions: heads are not retrained and top-k is not reselected within replicates, and random is matched to the realized rate in each replicate. These fixed-decision intervals therefore differ in definition from the reselected-top-k intervals of the frozen gates (Table 19) and should not be compared as the same statistic.

Table 26: Paired differences for the learned visual gate at 25% slow calls: three-split, six-backbone means in points with 95% paired video-bootstrap intervals conditional on the trained heads and decisions.

Contrast	Difference [95% CI]
Learned visual − random	$+2.22$ $[1.79, 2.63]$
Learned visual − frozen visual	$+2.26$ $[1.57, 2.94]$
Learned visual − frozen text	$+1.71$ $[1.01, 2.41]$
Learned visual − hashed rescue	$+0.35$ $[-0.20, 0.90]$
Learned visual − MiniLM model-specific gain	$+0.27$ $[-0.27, 0.79]$
Learned visual − learned text	$+0.05$ $[-0.40, 0.47]$

Frozen thresholds and budget curves. For online decisions, each head's threshold is the midpoint between the validation scores ranked $\mathrm{round}(0.25N)$ and next; test questions are escalated independently when their score reaches it, without forcing an exact rate. The frozen gates are recomputed under the same rule. The learned visual gate then reaches 49.99% at a 25.00% mean test rate, $+2.24$ points over its rate-matched random ($[1.82, 2.64]$); per-backbone rates range from 20.66% to 27.38% and gains from $+1.50$ to $+2.99$ points (Table 27). Its difference from the learned text gate, $+0.05$ ($[-0.40, 0.48]$), is unchanged. Because the realized rates differ slightly, the fixed-budget comparison remains the rate-exact one. These are offline evaluations of per-question thresholds on cached stateless answers, not an online replay. Across budgets, both learned gates dominate random and the frozen gate at every interior point, and the visual and text gates differ by at most 0.41 points (Table 28).

Where frames change decisions. Relative to the learned text gate at 25%, we decompose the accuracy difference into added escalations (visual escalates, text does not) and cancelled escalations, as in Appendix E.1; contributions are points of total test accuracy, averaged over backbones and then splits. By source, OVBench contributes $+0.24$ points, while CG-Bench (-0.04), OVO-Bench (-0.05), and StreamingBench (-0.10) are negative. By task, state verification ($+0.16$), memory retrieval ($+0.11$), and temporal localization ($+0.07$) contribute most, offset by text recognition (-0.08), spatial localization (-0.07), and event recognition (-0.04). These descriptive attributions carry no multiple-comparison correction and do not establish a visual advantage on any task.

Table 27: Frozen validation thresholds at a nominal 25% target: three-split, six-backbone means. Δ is accuracy minus random at each policy's own realized rate, in points.

Policy	Slow calls (%)	Accuracy (%)	Rescue (%)	Harm (%)	Δ vs. random
Visual-2B, learned	25.00	49.99	5.84	1.85	+2.24
Text-2B, learned	25.37	49.94	5.78	1.85	+2.15
Visual-2B, frozen	26.30	47.76	4.32	2.57	−0.07
Text-2B, frozen	26.01	48.32	4.25	1.94	+0.50

Table 28: Budget sweep with fixed scores: three-split, six-backbone mean accuracy (%). No head is retrained or reselected per budget; 0% and 100% are always-fast and always-slow.

Budget	0%	10%	25%	50%	75%	100%
Random	46.01	46.70	47.74	49.47	51.21	52.94
Visual-2B, frozen	46.01	46.68	47.71	49.74	51.77	52.94
MiniLM model-specific gain	46.01	47.86	49.69	51.68	52.84	52.94
Text-2B, learned	46.01	48.04	49.91	51.77	52.65	52.94
Visual-2B, learned	46.01	48.03	49.96	51.91	53.06	52.94
Oracle	46.01	55.98	62.79	62.79	62.79	52.94

Scope, cost, and inputs. These results concern one frozen 2B encoder, one prompt, sparse recent frames, and small heads trained on paired gain; they do not test end-to-end visual fine-tuning or richer visual context. The router reuses the fast path's recent frames and does not read the additional history before deciding. Feature extraction ran once on four GPUs identified by the driver as NVIDIA L20X (about 65–70 s per shard including initialization), and each split's twelve heads trained in under five seconds; these offline timings are not per-query latency. The learned path extracts a prefill feature in PyTorch rather than generating a token in vLLM, so the frozen-gate latencies of Table 23 do not transfer to it. No input frame follows its query time. Four training or validation questions from one source video have frames that stop well before the target five-second window; they were kept unchanged, none is in any test split, and all test frames lie within the intended window.

