Q-SPAN: QUERY-ADAPTIVE LOOK-BACK WINDOWS FOR STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

What a streaming video assistant can answer is bounded by how much history it keeps, but decided by which slice of that history the model is actually shown, and that slice cannot be fixed. Measuring accuracy against the look-back window on three streaming benchmarks, we find the window a question needs varies at least $64\times$ in length, and that populations disagree about it as sharply *within* a benchmark as between them: on one level of OVO-S-Bench a $4\,\mathrm{s}$ window beats a $256\,\mathrm{s}$ one by 13 points, and on another the longer window wins by 15. No single window serves a stream of mixed questions. We propose Q-SPAN, which reads the question and nothing else to choose the span. Which of three windows a question calls for, a close look at the present, a recent span, or a thin pass over the whole prefix, is settled by its wording, so one text call picks the window before a frame is decoded. How far back the evidence sits is not settled by the wording: the same sentence can sit three seconds from its answer in one recording and three minutes in another. That residue is left to the frames, asked in one further call whether the ones in hand suffice. The method is training-free and its rules name no benchmark's templates, though written for the kinds of question streaming benchmarks ask: planner, probe and answering model are one frozen backbone under different prompts. Across three backbones and three benchmarks it holds both ends at once, which no fixed window does, and the gain follows the mixture: three points where a benchmark spreads its questions, and a tie where they do not.

1 INTRODUCTION

A streaming assistant watches a video that never stops growing, and is asked questions about it at arbitrary moments (Li et al., 2025b; Lin et al., 2024; Huang et al., 2025). The evidence for any one of those questions is usually a few seconds long. Everything the assistant has seen cannot go to the model, so something has to pick a subset of frames, and most work makes that a memory problem: what to keep as the stream arrives and what to fetch back once a question comes. Recent systems compress the visual cache (Xu et al., 2026), build hierarchical or event-structured memories over the prefix (Liang et al., 2026), or retrieve into it at query time (Ning et al., 2025).

Our decision comes after that one. Whatever a system has kept, something still has to settle how much of it to put in front of the model, and the question settles more of that than it appears to. *"What colour is the mug on the desk?"* has to be answered from the current instant: the object it names must be in view for it to mean anything, and no amount of earlier footage helps. *"How many times has the door opened?"* is at the other extreme and is answered only by covering the whole prefix. Those are different regimes of looking, and the wording picks between them. But *"What did they do after taking the lid off the pot?"* shows where the wording stops: it fixes a reference event and says nothing whatever about when that event occurred, and the interval between that event and now belongs to the recording, not to the sentence. So we ask each source for what only it knows. The regime comes from the text, in one call that decodes no frame, and the part the text cannot supply is put to the frames, in one further call asking whether they hold what is needed. Both calls are the answering backbone under different prompts, so neither costs training or extra weights. Figure 1 is the evidence that this matters. Across three streaming benchmarks and eleven question populations within them, every population has a clear optimum, those optima span at least $64\times$, and they disagree inside single benchmarks as much as across them; a system forced to pick one

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.

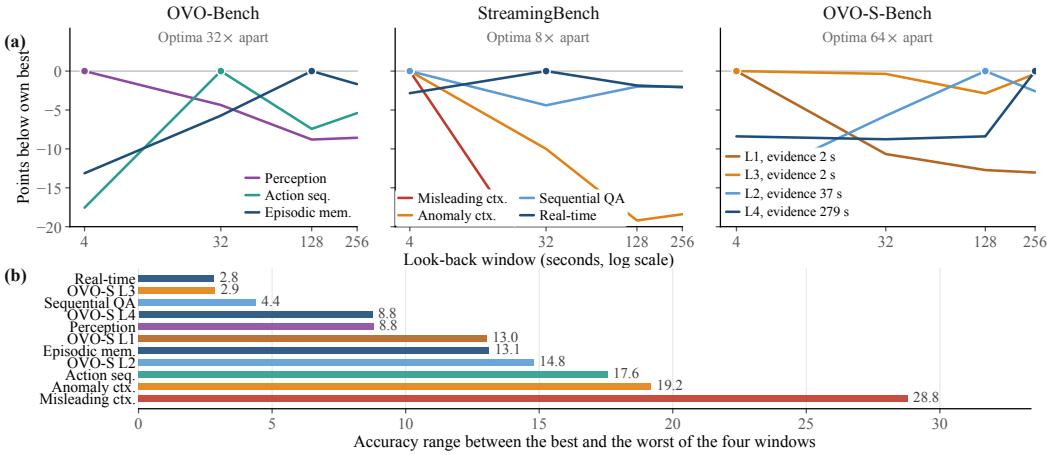

Figure 1: **No single look-back window serves streaming video understanding.** One panel per benchmark, one curve per question population; OVO-S-Bench's levels carry their median evidence distance. (a) Accuracy relative to each curve's own best, so what the panels compare is where a curve peaks and how far it falls, not one population's level against another's. The filled marker is its best window; a hollow one, where the two differ, is the narrowest window within two points of it. (b) The range each population covers between its best and worst window, which needs no compromise window to be chosen and so does not depend on how the populations are counted.

window pays the gap on every population it does not suit, which runs to 28.8 points between one population's best window and its worst. Our contributions:

- A measurement of how far back each kind of question needs to look, which establishes that no fixed sampling policy is adequate and locates the optima the method then has to predict.
- A distinction that makes the choice well posed: which *kind* of evidence a question refers to is categorical and carried by its wording, while how far back it lies is metric and belongs to the recording, so only the first is predictable from it. Q-SPAN follows it: one text call, no frame decoded, chooses which of three windows a question calls for, and one visual call on frames already decoded buys what the wording cannot settle. The same three windows serve every benchmark and backbone, so the planner holds no per-benchmark quantity.
- Gains over the published training-free policies on every benchmark, with the answering model frozen and one plan serving backbones from 7B to 27B. Against any window fixed in advance the gain averages at least 2.8 points across three benchmarks and three backbones; against the strongest window chosen in hindsight it is 1.2 to 4.9 where questions differ in how far back they reach and a tie whose interval straddles zero where they do not.

2 THE OPTIMAL LOOK-BACK WINDOW IS NOT A CONSTANT

Accuracy against the look-back window, the interval ending at the query that frames are drawn from, has a different optimum for every question population, spanning at least 64× (Figure 1a). They disagree within single benchmarks as much as across them. On OVO-Bench perception wants 4 s while action sequences want 32 and episodic memory 128; on StreamingBench it reverses: real-time questions peak at 32 s, the three contextual subsets at 4, misleading context losing 26.0 points over that step. A fixed window therefore concedes accuracy on some population: between best and worst window the least sensitive moves 2.8 points and the most sensitive 28.8 (Figure 1b). The fall is not the backbone mistaking which frame is the present; telling it which one is leaves it unchanged (App. §A). Within a curve only the window varies. App. §A gives the settings and runs. OVO-S-Bench annotates each question's evidence distance, and its four levels pull in opposite directions: L1 loses 13.0 points as the window grows while L2 gains 14.8, so their average barely moves and a sweep calls the axis inert. They move apart at a fixed frame count too (App. §B.15). The annotation gives only half of what a policy needs. L1 and L3 share a median distance of 2 s, yet L1 falls steeply while L3 is the flattest in the figure, 2.9 points against L1's 13.0. L4 is clearest: its evidence sits a median 278.7 s back, so 128 s buys nothing over 4 and only 256 s reaches it, worth 8.4 points.

The distance fixes the shortest window that can contain the evidence, not the price of exceeding it, the larger effect here. The windows here are lengths; the ones the method deploys come from prior work rather than from this figure, and the widest is a frame budget rather than a length, which is why Sec. 4 defines it and this axis omits it. The curves establish only that the choice matters.

3 Related work

We organise prior work on streaming video understanding by *what* its frame selection is over and *when it is made*, since that is what separates our method from work it otherwise resembles.

Deciding what to keep, before the question. The largest group chooses which arriving frames survive. Some keep reduced content: Flash-VStream (Zhang et al., 2025a) compresses history into a bounded store, StreamForest (Yan et al., 2025), OASIS (Liang et al., 2026) and FluxMem (Xie et al., 2026) maintain event-structured or hierarchical memories, VideoStreaming (Qian et al., 2024) summarises the stream into a recurrent state, and SAVEMem (Wu et al., 2026) varies retention by semantic content. Others keep the frames and shrink them, compressing or evicting the key-value cache (Chen et al., 2026; Xu et al., 2026; Yang et al., 2025), dropping the tokens a new frame duplicates (Yao et al., 2025), organising the cache hierarchically (Zhang et al., 2026), or holding it to a fixed budget (Kim et al., 2025). A third decides when to respond as much as what to store (Chen et al., 2024a; Qian et al., 2025). All of it happens before the question exists, so retention cannot be conditioned on it. Some add a question-conditioned window afterwards, as LiveVLM (Ning et al., 2025) does when it retrieves into the cache at query time, but that window can only reach what retention kept. This is a line we build on rather than compete with: what a memory retains bounds what any reader can reach, and what it returns can stand in for our widest window. What it does not decide is how much of what it holds to show, and raising the ceiling is not making that choice. Substituting retrieved event nodes for the uniform pass costs 1.2 points (Table 5): narrowing to a couple of nodes gives up more coverage than it gains density.

Deciding what to read, by looking at it. A second conditions on the question but works on a complete video, not a stream. Frame-Voyager (Yu et al., 2025) ranks frame combinations for a query, FOCUS (Zhu et al., 2025) selects keyframes, an MLLM can serve directly as a frame selector (Hu et al., 2025), and temporal grounding localises a described moment (Lei et al., 2021). The offline setting has two consequences. Scoring candidates means the selector sees the video, at a cost that grows with its length, whereas ours are made from the question text with no frame decoded. And with the whole video in hand there is no query timestamp, so nothing for a window to end at: they choose *which* frames, never how much of a growing past to spend them on. The two compose, since a selector can operate *inside* the window we choose.

Deciding how much to compute, after answering once. A third allocates computation from a signal the model produces: confidence cascades escalate when an early attempt looks unreliable (Narasimhan et al., 2025), and OASIS retrieves when its short-context answer is uncertain. Escalating after a first pass pays for a look before knowing whether it was the right one, and can then spend only *more* of the same. Our first window comes from the question, so most questions never escalate, and our one confidence-like component, the probe, comes last, reuses frames the short window already decoded, and widens it by at most one step.

Where our decision sits. None of these lines decides, from the question and before a frame is decoded, how much of the prefix to put in front of the model, and Sec. 2 is the evidence that the choice is worth making: what it should be is neither a constant to tune nor a property of the stream. A fourth line decides something else again, whether to answer at this moment at all (Wang et al., 2025c;b; Zhang et al., 2025b; Azad et al., 2026; Chen et al., 2025); that axis is orthogonal to ours and we do not compare against it. Our decision also presupposes an evaluation that poses questions at annotated timestamps over a growing prefix, which is the minimum condition for a look-back window to mean anything: offline long-video benchmarks (Mangalam et al., 2023; Wang et al., 2025a) hand over the whole video, leaving no timestamp for a window to end at.

4 Method

A streaming assistant watches a video as it arrives. At some time t a user asks it a question, and the answer has to come out of the prefix it has seen so far. Something has to decide which of those

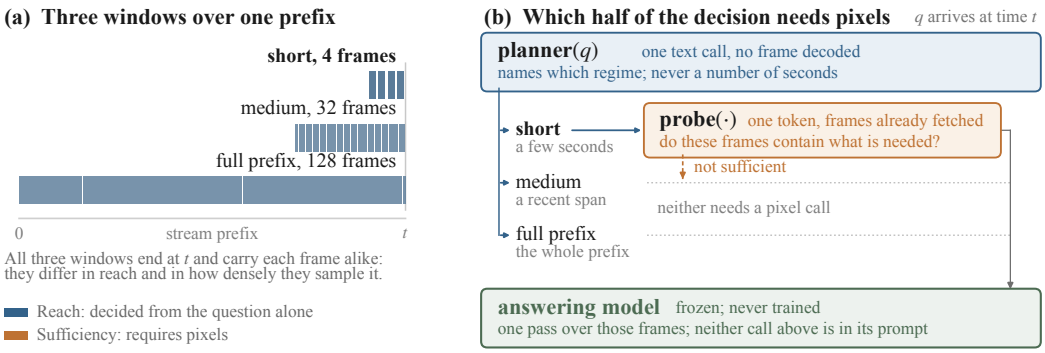

Figure 2: **Q-SPAN.** (a) The three windows of Algorithm 1 over one prefix, all ending at the query time t. Every frame is carried identically, so what separates them is how far back they reach and how densely they sample it; spans are ordered rather than to scale. (b) One text call, given the question and no frames, chooses the regime. That much the wording settles; how far back the evidence sits it does not, so the residue is put to frames the answer will use anyway, and its only power is to widen the short window one step.

frames the model receives, and Q-SPAN decides it in two calls that the answering model never sees, both of them that same frozen backbone under a different prompt, so nothing here is trained and no second model is loaded. Figure 2(b) draws them. The first, the planner, reads the question with no frame decoded and chooses one of three windows over the prefix. The second, the probe, runs only when the planner named the shortest window: it asks whether those frames hold what the question needs, and if they do not the window widens one step. The answering model then runs once over the frames that survive this.

Algorithm 1: Q-SPAN on a question arriving at t. All prompts go to one frozen backbone M.

Input: question q; the prefix seen up to t; frozen backbone M; prompts Plan, Probe, Answer
Output: an answer to q

$r \leftarrow M(\text{Plan}, q)$; `// r ∈ {short, medium, full}, no frame decoded`
if $r = \text{full}$ **and** $t \leq 32\,s$ **then**
  $r \leftarrow \text{medium}$; `// the medium window already covers the prefix`
$F \leftarrow 4$ frames at 1 fps over $[\max(0, t-4), t]$ if short, 32 at 1 fps over $[\max(0, t-32), t]$ if medium, 128 uniformly over $[0, t]$ if full, taking fewer frames rather than padding when the stream is shorter;
if $r = \text{short}$ **and** $M(\text{Probe}, q, F) = \text{insufficient}$ **then**
  $F \leftarrow 32$ frames at 1 fps over $[\max(0, t-32), t]$;
return $M(\text{Answer}, q, F)$;

Algorithm 1 states the procedure and Figure 2(a) draws the three windows, which we also call short, medium and widest. They answer the three kinds of thing a user can ask about: what is in view at that moment, something that just happened, or something much further back. Nothing in the stream announces which has been asked. Choosing before any frame is decoded is the part a stream forces on us: offline, a system can afford to watch a video before deciding how to watch it, while a streaming assistant cannot, since the pass it would spend deciding is the pass it is trying to avoid. The window we decide over ends at t, which is what makes one number enough to determine it: no index over the stream, no annotation, and an answer on every question rather than only on the ones a stored memory happens to cover. What such a window cannot do is reach a moment far back without covering everything in between, which makes the widest of the three also the coarsest. Avoiding that is what memory plus retrieval is for, and it answers where to look rather than how much to look at. The two compose in principle, since what a retriever returns can stand in for our widest window and spend its frames on the material the retriever named instead of the whole prefix. On the retrievers we substituted it does not help (Table 5): narrowing buys density inside the region it names and forfeits the rest, which under a fixed budget is the asymmetric side of the bet. The widest window is a uniform pass because it cannot be badly wrong. Compressing the cache changes what a frame costs, not which stretch of the stream is covered.

4.1 What the question settles, and what it does not

For many questions the wording fixes the regime. *What is the person holding* points at the instant asked: the thing asked about has to be in view for the question to mean anything, so a window spending its frames elsewhere cannot help: the answer is short. *Where did I leave the scissors* points at the last time the object was seen, not the current instant however the recording runs, so the answer is medium. A reader with only the sentence can tell them apart, and so can a text-only call.