F Full-Video Replay and Complete-Stream Cost

Observed-prefix execution. The frozen replay manifest contains 326 videos and all 1,654 core questions on those videos. Each video is replayed through its final query timestamp under always-fast, always-memory, random, and gain-router policies for seeds 17/29/43. The resulting 3,912 video–policy–seed runs contain 19,848 query events and 23,076 memory-update events. Router training and validation exclude every replay video. The random policy escalates each query with probability 0.25; the gain router applies its validation threshold to question text. Their realized mean rates are 25.74% and 13.18%, respectively.

Independent visual state. Memory-enabled policies summarize each completed sixty-second interval using sixteen uniformly selected frames from the causal 1 fps stream. The summary has a 96-token generation cap and is embedded with MiniLM. At a slow query, cosine retrieval selects up to five summaries; up to eight earlier keyframes complement the current fifty-second window. Fast queries use the recent five seconds. Both paths have a 256-token cap. Memory updates use observed video independently of generated query answers. Each policy starts from empty state on each video and performs every scheduled update before the final query; this gives 2,564 updates per memory-enabled run across the manifest.

Execution configuration. Qwen2.5-VL-72B runs in bf16 with four-way tensor parallelism on four GPUs identified by the driver as NVIDIA L20X. Inference requests execute serially at batch size one. PyAV decoding uses sixteen AUTO threads; causal frames use a 320-pixel long edge and JPEG quality 85. The context limit is 16,384 tokens; prefix caching and the multimodal processor cache are disabled. Policies follow a fixed seed–video hash order. The frozen execution profile is shared across all completed runs, and all sampled frames are bounded by their query or update timestamp.

Table 29: Full-video replay with Qwen2.5-VL-72B (bf16): 1,654 questions, 326 videos, three fixed seeds. Accuracy and slow-call rate are percentages; service costs are seconds per query. Memory policies perform 2,564 updates per run. The random policy samples at 25%; the learned policy uses frozen validation thresholds.

Policy	Accuracy	Slow calls	Query	Memory	Total
Always fast	53.75	0.00	0.55	0.00	0.55
Always memory	53.63	100.00	1.81	7.81	9.63
Random	53.53	25.74	0.86	7.75	8.61
Gain router	54.15	13.18	0.71	7.69	8.40

Service and response time. Each event's measured service time includes its decoding, CPU routing/retrieval or embedding, execution coordination, and model request. Let Q_v denote queries and M_v scheduled memory updates for video v. Full-stream amortized service cost is

$$C_\pi = \frac{\sum_v \left(\sum_{i \in Q_v} T_{i,\pi} + \sum_{j \in M_v} U_{j,\pi} \right)}{\sum_v |Q_v|}, \tag{7}$$

where T and U are query and memory-update service times. Table 29 decomposes this sum. For event-time response latency, events arrive at their annotated video timestamps. A single-server queue uses measured service durations: event k finishes at $f_k = \max(a_k, f_{k-1}) + s_k$, and query latency is $f_k - a_k$. Ties process memory updates before queries; video boundaries reset the queue. Table 30 reports the resulting response times separately from query service.

Table 30: Replay timing by policy and seed, in seconds. Response includes event-time queueing from measured service durations. Each row contains the same 1,654 questions; quantiles are computed within each run.

Policy	Seed	Query service P99	Response P50	Response P99	Queue wait P99
Always fast	17	1.94	0.35	1.94	0.00
Always fast	29	1.90	0.35	1.92	0.00
Always fast	43	1.62	0.35	1.63	0.00
Always memory	17	4.02	1.69	7.41	4.79
Always memory	29	3.94	1.68	7.35	4.78
Always memory	43	5.31	1.69	18.83	14.71
Gain router	17	2.46	0.38	5.68	4.91
Gain router	29	2.82	0.57	5.58	4.73
Gain router	43	2.81	0.50	87.54	84.10
Random	17	3.32	0.60	6.14	4.79
Random	29	3.34	0.61	6.08	4.66
Random	43	3.63	0.60	20.35	19.48

The seed-43 router run has a query-service P99 of 2.81 seconds and an event-time response P99 of 87.54 seconds. Its 84.10-second P99 queue wait locates the long tail in queued work. Always-memory and random also show larger tails in this seed. The decomposition makes these scheduling effects visible even when median query service remains short.

Joint video uncertainty. We first average each video's paired outcomes and service totals over the three fixed seeds, then draw 5,000 video-bootstrap replicates with all policies and events retained together. Each replicate recomputes accuracy, cost per query, and memory's fraction of total service from its sampled numerators and denominators. The sample size remains 1,654 questions and 326 videos. Routed total service is 8.40 seconds per query (95% CI: 7.32–9.65), compared with 9.63 (8.52–10.85) for always-memory and 0.55 (0.51–0.59) for always-fast. Memory contributes 91.59% of routed service (90.19–92.69%).

Paired router accuracy differences are $+0.40$ points against always-fast (95% CI: -0.38 to 1.22), $+0.52$ against always-memory (-1.61 to 2.63), and $+0.62$ against random (-0.39 to 1.57). Mean query service changes by $+0.16$ seconds against always-fast (0.12–0.21), -1.10 against always-memory (-1.17 to -1.03), and -0.15 against random (-0.21 to -0.10). Mean event-time response

changes by $+1.06$ seconds against always-fast (0.33–2.55), -0.59 against always-memory (-1.37 to 0.88), and $+0.19$ against random (-0.95 to 1.72). The cost decomposition and timing distribution describe complementary properties of each policy.

G Open-Answer Evaluation and Format Controls

The open set contains 590 visual, 530 action/event, 500 memory, 190 causal, 130 predictive, 130 localization, and 120 verification questions. Its 960 videos comprise 390 from OVBench, 300 from CG-Bench, and 270 from StreamingBench. There are 820 questions from OVBench videos, 770 from CG-Bench, and 600 from StreamingBench. All seventeen models produce fast and slow answers on every question, yielding 74,460 outputs. One frozen rubric compares each generated answer with its human-authored reference in the context of the question. InternVL3.5-38B provides the primary judgments; Qwen2.5-VL-72B independently scores all 74,460 outputs with the same rubric. Table 31 reports the paired model results.

Table 31: All seventeen models on 2,190 human-authored open-answer questions. Correctness uses the uniform InternVL3.5-38B judge. All quantities are percentages except net gain in points. Rescue and harm use the same model's paired fast/slow answers.

Model	Fast	Slow	Gain	Rescue	Harm	Oracle
GLM-4.6V-Flash	17.81	35.16	+17.35	19.18	1.83	36.99
InternVL3-14B	23.74	38.81	+15.07	17.81	2.74	41.55
InternVL3-38B	32.88	43.38	+10.50	15.07	4.57	47.95
InternVL3-8B	29.22	38.36	+9.13	12.79	3.65	42.01
InternVL3.5-14B	23.29	36.99	+13.70	15.98	2.28	39.27
InternVL3.5-38B	26.48	43.38	+16.89	18.72	1.83	45.21
InternVL3.5-8B	14.61	34.70	+20.09	22.83	2.74	37.44
LLaVA-NeXT-Video	1.83	8.22	+6.39	7.31	0.91	9.13
MiniCPM-2.6-8B	16.44	27.85	+11.42	13.70	2.28	30.14
Qwen2.5-VL-32B	26.03	38.81	+12.79	14.61	1.83	40.64
Qwen2.5-VL-7B	22.83	32.42	+9.59	12.79	3.20	35.62
Qwen3-VL-32B	23.74	47.49	+23.74	23.74	0.00	47.49
Qwen3-VL-8B	26.03	42.01	+15.98	19.18	3.20	45.21
Qwen3.5-27B	23.29	44.29	+21.00	21.92	0.91	45.21
Qwen3.5-35B-A3B	28.77	47.03	+18.26	21.46	3.20	50.23
Qwen3.5-9B	21.00	39.73	+18.72	20.55	1.83	41.55
Qwen3.6-27B	15.53	47.03	+31.51	31.51	0.00	47.03

Pathway gains across models. Under the primary InternVL3.5-38B judge, fast/slow accuracy over all models and questions is 21.97/37.98%, a gain of 16.01 points (95% paired video-bootstrap CI: 12.99–19.25). Across models, rescue and harm are 18.18% and 2.18%. Gain is 19.93 points on D_2 and 10.69 on D_1; their difference is 9.25 points (95% CI: 4.03–14.45). Resampling retains all models and pathways within each selected video.