How far before is another matter. *What happened after the machine was switched on* fixes a reference event and says nothing about when it occurred; the interval between it and now belongs to the video. Two recordings can pose the identical sentence with its answer three seconds back in one and three minutes in the other, and on OVO-S-Bench they do: of the wordings asked of five or more recordings, one is asked of 34 and its evidence sits anywhere from 9 to 706 seconds back, a spread of $75\times$ under a fixed sentence (App. §B.18). A model mapping such a sentence to a number of seconds is not reading the question. It reports how long ago such things tend to happen in the footage it was tuned on, a fact about that footage rather than about what was asked.

The line is between a categorical property of what a question refers to and a metric property of the video containing it: the first is in the sentence and we read it off, the second is not and we do not predict it. This is also why there are three windows and not a scale: three is the resolution of what the words carry, and a finer partition would be predicting the video from the sentence under another name. A calibrated distribution over reach would need supervision on the quantity this section argues the sentence lacks. The call is asked which of the three, in those terms, and nothing downstream treats it as a duration. One consequence: the middle label means the recent past and its rung reaches 32 seconds, so evidence between that and the whole prefix is covered only if the wording sent the question to the widest window. That costs about half a point (App. §B.16). The prompt is written in the same terms, which is what makes it portable: it describes each window by the property of the evidence it suits, and it names no benchmark's templates and no task labels; the phrasings it does quote, *after X* or *where did I put X*, are ordinary English rather than any benchmark's question forms. It gives 5 rules and 11 examples (App. §D), each keyed to what a question refers to rather than to a template it matches; only one rule, covering the layout of a space and the route through it, calls for the widest window. A list of literal keywords would be tied to the strings it was drawn from, which is why the prompt states its rules in ordinary English rather than as patterns to match.

4.2 What the frames settle

One question the words cannot decide either way remains. *Where is the pillow with the hexagonal pattern* and *where did I put the wood* are near-identical in form, and what separates them is whether the object is still in view, a fact about the frames rather than the words. So once the short window has been fetched, the model is asked over those frames whether they hold what the question needs, as a two-option letter choice with the question's own answers withheld, so the probe cannot become a disguised attempt at it (App. §D). A negative verdict widens it one step, to medium. Nothing else in the system looks at a frame before the answer does. What the probe may do is bounded by its verdict. A yes or no about the frames in front of it cannot name a distance: it says this window lacks the evidence, not that the evidence lies far back, so one step is all it licenses. That one step is also the cheaper of the two undominated operating points, and that is measured rather than assumed.

4.3 Why the auxiliary calls stay outside the answer

Both calls are separate generations, and the answering prompt contains neither. Asking the answering model to reason before it answers, a setting studied by Liu et al. (2026), makes it worse on the same frames, in each of the two prompts we compared, and worst where those frames carry the most (App. §B.4): the wider the window, the more the deliberation displaces, so it substitutes for reading the evidence rather than helping to read it. It also costs latency in the one place a streaming assistant has none to spare, since it lengthens the call the user waits on. Our two auxiliary calls are on that path too, but each returns a label rather than a paragraph, so together they cost tens of milliseconds against the answering call's hundreds (Table 6). It matters for comparison as well: a protocol that reasons by default is trading perception for a different ability, so mixing prompt styles makes the comparison partly one of prompts. Neither call costs a pass over the video either. The text call decodes no frames and the probe reuses frames already fetched for the short window, so neither costs more on an hour of stream than on a minute. One cost this does not remove is storage: sampling the whole prefix requires the prefix to still be addressable, which is a buffer, not a computation. A semantic memory trades that buffer for continuous compute; we trade it the other way.

Table 1: **Main results on OVO-S-Bench, OVO-Bench and StreamingBench.** "—": unreported results. †: results we reproduced. **Bold** and underline are the best and second best within each group.

(a) Comparisons on OVO-S-Bench

Method	$L1_{2.0s}$	$L2_{36.8s}$	$L3_{2.0s}$	$L4_{278.7s}$	**Avg.**
Human, streaming	93.2	81.0	86.4	79.2	86.6
Proprietary					
GPT-5.4	54.6	57.6	50.8	40.5	50.9
Gemini-3.1-Pro	61.9	64.0	55.9	54.9	59.2
Open-source offline MLLMs					
InternVL-3.5-241B	**55.6**	**55.7**	51.6	40.5	50.9
Qwen3.5-397B	49.6	55.4	58.1	45.4	52.1
Qwen3-VL-235B	52.5	55.2	**61.2**	**45.7**	**53.6**
Open-source streaming MLLMs (Training-based)					
Flash-VStream-7B [ICCV'25]	18.7	29.9	22.5	28.7	24.9
StreamForest-7B [NeurIPS'25]	**46.6**	45.2	**49.8**	34.9	**44.1**
VST-7B [ECCV'26]	45.7	44.2	40.9	38.0	42.2
StreamingVLM-7B [ICLR'26]	38.7	**50.5**	41.8	**41.2**	43.0
Training-free streaming MLLMs, frozen Qwen2.5-VL-7B					
[Base] *Qwen2.5-VL-7B*	40.7	45.5	45.9	**44.7**	44.2
InfiniPot-V [NeurIPS'25]	39.1	35.7	41.9	40.6	39.3
StreamingTOM [CVPR'26]	37.2	**48.2**	38.7	33.5	39.4
HERMES (4K tokens) [ACL'26]	40.9	45.4	49.4	42.9	44.6
FluxMem [CVPR'26]	43.0	47.6	45.5	42.6	44.7
SAVEMem† [arXiv'26]	43.6	44.4	48.0	33.2	42.3
SimpleStream† [arXiv'26]	44.5	40.0	48.4	38.3	42.8
Q-SPAN (ours)	**48.8**	47.6	**51.3**	39.4	**46.8**
Training-free streaming MLLMs, frozen Qwen3-VL-8B					
[Base] *Qwen3-VL-8B*	46.1	**55.7**	**54.5**	**43.1**	49.9
SimpleStream† [arXiv'26]	51.4	38.0	53.0	33.2	43.9
OASIS† [CVPR'26]	46.1	50.4	52.7	38.0	46.8
Q-SPAN (ours)	**59.9**	54.3	**54.5**	**43.1**	**52.9**
Training-free streaming MLLMs, frozen Qwen3.8-27B					
[Base] *Qwen3.8-27B*	53.3	57.8	53.8	**51.1**	54.0
Q-SPAN (ours)	**69.6**	**58.9**	**60.6**	**51.1**	**60.1**

(b) Comparisons on OVO-Bench and StreamingBench

	OVO-Bench			SB
Method	**RT Avg.**	EPM	ASI	**RTVU**
Proprietary				
GPT-4o	64.5	57.9	75.7	73.3
Gemini 1.5 Pro	69.3	58.6	76.4	75.7
Open-source offline MLLMs				
InternVL2-8B	60.7	43.1	61.5	63.7
Qwen2-VL-7B	60.7	44.4	**66.9**	69.0
LLaVA-OneVision-7B [TMLR'25]	**64.0**	**54.2**	55.4	**71.1**
Open-source streaming MLLMs (Training-based)				
Dispider-7B [CVPR'25]	54.6	48.5	55.4	67.6
Flash-VStream-7B [ICCV'25]	29.9	36.4	33.8	23.2
TimeChat-Online-7B [ACM MM'25]	61.9	55.9	59.5	75.8
StreamForest-7B [NeurIPS'25]	61.2	**58.9**	**64.9**	77.2
VST-7B [ECCV'26]	**67.2**	56.9	**64.9**	**79.5**
Training-free streaming MLLMs, frozen Qwen2.5-VL-7B				
[Base] *Qwen2.5-VL-7B*	64.6	50.8	56.8	75.4
InfiniPot-V [NeurIPS'25]	65.9	—	—	76.4
FluxMem [CVPR'26]	67.2	—	—	76.4
HERMES (4K tokens) [ACL'26]	69.0	48.5	62.2	79.4
OASIS [CVPR'26]	67.3	51.9	58.8	70.6
Q-SPAN (ours)	**77.6**	**57.9**	**62.8**	**84.9**
Training-free streaming MLLMs, frozen Qwen3-VL-8B				
[Base] *Qwen3-VL-8B*	66.9	58.6	70.3	75.4
HERMES (4K tokens) [ACL'26]	73.3	—	—	81.3
OASIS [CVPR'26]	78.1	62.0	60.1	78.2
Q-SPAN (ours)	**82.4**	**62.3**	**75.7**	**84.0**
Training-free streaming MLLMs, frozen Qwen3.8-27B				
[Base] *Qwen3.8-27B*	71.0	64.3	66.9	81.7
Q-SPAN (ours)	**84.5**	**65.0**	**77.0**	**87.7**

5 EXPERIMENTS

5.1 SETUP

We evaluate on three streaming benchmarks: OVO-S-Bench (Li et al., 2026), OVO-Bench (Li et al., 2025b) and StreamingBench (Lin et al., 2024). The latter two are standard and widely reported, but their questions are weighted towards the present, narrower in mixture than a deployed stream; we include them for comparability with prior work. OVO-S-Bench is built the other way round, covering current-view perception and out-of-view memory in comparable proportion, and annotates every question's evidence interval, so its four levels differ widely in how far back that interval lies, the only ones that separate a gain on recent evidence from one on distant evidence; the level columns of Table 1a carry their medians. It is our primary benchmark. Each benchmark is run with its own official prompt, scorer and task grouping, and every run presents frames as images (App. §A). Three answering backbones: Qwen2.5-VL-7B (Bai et al., 2025a) and Qwen3-VL-8B (Bai et al., 2025b) because prior work reports them, and Qwen3.8-27B (Qwen Team, 2026) so that nothing rests on one generation. The planner and probe use the answering backbone unless stated otherwise, so the system carries one set of weights; Table 3 reports separate planners. The setting is training-free and zero-shot: no component sees labels from any evaluated benchmark, and the eleven demonstrations that make the planner prompt few-shot are written rather than drawn from one. The prompt is one wording, unchanged across every benchmark and backbone (App. §B.17).

5.2 MAIN COMPARISON

Table 1 reports three benchmarks. The training-free entries are grouped by backbone, so differences within a group are down to the mechanism rather than capacity. Accuracy is compared at each system's budget, and ours is not the smallest: routing spends 20 to 60 frames a question against a fixed recent window's four, a trade App. §A and Table 6 price. App. §C gives a full table per benchmark. The tables also quote general-purpose models, proprietary (Hurst et al., 2024; Gemini Team et al., 2024) and open-weight (Chen et al., 2024b; Wang et al., 2024; Li et al., 2025a): they make no frame-selection decision, so they say what the benchmark costs a passive watcher.

OVO-S-Bench. On the newest benchmark, routing a frozen 7B backbone reaches 46.8, the best training-free entry there; on a frozen 8B, 52.9, 3.1 above the bare backbone and at least 6.1 above either system we ran on it; at 27B, 60.1, the highest entry in Table 1a apart from streaming humans,

above Gemini-3.1-Pro (Google DeepMind, 2026) at 59.2 and every open-source entry including a 235B mixture at 53.6. The level columns say where routing's advantage ends. Choosing a span presupposes some idea of where to look, which the wording supplies only when the referent is recent or bounded. L1 and L3 sit a median 2 s from the query and are exactly that case: at 27B routing gains 16.3 and 6.8 over the same backbone under the benchmark's default 128-frame sampling of the prefix. L4 sits a median 278.7 s back with nothing in the wording to locate it, and the gain falls to zero: a span chosen from the question cannot reach evidence it gives no clue about.

OVO-Bench. How much of this benchmark is answered from the present shows in one number: four recent frames reach 82.3 on its real-time tasks, a macro over six perception subsets (App. §B.4). From backward tracing we keep the two subsets that ask about the past, episodic memory (EPM) and action-sequence identification (ASI), and drop hallucination detection (HLD), where a constant answer scores perfectly and what is measured is robustness rather than memory (App. §C). The two families fail in opposite directions, and what separates them is what each keeps. A recent window keeps the last few seconds intact and discards everything else, so it is strong on perception and weak on memory. A memory keeps a compressed trace of everything, and the compression costs it the present: on our backbone both OASIS and HERMES fall *below* the plain recent window on perception, by 4.2 and 9.0 points, and OASIS gains on the backward subsets in exchange. Neither is choosing badly; both pay a standing charge, one in coverage and one in fidelity, because both fix a representation before the question is known, and one that must serve every question serves none at full resolution. Routing fixes a span only once the question is available, so it pays no such charge. This is why the perception margin over the memory rows, 4.3 to 9.1 at 8B and 8.6 to 11.7 at 7B, is not borrowed from the backward subsets, where we match or exceed those same rows.

StreamingBench earns its place here through an internal tension: real-time understanding prefers the medium window while all three contextual subsets prefer the short one, each costing the other (Figure 1). No fixed span serves both, so any system tuned on one subset concedes the other. Routing declines the choice rather than splitting it: 87.7 real-time at 27B and 84.0 at 8B, the top three entries in that column of Table 1b being ours, from a system never told which subset a question came from. The same configuration also takes the best average over the three contextual subsets, by the wider margin of the two (59.1 against 46.9 for OASIS). App. §C reports those subsets in full. They reward looking at *less*, which a system built to retain more has no way to express.

5.3 Ablation studies

No fixed window works at both ends. Table 2 sets the planner beside the three windows it chooses among, on the levels whose reversal Sec. 2 measured. Against the best of the three on each level, every fixed window falls at least 10.66 points short somewhere, while the planner is never more than 3.89 short and exceeds it on L3. It leads the table, but what matters is the absence of a bad level.

Neither the budget nor matching the wording accounts for the gain. Two rows of Table 2 replace the planner and keep everything else. The *shuffle* permutes the planner's own labels across questions, so the share of the three windows and the mean frame count are the planner's exactly, $61/10/29$ and 43 frames, and only which question receives which window is lost. It scores 48.13, below the 49.85 the widest window reaches alone, so spending 43 frames is not what produces the planner's 52.33. The *regex* replaces the planner with literal string patterns transcribed from the rules its prompt states, fixed before they were scored, and calls no model. It reaches 49.22, so the wording does carry the

Table 2: **Ablation studies on the planner.** On OVO-S-Bench, whose evidence lies nearest on L1 and L3, further on L2 and furthest on L4. *Shuffle* permutes the planner's own labels across questions, so the window mix and frame count stay identical and only the pairing is lost; *regex* applies the planner's prompt rules as literal string patterns, calling no model. Default: Qwen3-VL-8B.

	L1	L2	L3	L4	Overall	Avg. frames	short/med/wide (%)
no planner: the same window for every question							
the short window only	**59.94**	39.63	52.69	38.69	47.74	4.0	100/0/0
the medium window only	49.28	48.70	52.33	38.32	47.16	32.0	0/100/0
the widest window only	46.10	**55.74**	54.48	**43.07**	49.85	128.0	0/0/100
routing without a language model							
shuffle, mix matched	54.89	44.99	52.99	39.64	48.13	42.7	61/10/29
regex, rules as patterns	54.85	46.67	53.76	41.61	49.22	30.7	59/25/16
the planner alone	**59.94**	51.85	**54.84**	42.70	**52.33**	42.7	61/10/29

signal, but matching it recovers only about a quarter of the distance from the *shuffle* to the planner: a pattern cannot tell a direction from a past tense, and *left* matches both (App. §B.19).

The planner needs a language model, not a large one. Closing the gap the *regex* leaves needs a model that reads the sentence rather than scans it, and size is not what supplies it: text-only planners from 1.7B to 32B all land between 51.02 and 52.44 in Table 3. The answering backbone planning for itself reaches 52.33, inside that range, adding no weights. At the smallest size it is absent: 0.6B scores about what not routing does, because it hardly routes, naming the short window for 78% and 85% of questions on two benchmarks against 43% and 60% at 1.7B (App. §B.1).

Table 3: **Ablation studies on the probe and planner size.** On OVO-S-Bench and StreamingBench's real-time visual understanding (RTVU) set. *None*: routes nothing, the short window only.