Demand and source slices. Across models, fast/slow accuracy is 29.66/40.35% for the 930 D_1 questions and 16.29/36.23% for the 1,260 D_2 questions. By source, it is 14.67/27.88% for CG-Bench, 29.48/50.22% for OVBench, and 21.08/34.22% for StreamingBench. The evaluation metadata retain source, category, and demand labels, allowing joint recomputation of these slices.

Judge sensitivity. Qwen2.5-VL-72B judging yields fast/slow accuracy across models of 23.34/37.20%, a gain of 13.86 points (95% video-bootstrap CI: 10.94–17.01), and positive gains for all seventeen models. The D_2–D_1 gain difference is 7.71 points (CI: 2.04–13.13). Both judges preserve the direction of the pathway and demand comparisons while differing in their absolute scores.

G.1 Blind Human Audit of Open Answers

Two annotators independently score 6,000 anonymous outputs: 3,000 fast/slow model–question pairs drawn from 1,680 questions, 930 videos, and all seventeen models. Sampling balances models (170 or 180 pairs each), uses seed 20260918, and does not inspect outcomes. The interface hides model identity, pathway, and automated scores; annotators judge semantic correctness from the answer together with its accompanying reasoning.

Independent agreement and adjudication. The audit comprises 12,000 independent judgments. Before adjudication, the raters agree on 1,900 correct and 3,540 incorrect outputs; 500 outputs receive opposing binary judgments, and 60 receive at least one uncertain judgment. Across all 6,000 outputs, categorical agreement is 91.00% and Cohen's $\kappa = 0.809$, counting uncertainty as a third response. On the 5,940 outputs with two binary judgments, agreement is 91.58% and $\kappa = 0.819$. Adjudication resolves the 560 remaining outputs as 540 correct and 20 incorrect, yielding a complete human reference of 2,440 correct and 3,560 incorrect outputs.

Human-scored pathway gains. Under the adjudicated reference, fast accuracy is 33.33% and slow accuracy is 48.00% over all 3,000 pairs, a gain of 14.67 points (95% video-bootstrap CI: 8.47–21.04). The slow path rescues 560 pairs and harms 120, for a net of 440 additional correct answers. Each annotator also finds a positive gain before adjudication (Table 32). On the same 3,000 pairs, the two automated judges find gains of 12.00 and 10.00 points. Excluding the ten pairs with a flagged video/query-time mismatch gives a human gain of 14.38 points (8.30–20.63) on 2,990 pairs across 920 videos.

Table 32: Human-scored fast/slow accuracy and paired gains on the open-answer audit sample. The adjudicated row uses all 3,000 pairs; each annotator row uses that annotator's pairs with two binary judgments. Accuracy is in percent and gains in points; intervals use 5,000 video-bootstrap replicates. All three references find a positive slow-path gain.

Human reference	Pairs	Videos	Fast	Slow	Gain [95% CI]
Annotator 1	2,960	920	33.11	47.30	+14.19 [8.27, 20.28]
Annotator 2	2,990	920	27.76	35.79	+8.03 [3.19, 12.98]
Adjudicated	3,000	930	33.33	48.00	+14.67 [8.47, 21.04]

Automated judge calibration. Against the full adjudicated reference, the InternVL3.5-38B and Qwen2.5-VL-72B judges agree on 91.00% and 90.67% of outputs, with Cohen's $\kappa = 0.807$ and 0.800 (Table 33). Their errors are predominantly false negatives: they reject 530 and 520 human-correct answers but accept only 10 and 40 human-incorrect ones. Automated accuracy is therefore 8.67 and 8.00 points below human accuracy, and the judges understate the paired pathway gain by 2.67 points (95% CI: -1.00 to 6.27) and 4.67 points (1.01–8.41). The automated scores are thus conservative: they are stricter than humans in accepting semantically valid answers, and if anything they understate the advantage of slow execution.

Table 33: Judge–human correctness agreement and signed accuracy bias (automated minus human). N counts individual outputs; bias and its paired video-bootstrap interval are in points. The adjudicated reference covers all 6,000 outputs. Negative bias indicates that the judge accepts fewer answers than humans.