	OVO-S-Bench		StreamingBench RTVU	
Planner (text-only)	Alone	+ *Probe*	Alone	+ *Probe*
None	47.74	48.81 (+1.07)	80.96	82.76 (+1.80)
Qwen3-0.6B	48.68	49.56 (+0.89)	81.44	82.76 (+1.32)
Qwen3-1.7B	51.02	51.20 (+0.18)	**83.68**	84.16 (+0.48)
Qwen3-4B	51.70	51.98 (+0.28)	83.48	84.24 (+0.76)
Qwen3-8B	52.14	52.56 (+0.42)	83.64	**84.32** (+0.68)
Qwen3-32B	**52.44**	**53.13** (+0.69)	83.44	84.28 (+0.84)
Qwen3-VL-8B	52.33	52.94 (+0.60)	82.84	83.96 (+1.12)

What the wording cannot carry, the probe supplies. How far back the evidence sits is a property of the recording rather than of the sentence, so no amount of reading the sentence recovers it. The probe asks the frames instead: in Table 3 it improves every planner it is placed behind, by 0.18 to 0.89 on OVO-S and 0.48 to 1.32 on StreamingBench, and it lifts a bare short window, with no planner at all, by 1.07 and 1.80. On OVO-Bench, where almost every question is about the present and the planner keeps 60% of them at the short window already, it is a tie (Table 25). That it adds behind every planner is the point: what it supplies is orthogonal to how well the wording was read, so a stronger planner does not make it redundant.

The planner's choices track where the evidence is. The planner never sees the evidence annotation, yet its choices follow it: in Table 4 the short window takes 97% of L1 and the widest 80% of L4, and across benchmarks the short window takes 80% of OVO-Bench's real-time (RT) perception tracks against the medium window's 73% of its backward-tracing (BW) ones. StreamingBench, whose questions are largely about the present, draws 76% short and only 9% widest.

Table 4: **Ablation studies on the routing distribution.** Shares within each split, and beneath them the accuracy the planner alone reaches on it.

	OVO-S-Bench					OVO-Bench		SB
Share (%)	L1	L2	L3	L4	Overall	RT	BW	RTVU
short	**97**	40	61	18	61	**80**	23	**76**
medium	1	22	15	3	10	17	**73**	15
widest	2	37	23	**80**	29	3	4	9
Accuracy (%)	59.94	51.85	54.84	42.70	52.33	82.78	63.84	82.84

The choice of the widest window. Table 5 varies the widest window alone, against three alternatives to a uniform pass, each alone and then inside our system with the planner and probe unchanged. *Text-RAG* is OASIS's own query-time retrieval here: the stream is cut into event nodes, each summarised and embedded, the question is matched against those summaries, and the top-2 nodes contribute the keyframes the system stores. *Visual-RAG* follows SimpleStream's adaptation of retrieval-augmented generation (Lewis et al., 2020) to frames: every second of the prefix is one chunk embedded with CLIP (Radford et al., 2021), the question in the same space, and the 5 nearest chunks join the 4 recent frames. *Log-decay* keeps the whole prefix but spends 60% of its frames in the last 30 s, 30% up to 5 min, and the rest earlier. Both retrievers pass over the prefix at 1 fps to build what they search, and Text-RAG summarises it too, so neither is cheaper than the uniform pass it replaces. App. §B.2 matches their frame budgets. Two observations follow: **(a)** Used alone, a retriever trades one end for the other and loses more than it gains: against the short window they gain 15.37 and 8.15 on L2 but lose 13.83 and 15.10 on L1, so neither ends up ahead. **(b)** With the planner the trade disappears: all four score 59.94 on L1, because those questions never reach it. What survives is the history side, where they still differ by 2.22 on L2 and 2.92 on L4, and there Uniform is first or tied, reaching 52.94 overall against 51.76 for the best of the rest. It also needs nothing built in advance, so it is the one we deploy.

Table 5: **Candidates for the widest window.** Subscripts are median evidence distances on OVO-S-Bench; deltas are computed before rounding.

	$L1_{2.0\,s}$	$L3_{2.0\,s}$	$L2_{36.8\,s}$	$L4_{278.7\,s}$	Overall
short window	59.94	52.69	39.63	38.69	47.74
the window alone					
Text-RAG	46.10 (−13.83)	49.10 (−3.58)	55.00 (+15.37)	39.05 (+0.36)	47.31 (−0.43)
Visual-RAG	44.83 (−15.10)	50.54 (−2.15)	47.78 (+8.15)	39.78 (+1.09)	45.73 (−2.01)
Log-decay	**46.42** (−13.51)	52.33 (−0.36)	55.19 (+15.56)	40.51 (+1.82)	48.61 (+0.87)
Uniform	46.10 (−13.83)	**54.48** (+1.79)	**55.74** (+16.11)	**43.07** (+4.38)	**49.85** (+2.11)
with the planner and the probe					
Text-RAG	59.94	51.97	**54.26**	40.88	51.76
Visual-RAG	59.94	51.97	52.04	40.15	51.02
Log-decay	59.94	52.33	53.89	40.88	51.76
Uniform	59.94	**54.48**	**54.26**	**43.07**	**52.94**

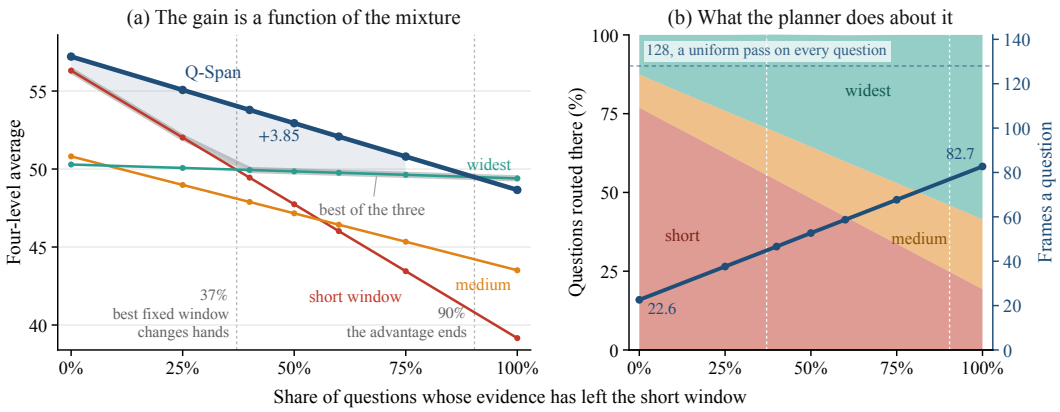

Figure 3: **The gain is a function of the mixture.** OVO-S-Bench at 8B, per-question outcomes re-weighted so a share w of questions has evidence outside the short window; the benchmark is $w = 0.5$; accuracy is linear in w by construction. **(a)** The three fixed windows, their envelope and Q-SPAN. **(b)** Where the deployed policy sends them and what it spends; it is never given w.

6 WHEN THIS DOES NOT HELP

The gain depends on how the questions divide among the three regimes, not on whether they disagree. They disagree everywhere: inside StreamingBench's unannotated real-time split, counting needs the widest window by 13.8 points at 8B and spatial understanding the shortest by 12.2 (App. §B.5). What differs is the balance. OVO-S-Bench's four levels are close to even, and routing beats the best single window by 3.09 points. Elsewhere one of the three holds most of the questions, so a window tuned to it is near the ceiling and the same system lands within half a point either way: $+0.05$ and -0.51 on OVO-Bench at 8B and 27B, $+0.16$ at both on StreamingBench's real-time subset (Table 13). That length has to be measured and is not the same twice, so against a window fixed in advance these runs stay ahead. Figure 3 measures that dependence, re-weighting outcomes already collected so that a share w of questions has evidence outside the short window. Against the best window a practitioner could tune at each mixture, the advantage is largest where that window changes hands, at $w = 37\%$, and closes at 90%. The planner is never told w, yet the share it sends to the widest window rises from 13% to 59% and the frames it spends from 22.6 to 82.7, under the 128 a uniform pass would spend on every question. Which case a deployment is in can be told before committing to either: the share the planner sends to the widest window tracks the gain across the nine cells, and needs no labels and no frames (App. §B.11). Four further limits follow from the mechanism, not the benchmark. The planner reads one question and no history, so a follow-up whose referent was established in an earlier turn carries less than it appears to; nothing here evaluates a dialogue. The widest window is a fixed budget rather than a rate, so a prefix of hours would thin it to a frame a minute and the widths would have to be reset. Choosing a span presupposes a cue to where the evidence lies, which the wording does not always carry, which is where the gain on OVO-S-Bench stops (Sec. 5). And a span is contiguous: where the evidence is scattered, no single choice covers it, the case a retriever should serve and does not yet.

7 CONCLUSION

The accuracy of a streaming video assistant is decided by which slice of the past reaches the model, and that slice cannot be fixed: across three benchmarks the best look-back window varies by at least $64\times$, and between subsets of one benchmark as much as between benchmarks. We choose it per question instead, asking each source for what only it knows: which of three windows a question calls for is settled by its wording, so one text call chooses it before any frame is decoded, while whether that window reaches the evidence is settled by the video and asked of frames already decoded. Both calls are that same backbone under a different prompt, so the choice adds two labels and no weights, and costs a fraction of what getting it wrong does. Two results seem worth carrying forward. Asking the answering model to reason first cost accuracy on perception and episodic memory, and the more evidence the prompt carried the more it cost, so deliberation belongs in separate calls. And an aggregate sweep can be nearly flat while per-question choice is worth three points on the same benchmark: a policy that looks inert in dataset averages may still be the axis that matters, until nearly every question reaches back and one wide window is right again.

REPRODUCIBILITY STATEMENT

The method is a pair of prompts over frozen backbones, so what has to be shared to reproduce it is small and all of it is here. Both prompts are given in full, and the appendix states the frame counts, sampling rates, pixel budgets, decoding settings and frame presentation behind the runs we report, together with which of these are held fixed across benchmarks and backbones. Each benchmark is scored by its own official scorer and task grouping; our code composes those per-question outcomes into the policies we compare. The backbones are public checkpoints, cited where they are introduced. Where a number depends on a choice a reader might make differently, we measure the alternatives beside it rather than leave it implicit.

USE OF LARGE LANGUAGE MODELS

An LLM-based assistant was used to aid and polish the writing of this paper: drafting and revising prose, and editing it for length and readability. The authors reviewed the resulting text and are responsible for every claim, number and artifact in this submission.

REFERENCES

Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. StreamReady: Learning what to answer and when in long streaming videos. *arXiv preprint arXiv:2603.08620*, 2026.

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*, 2025a.

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

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 *CVPR*, 2025.

Joya Chen et al. VideoLLM-online: Online video large language model for streaming video. In *CVPR*, 2024a.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. StreamingTOM: Streaming token compression for efficient video understanding. In *CVPR*, 2026.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024b.

Gemini Team, Petko Georgiev, Ving Ian Lei, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Google DeepMind. Gemini 3.1 Pro. `https://blog.google/products/gemini/gemini-3-pro/`, February 2026.

Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, and Trishul Chilimbi. M-LLM based video frame selection for efficient video understanding. In *CVPR*, 2025.

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 *CVPR*, 2025.

Aaron Hurst, Adam Lerer, Adam P. Goucher, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot-V: Memory-constrained KV cache compression for streaming video understanding. In *NeurIPS*, 2025.

Jie Lei, Tamara L. Berg, and Mohit Bansal. QVHighlights: Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.

Bo Li, Yuanhan Zhang, Dong Guo, Rui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *TMLR*, 2025a.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How far is your video-LLMs from real-world online video understanding? In *CVPR*, 2025b.

Yifei Li, Pengyiang Liu, Yuhang Zang, Zhongyue Shi, Qi Fu, Hongye Hao, and Jiwen Lu. OVO-S-Bench: A hierarchical benchmark for streaming spatial intelligence in multimodal LLMs. In *EMNLP*, 2026.

Zhijia Liang, Jiaming Li, Weikai Chen, Yanhao Zhang, Haonan Lu, and Guanbin Li. OASIS: On-demand hierarchical event memory for streaming video reasoning. In *CVPR*, 2026.

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. *arXiv preprint arXiv:2411.03628*, 2024.

Zikang Liu, Longteng Guo, Handong Li, Ru Zhen, Xingjian He, Ruyi Ji, Xiaoming Ren, Yanhao Zhang, Haonan Lu, and Jing Liu. Thinking in streaming video. *arXiv preprint arXiv:2603.12938*, 2026.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. Faster cascades via speculative decoding. *arXiv preprint arXiv:2405.19261*, 2025.

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

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *NeurIPS*, 2024.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video LLMs with active real-time interaction via disentangled perception, decision and reaction. In *CVPR*, 2025.

Qwen Team. Qwen3.8-27B. `https://www.qwencloud.com/models/qwen3.8-27b`, 2026. Weights at `https://huggingface.co/Qwen/Qwen3.8-27B`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

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

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

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *ICCV*, 2025a.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. VideoLLM knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. In *Findings of EMNLP*, 2025b.

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 *CVPR*, 2025c.

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

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

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-time understanding for infinite video streams. In *ICLR*, 2026.

Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. StreamForest: Efficient online video understanding with persistent event memory. In *NeurIPS*, 2025.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. StreamMem: Query-agnostic KV cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. TimeChat-Online: 80% visual tokens are naturally redundant in streaming videos. In *ACM MM*, 2025.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. Frame-Voyager: Learning to query frames for video large language models. In *ICLR*, 2025.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-VStream: Efficient real-time understanding for long video streams. In *ICCV*, 2025a.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. HERMES: KV cache as hierarchical memory for efficient streaming video understanding. In *ACL*, 2026.

Yulin Zhang, Cheng Shi, Yang Wang, and Sibei Yang. Eyes wide open: Ego proactive video-LLM for streaming video. In *NeurIPS*, 2025b.

Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. *arXiv preprint arXiv:2510.27280*, 2025.

Appendix

A MEASUREMENT PROTOCOL

The three windows. A window is N frames at K frames per second ending at the query timestamp, so it covers N/K seconds. Short is $(4, 1)$ and covers 4 s; medium is $(32, 1)$ and covers 32 s. Full drops the rate and spreads 128 frames uniformly from the start of the stream to the query, so it is a frame cap rather than a rate; the cap binds on most questions, since median query times are 134–264 s, which is why we describe that window by its count rather than as "1 fps". The curves in Figure 1 are four windows of the first kind, 4, 32, 128 and 256 frames at one frame a second, so the axis there is a length throughout and the widest read this method deploys, being a budget, is not on it; each panel varies nothing else.

Whether the penalty is the frame order. The curves in Figure 1 are read as a claim about reach, so it is fair to ask whether the wide window's penalty is instead the backbone not knowing which frame is the present: the frames arrive as independent image blocks and nothing in the prompt says the last one is now. The four widths were rerun with one sentence added to the answering prompt, that the frames are in chronological order and the last is the current moment, and nothing else changed. The penalty does not shrink. On OVO-S-Bench's nearest level the fall from 4 frames to 256 is 13.0 points as reported and 15.1 with the sentence added ($59.9 \rightarrow 46.9$ against $59.3 \rightarrow 44.2$), and the level whose evidence sits half a minute back gains 14.8 and 15.2. Four-level averages move by at most 0.9 at any width. Telling the model where the present is does not make a wide window behave like a narrow one, so the curves are not an artefact of how the frames are presented.

Presentation. Frames are presented as independent image blocks rather than as a single video block.

Decoding. Decoding is greedy throughout, so a run is reproducible question by question.