Human reference	Judge	N	Agreement (%)	κ	Bias [95% CI]
Annotator 1	InternVL3.5-38B	5,940	91.58	0.819	−8.08 [−11.43, −5.07]
Annotator 1	Qwen2.5-VL-72B	5,940	91.25	0.813	−7.41 [−10.93, −4.11]
Annotator 2	InternVL3.5-38B	5,980	94.98	0.885	+0.33 [−2.13, 3.14]
Annotator 2	Qwen2.5-VL-72B	5,980	94.65	0.878	+1.00 [−1.67, 3.86]
Adjudicated	InternVL3.5-38B	6,000	91.00	0.807	−8.67 [−12.25, −5.37]
Adjudicated	Qwen2.5-VL-72B	6,000	90.67	0.800	−8.00 [−11.64, −4.58]

Table 34: Paired answer-format controls on the full 4,052-question rewrite pool (1,264 videos). Text-only, 5s, and 50s conditions share question text. Open answers use the uniform InternVL3.5-38B judge. Accuracy is in percent; context gain is in points.

Model	Answer format	Text-only	5s	50s	Gain
Qwen3.5-35B	Multiple choice	42.42	56.64	56.42	−0.22
	Open answer	12.07	26.31	28.90	+2.59
Qwen2.5-VL-72B	Multiple choice	40.05	54.74	56.19	+1.46
	Open answer	14.44	24.26	27.74	+3.48

G.2 Paired Format Controls

Paired format intervention. The pool contains all 4,052 core multiple-choice questions over 1,264 videos. A frozen rewrite model sees the choices without a correct-answer indicator and produces one self-contained question. Fresh bf16 vLLM runs answer that same text with and without options under text-only, five-second, and fifty-second direct conditions. Visual input uses causal PyAV frames at 1 fps, resized to a 320-pixel long edge. All 48,624 answers remain in the denominator, including empty answers; all 24,312 open answers receive the uniform InternVL3.5-38B rubric. Video bootstrap jointly retains formats and windows.

The 35B context gain is -0.22 points for multiple choice (95% CI: -1.51–1.11) and $+2.59$ for open answers (1.44–3.75); the 72B gains are $+1.46$ (0.07–2.86) and $+3.48$ (2.37–4.59). In open generation, 35B rescues 8.66% of questions and harms 6.07%, while 72B rescues 7.90% and harms 4.42%. These transitions explain the modest net improvement despite substantial changes at the individual-question level. The paired open-minus-MCQ context-gain differences are 2.81 points for 35B (95% video-bootstrap CI: 1.31–4.30) and 2.02 for 72B (0.39–3.65).

Visual inputs improve on text-only accuracy in every format and window. The 50-second open-answer gains over text-only are 16.83 points for 35B and 13.30 for 72B; the corresponding paired 95% intervals are 15.20–18.54 and 11.93–14.71. Table 34 reports all absolute operating points.

H Source Transfer and Temporal Diagnostics

Table 35: Source-held-out Qwen3.5-35B long-direct routing. Training uses the other three sources; random matches the held-out escalation rate.

Held-out source	Realized slow rate (%)	Shared gain minus random (pp)
CG-Bench	16.0	+0.37
OVBench	1.6	+0.00
OVO-Bench	56.2	−0.84
StreamingBench	14.0	−0.73
Macro gain		−0.30

Table 36: SVBench adaptation: 7,734 questions, 197 videos, with five-/fifty-second observed windows. Three-seed mean accuracies (%); random matches each frozen policy's realized escalation rate.

Model	Fast	Slow	Random	Routed	Oracle
Qwen3.5-35B	34.03	36.10	34.25	34.28	45.71
InternVL3.5-38B	24.64	27.05	24.89	25.10	36.80

External transfer protocol. The available SVBench English test release (Yang et al., 2025b) contains 7,734 questions from 197 videos. Queries use their annotated segment end and preceding question text, omitting prior answers. Five- and fifty-second windows follow the paired-format PyAV/bf16 setup. Qwen2.5-VL-72B scores all 30,936 outputs with the frozen rubric. ID and fingerprint checks yield no overlap flags.

Frozen allocation and uncertainty. Three core-trained shared-gain policies fix thresholds at a 25% validation budget and share question-only scores across backbones. External rates are 4.37/10.29/16.49% (seeds 17/29/43; mean 10.38%). Uncertainty uses 5,000 video-cluster draws, preserving paired outputs and all seed policies. Matched-rate random accuracy and the three-seed mean are recomputed within each draw.