What the planner and the probe cost in time. Sec. 4.3 argues against reasoning inside the answering prompt partly on latency, so the same measurement is owed for the two calls this method adds, and Table 6 reports it. All three run (the planner, the probe and the answering model) on one vLLM server at tensor parallel four, since that is how a deployment would host them and the only difference between the calls is whether images are attached; frames are already in hand in a streaming setting, so nothing here charges video decoding. What is timed is the wait before the answer's first token: the planner and the probe have to finish before frames can be chosen, so each contributes its whole generation, while the answering call contributes its prefill and first token. Every question is given distinct frames, because repeating one frame lets the prefix cache serve one question's pixels from another's and understates the prefill by several times. Each question is timed on its own, with no cache carried over from the previous one. A deployment that kept a growing prefix in cache would pay less than this for every policy in Table 6, and most of all for the widest window, since it is the one whose prefill is largest, so the saving it reports is an upper bound on what choosing the window buys in time. On OVO-S-Bench the planner costs 32 ms on every question and the probe 23 ms on the ones it sends to the short window, so at most 55 ms together, and neither grows with the stream. Against them, the window they choose is worth 523 ms between the short and the medium one and 2604 ms between the short and the widest, so the choice costs a fraction of what getting it wrong does. It also keeps the system quick: most questions take the short window, which answers in 76 ms including both auxiliary calls, where a system that read the whole prefix every time would pay 2625 ms on every question.

What a memory costs before the question. The calls this method adds are paid at the question and nowhere else. A memory built over the prefix is the opposite: its cost is paid while the stream runs, whether or not a question ever arrives, and it grows with the stream. The forests our OASIS (Liang et al., 2026) reproduction wrote hold one event node per 16.6 seconds of stream, averaged over 337 of them and 31708 nodes, and summarising one node, 16 frames at 256 tokens each, takes a median of 1187 ms on the server Table 6 is timed on. That is 4.3 seconds of this server for every minute of stream: 4.3 minutes for an hour, and 19 minutes for the four-and-a-half-hour prefix this benchmark reaches (App. §B.12). It is a throughput cost rather than a wait, since the summarising keeps up with the stream on this hardware, but it occupies all four of its GPUs continuously and it is the reason a memory has to be built in advance to be there when asked. Choosing a window costs nothing until a question exists. Timed at the question, that system is faster: embedding the question takes 21.6 ms,

Table 6: **What choosing the window costs and what it buys.** Time to the answer's first token (TTFT) in milliseconds, medians on one Qwen3-VL-8B server at tensor parallel four, every frame given at most 256 visual tokens, beside the accuracy each policy reaches. Each Q-SPAN row's *answering* column averages the three windows by how often they are picked, and its TTFT is an upper bound, since the probe runs only where the planner names the short window. OASIS has no probe, and its *planner* column is embedding the question and searching the forest. ‡It also summarises the stream as it arrives, whether or not a question follows: one question about an hour of stream costs it 4 minutes on this server before the question exists, against our 1 second.

		Latency (ms)				
Policy	Frames	Planner	Probe	Answering	TTFT	Accuracy (%)
OVO-S-Bench, where most questions reach back						
Always the short window	4	0	0	21	21	47.74
Always the medium window	32	0	0	544	544	47.16
Always the widest window	128	0	0	2625	2625	49.85
OASIS [CVPR'26], a memory	16	22	—	279	301‡	46.8
Q-SPAN, the window chosen per question	44.6	32	23	863	**918**	**52.94**
StreamingBench real-time set, where most ask about the present						
Always the short window	4	0	0	24	24	80.96
Always the medium window	28.6	0	0	625	625	<u>83.80</u>
Always the widest window	125.9	0	0	3765	3765	80.28
OASIS [CVPR'26], a memory	16	22	—	305	327‡	78.2
Q-SPAN, the window chosen per question	20.1	52	26	463	**540**	**83.96**

searching a forest of 76 nodes is a dot product and takes 0.01, and answering over the 16 frames of its now-window together with two retrieved summaries reaches its first token in 279 ms, so it waits 301 ms against our 918. The reason is the same as the reason it scores lower: it answers from 16 frames and some text, so there is little to prefill. Against the short window the trade is starker still, since 4 frames at 21 ms score 47.74 against its 46.8. What a memory buys is not speed at the question and, on this benchmark, not accuracy either; what it costs is that standing charge.

The frozen-backbone rows. The row marked [Base] in each block is the answering model with no policy of any kind: frames at 1 fps ending at the query instant, up to 256 of them, so a literal 1 fps reaching back 256 s. On OVO-S-Bench that row reproduces the benchmark's own protocol, 128 frames sampled uniformly from the prefix. Sampling the whole prefix at a literal 1 fps instead, which on OVO-Bench needs up to 1728 frames and drops the per-frame budget to 134 tokens, costs 13 points of OCR there on Qwen2.5-VL-7B and is worse on every column, so the frame cap is doing real work rather than saving time.

The Qwen3.8-27B runs. One detail is specific to the larger backbone. It reasons by default: its chat template prefills an opening think tag, so under the letter-only decode budget it is still deliberating when generation stops, never emits the closing tag, and scores near zero on every task; an artefact that is indistinguishable from a model with no video understanding, and one worth naming because it silently voids an entire run. We therefore disable reasoning at the template level for these runs, which prefills an *empty* think block and puts the 27B on the same letter-only footing as the 8B runs; responses are then single option letters, as they are for the Qwen3-VL-8B.

B ADDITIONAL ABLATIONS

B.1 WHAT EACH PLANNER CHOOSES

Two planners can reach the same score by different routes, so the planners of Table 3 are read alongside what each of them actually does. Table 7 gives the share of questions each sends to each window. It is what identifies the failure at the smallest size as a collapse rather than as noise: the 0.6B planner names one window for most questions on both benchmarks, so it scores near not routing at all because it very nearly does not route.

Table 7: **What each planner chooses.** Where each text-only planner of Table 3 sends its questions, and where the answering backbone sends them when it plans for itself. The smallest planner does not route: it sends most questions to the short window on both benchmarks, which is why its score sits next to that of not routing at all.

	OVO-S-Bench			StreamingBench RT		
Planner	Short	Medium	Widest	Short	Medium	Widest
Qwen3-0.6B	78%	14%	8%	85%	14%	1%
Qwen3-1.7B	43%	22%	35%	60%	24%	16%
Qwen3-4B	53%	16%	31%	69%	19%	12%
Qwen3-8B	49%	19%	32%	68%	21%	11%
Qwen3-32B	51%	15%	34%	70%	18%	12%
Qwen3-VL-8B, the answering backbone	61%	10%	29%	76%	15%	9%

Table 8: **The widest window on both benchmarks, at native and matched budgets.** Each policy at the budget its own method specifies (Table 5), and each retriever again at the uniform pass's 128 frames; Visual-RAG's two matched rows differ in the window it spreads those frames over. Every run on OVO-Bench is decoded through one frame-selection path, baselines included.

		OVO-S real-time		OVO-S history		OVO-B real-time	OVO-B history	
	Frames	L1	L3	L2	L4	RT	EPM	ASI
The short window	4	59.94	52.69	39.63	38.69	80.87	52.19	62.16
Text-RAG, native	32	46.10	49.10	55.00	39.05	57.88	49.83	59.46
Text-RAG, matched	128	44.83	50.54	55.93	40.88	57.89	50.51	60.14
Visual-RAG, native	9	44.83	50.54	47.78	39.78	79.79	61.28	65.54
Visual-RAG, matched (64 s)	128	39.27	48.39	52.04	39.78	62.12	58.25	62.16
Visual-RAG, matched (32 s)	128	38.31	48.39	48.89	34.31	59.80	53.87	62.16
Log-decay	128	46.42	52.33	55.19	40.51	76.93	64.98	67.57
Uniform	128	46.10	54.48	55.74	43.07	73.65	63.64	68.92

B.2 The widest window at matched budgets

Table 5 gives each candidate for the widest window the frame budget its own method specifies, which is how those methods are published but leaves the comparison confounded with how many frames each one spends. Table 8 repeats it with the retrievers given the uniform pass's 128 frames, and adds OVO-Bench, where the retrievers can be run end to end; App. §B.3 adds the third benchmark.

Matching the budget does not rescue retrieval. Text-RAG moves by hundredths, since the two nodes it returns already span the same stretch either way, and Visual-RAG gets worse the more it retrieves, by 17.7 points on OVO-Bench's real-time tasks, because the frames it adds are scattered and past a handful they crowd out the recent ones. The table also shows the disagreement this paper is about: the short window is best on two of the three real-time columns and the uniform pass on three of the four whose evidence lies further back. Neither ordering holds on the other kind of question, which is why the widest window is a uniform pass and why something must choose between them.