For Qwen3.5-35B, extending context gives 2.07 points (95% CI: 0.88–3.21), comprising 11.68% rescue and 9.61% harm. InternVL3.5-38B gains 2.40 points (0.99–3.94), with 12.15% rescue and 9.75% harm. Seed-specific routed-minus-random changes are 0.01/0.05/0.03 points for 35B and 0.18/0.02/0.41 for InternVL3.5-38B; the joint three-seed routed-minus-random differences are 0.03 points for 35B (95% video-bootstrap CI: -0.17 to 0.24) and 0.20 for InternVL3.5-38B (-0.03 to 0.43). These intervals include zero for both models.

Rank changes. Across the six core models' independent memory-enabled outputs, the rank ordering differs across capabilities. Relative to aggregate accuracy, temporal-localization accuracy has 7/15 pairwise inversions (Spearman correlation 0.09), clue-interval IoU has 12/15 (-0.77), and the updated 3,477-question D_2 slice has 2/15 (0.89). Multiple-choice-only accuracy is more aligned with the aggregate, with 1/15 inversions (0.94). Clue IoU uses all 819 interval references, assigning zero to missing predicted intervals.

Proactive triggering. The 869 typed proactive questions contain 567 answer-positive and 302 silence-negative timestamps. Gemini-3-flash has precision 0.889, recall 0.707, Trigger-F1 0.788, and false-alarm rate 0.166. Claude-Sonnet-4.5 has precision 0.911, recall 0.127, F1 0.223, and false-alarm rate 0.023. LLaVA-NeXT-Video answers on every question, obtaining recall 1.000 and F1 0.790 with false-alarm rate 1.000. These operating points distinguish silence, responsive answering, and indiscriminate emission, motivating joint reporting of false alarms and Trigger-F1.

H.1 Complete Reference-System Temporal Results

The reference evaluation reports proactive response, evidence grounding, and occurrence counting separately. Tables 37–39 report the reference systems under these complementary metrics.

Table 37: Reference proactive triggering on 869 answer/silence questions (567 answer-positive, 302 silence-negative). False-alarm rate is emission on a silence-negative question; emission rate covers all questions. LLaVA-NeXT-Video emits at every timestamp.

Model	Precision	Recall	F1	False alarm	Emission
Closed-Source Models					
Claude-Sonnet-4.5	0.911	0.127	0.223	0.023	0.091
GPT-4o	0.800	0.601	0.687	0.281	0.490
GPT-5.2	0.831	0.640	0.723	0.245	0.503
Gemini-3-flash	0.889	0.707	0.788	0.166	0.519
Open-Source Models					
MiniCPM-2.6	0.732	0.637	0.681	0.437	0.567
Qwen2.5-VL-7B	0.827	0.515	0.635	0.202	0.406
InternVL3.5-8B	0.782	0.600	0.679	0.315	0.501
LLaVA-NeXT-Video	0.652	1.000	0.790	1.000	1.000

I Benchmark Positioning and Diagnostic Value

The evaluation target. DualStreamBench measures whether a streaming system spends additional computation on the questions that benefit from it. Each timestamped question fixes the observed video prefix and the reference answer, and paired resource configurations expose how correctness changes when a resource is added. The evaluation therefore connects three decisions that answer-only benchmarks leave implicit: which questions to escalate, when to respond, and how much persistent state to maintain. Table 40 relates this target to established video-understanding and routing benchmarks. DualStreamBench does not replace these benchmarks; it reuses the task coverage of four of them and adds the paired diagnostic in the right-hand column.

Table 38: Reference evidence grounding on the 819 clue questions. Mean IoU and threshold rates condition on evaluable, non-empty predicted intervals. The full-core rank diagnostics above assign zero IoU to missing intervals. Threshold rates are percentages.

Model	**Mean IoU**	**IoU** ≥ 0.3	**IoU** ≥ 0.5
Closed-Source Models			
Claude-Sonnet-4.5	0.220	30.4	18.0
GPT-4o	0.215	28.9	17.1
GPT-5.2	0.232	31.1	15.0
Gemini-3-flash	0.294	39.4	25.2
Open-Source Models			
MiniCPM-2.6	0.194	24.1	14.4
Qwen2.5-VL-7B	0.141	18.9	6.1
InternVL3.5-8B	0.093	11.6	5.1
LLaVA-NeXT-Video	0.008	1.0	0.9

Table 39: Reference occurrence counting on 698 questions. MAE is the mean absolute count error; Exact is routed exact-count accuracy (%).

Model	**Fast MAE**	**Slow MAE**	**Routed MAE**	**Exact**
MiniCPM-2.6	2.34	4.26	2.50	16.2
Qwen2.5-VL-7B	2.73	2.43	2.43	20.5
InternVL3.5-8B	2.56	2.12	2.12	23.4
LLaVA-NeXT-Video	3.81	3.73	3.73	17.5

Table 40: Evaluation targets of closely related benchmarks (middle) and the additional question DUALSTREAMBENCH asks in each setting (right). The first four benchmarks contribute source questions; SVBench is used for external transfer; StreamReady and MMR-Bench are the closest prior work on response timing and budgeted routing.

Benchmark	Established evaluation target	Paired diagnostic in DUALSTREAMBENCH
StreamingBench (Lin et al., 2024)	Real-time visual, omni-source, and contextual understanding at successive query times.	Which timestamped questions benefit from a larger inference resource budget?
OVBench (Huang et al., 2025b)	Online perception, memory, and reasoning across past, present, and future contexts.	How do individual answers change when historical access is added?
CG-Bench (Chen et al., 2025a)	Answer correctness together with retrieval of supporting temporal clues.	How do paired answer outcomes and evidence localization distinguish model capabilities?
OVO-Bench (Niu et al., 2025)	Backward tracing, real-time perception, and forward active responding.	How does escalation interact with timestamp-specific correctness and response control?
SVBench (Yang et al., 2025b)	Sustained temporal understanding through linked multi-turn question–answer chains.	Do resource gains and frozen allocation policies transfer across streaming sources?
ProReady-QA / StreamReady (Azad et al., 2026)	Evidence readiness and the correctness and timing of proactive answers.	Which resources change an answer, and what query and state-maintenance costs are incurred?
MMR-Bench (Ma et al., 2026)	Budget-aware multimodal model selection under fixed candidate sets and cost models.	How does allocation behave with causal video prefixes, temporal interaction, and persistent state?

Task demand and resource utility answer different questions. The operating modes, semantic taxonomy, and human demand labels describe what evidence and operations a question requires, independent of any model. Paired executions determine whether a particular model can turn an added resource into a better answer. The two need not coincide: a question can require historical evidence while a model fails to retrieve or use it, as the opposite-signed memory contrasts in Table 8 show. Keeping both measurements makes it possible to diagnose resource utility within a capability or demand level, and to test whether an allocation policy predicts that utility on unseen videos.

Similar answer accuracy can hide different escalation behavior. On the same 5,619 questions, the memory-enabled slow paths of Qwen3.5-35B and Qwen2.5-VL-72B reach nearly identical accuracy, 54.69% and 54.00% (Table 12). Their harm rates, however, are 13.10% and 6.74%: a 0.69-point accuracy difference accompanies an almost twofold difference in how often each slow path breaks its own fast answer. An aggregate leaderboard would rank the two systems as equivalent; paired outcomes show that a selective policy must protect far more correct answers for the first. They identify the questions a policy should preserve as well as those it should rescue. Figure 10 illustrates both transitions with human-verified answers from one backbone.

Fixed endpoints do not determine allocation quality. All policies in the 35B budget comparison share the same fast and long-direct predictions, so they differ only in which questions they escalate. At 25% escalation, random allocation reaches 50.28%, shared-gain routing 50.15%, and hashed-rescue routing 53.32% (Figure 8). The 3.04-point spread therefore measures selection quality alone, with answer quality held fixed. Frozen-threshold evaluation asks a further question: whether the selection rule and its operating budget transfer to an arriving stream from a new source. The external results in Appendix H separate this allocation behavior from the gains of the underlying slow answers, which do transfer.

Per-question savings and complete-stream savings differ. The replay study links answer selection to the work needed to preserve earlier observations. Relative to always-memory execution, the router reduces query service by 61.00%, but total service falls by only 12.74% once scheduled memory updates are included (Table 29). A benchmark that timed queries alone would report a roughly fivefold larger saving. This decomposition makes state-update policies an explicit part of the evaluation. Reporting paired answer outcomes, temporal behavior, and complete-prefix cost together gives a system-level view of adaptive streaming competence.



Figure 10: Human-verified rescue and harm for InternVL3-8B on two open-answer questions, with original video frames at or before each query. At 816 s, the fast answer incorrectly declares first aid complete and the slow answer matches the reference. At 37 s in a different video, the fast answer correctly counts two horses and the slow answer changes the count to one. Both independent annotators agree on all four correctness judgments. Questions, query times, references, and displayed answers are taken from the evaluated records.