B.3 A retriever on the third benchmark

Table 8 compares retrieval against a chosen window on the two benchmarks that separate questions about the present from questions that reach back, by level on one and by task on the other. Table 9 adds StreamingBench's real-time split. Visual-RAG is run there as its authors specify, a CLIP index over one-second chunks of the observed prefix with the top five chunks appended to the 4 recent frames, and then again with the retrieved count raised to sixteen so that it spends what the router spends. Text-RAG is not run: its index is a written summary per event, so building one for this split means captioning every video in it, the retriever's offline cost rather than a number we can borrow.

The conclusion is the same as on the other two benchmarks. Retrieval is worth something over the recent window alone and less than choosing how far back to look: at its own budget it gains 0.64 points over the 4 recent frames and sits 2.36 below the router, and given the router's budget it loses

Table 9: **Visual-RAG on StreamingBench's real-time split.** Accuracy (%) with every policy scored on the same questions. Frames are per question, and the retriever's count falls below its nominal budget where the prefix is shorter than the window it would fill.

Policy	Accuracy	Frames
The short window	80.96	4.0
The medium window	83.80	32.0
The widest window (Uniform)	80.28	128.0
Visual-RAG (5 retrieved, 4 recent)	81.60	8.8
Visual-RAG (16 retrieved, 4 recent)	81.08	18.4
The same path given the 4 recent frames alone	80.96	4.0
Q-SPAN, the window chosen per question	**83.96**	20.1

Table 10: **Reasoning inside the answer prompt costs most where there is most to read.** Perception is the macro over the six OVO-Bench perception tasks; EPM, ASI and HLD are episodic memory, action-sequence identification and hallucination detection. Differences are computed before rounding.

	4 frames		32 frames		128 uniform	
	Letter	Step-by-step	Letter	Step-by-step	Letter	Step-by-step
Perception	**82.34**	80.29	**78.83**	76.24	**72.90**	64.27
EPM	**52.86**	51.18	**64.65**	55.56	**64.98**	54.55
ASI	**58.11**	57.43	**75.68**	67.57	**69.59**	66.22
HLD	43.01	**62.37**	35.48	**54.84**	27.96	**38.71**
Δ Perception	−2.04		−2.59		−8.63	
Δ EPM	−1.68		−9.09		−10.44	
Δ HLD	+19.35		+19.35		+10.75	

0.52 of that gain rather than closing the gap, ending 2.88 behind at the same frame count. Spending more on retrieval makes it worse here, which is what Table 8 finds on the other two benchmarks as well: the frames a retriever adds are scattered, and past a handful they crowd out the recent ones that most of this split turns on. The last retrieval row is the check that makes the rest of the table readable, since the same code path given only the 4 recent frames reproduces the short window measured through the ordinary contiguous path to the digit; the difference between these rows is therefore where the frames are taken from and nothing else.

B.4 REASONING INSIDE THE ANSWER PROMPT

Table 10 is the measurement behind Sec. 4.3: two prompts differing in nothing but the instruction, one asking for a letter, the other asking the model to think step by step and put its answer in a tag, at the same window, resolution, presentation, decoding and seed, at each of the three windows the method deploys. Reasoning costs perception at every width, and it costs more the more there is to read: 2.04 points on 4 frames, 2.59 on 32 and 8.63 on the uniform pass. Episodic memory behaves the same way, 1.68, 9.09 and 10.44. That is the opposite of a method reasoning *about* the evidence, which would pay least where the evidence is richest.

What it buys is one ability, declining to answer: hallucination detection gains 19.4 points at both narrower windows and 10.8 at the widest, because a model that reasons first is less willing to affirm something it did not see. Hallucination detection measures robustness to unsupported options rather than recall of what was seen (Shen et al., 2026), and an assistant answering in free form has no option list to detect, so a rule keyed on one would not transfer out of multiple-choice evaluation. The gain is confined to the 186 questions that offer an abstention choice, so we note the measurement and leave it out of the method.

Table 11: **StreamingBench real-time, by its own task labels.** Accuracy (%) at each of the three windows, then the planner alone and the deployed planner-and-probe, on the same questions. Tasks are ordered by size and **bold** marks the best of the three windows for each backbone. The last row pools all 2500 questions, which is the metric the benchmark defines and the tables report.

	Qwen2.5-VL-7B					Qwen3-VL-8B					Qwen3.8-27B				
Task	Short	Medium	Widest	Planner	+Probe	Short	Medium	Widest	Planner	+Probe	Short	Medium	Widest	Planner	+Probe
Object perception	85.4	**90.5**	89.4	87.0	89.7	86.2	**90.0**	86.2	87.3	89.4	87.8	**92.1**	91.1	88.9	91.3
Action perception	76.7	**82.4**	77.3	77.3	82.4	77.6	**78.7**	72.2	79.0	79.3	85.5	**87.2**	84.9	84.9	85.8
Text-rich understanding	92.2	**95.6**	92.5	93.8	95.6	90.7	**93.2**	91.0	91.6	94.1	91.9	**97.5**	94.7	93.1	94.1
Clips summarisation	91.8	**94.0**	92.4	92.4	94.0	91.5	**95.6**	89.9	91.5	91.5	93.1	**95.3**	91.8	92.1	92.4
Attribute perception	88.1	90.1	**91.0**	88.5	90.1	90.1	**91.4**	86.5	90.4	92.0	**93.3**	93.0	89.1	93.3	93.6
Spatial understanding	**80.5**	80.1	74.8	79.3	77.6	**79.3**	75.2	67.1	78.5	78.5	**81.7**	78.5	74.4	82.5	82.1
Counting	41.5	44.7	**50.5**	50.0	51.1	41.0	43.6	**54.8**	54.8	54.8	46.8	52.7	**66.0**	67.6	67.6
Event understanding	75.5	**83.0**	81.8	80.5	82.4	73.6	**83.0**	81.8	78.0	79.2	79.3	**89.3**	83.0	83.0	84.9
Causal reasoning	71.1	81.2	**84.4**	78.1	80.5	69.5	**82.8**	79.7	71.1	74.2	83.6	87.5	**90.6**	85.9	87.5
Prospective reasoning	75.0	**86.1**	84.3	86.1	86.1	86.1	**87.0**	81.5	87.0	87.0	80.6	**84.3**	84.3	84.3	84.3
All 2500 questions	80.6	84.8	83.4	82.9	84.9	81.0	83.8	80.3	82.8	84.0	84.6	87.6	86.2	86.9	87.7

Table 12: **An *oracle* router given the evidence distance, against the deployed one.** Accuracy (%) on the 1669 OVO-S-Bench questions that carry an evidence interval and were measured at all three windows. The last row names the narrowest window whose reach covers where the evidence begins; it sees the annotation but never an answer, so unlike a maximum over the three windows it cannot be inflated by guessing. *Frames* is the mean per question.

Policy	L1	L2	L3	L4	Average	Frames
Always the short window	**59.94**	39.63	52.69	40.27	48.13	4.0
Always the medium window	49.28	48.70	52.33	39.82	47.53	32.0
Always the widest window	46.10	55.74	54.48	**47.96**	51.07	128.0
Q-SPAN, the planner alone	**59.94**	52.04	54.84	46.15	53.24	43.8
Q-SPAN, planner and probe	**59.94**	54.26	54.48	46.61	53.82	44.4
Oracle, given the evidence distance	59.78	**59.26**	**55.91**	**47.96**	**55.73**	56.7

B.5 STREAMINGBENCH REAL-TIME, BY TASK

Sec. 2 shows populations disagreeing about the window, and OVO-S-Bench's levels are the cleanest case because the benchmark annotates evidence distance. Table 11 shows the same disagreement inside a single subset that carries no such annotation: StreamingBench's real-time split, broken out by its own ten task labels. Counting needs the widest window on all three backbones, by 9 to 19 points over the short one, and it is the only task that does; spatial understanding needs the short window and loses 6 to 12 points at the widest; six of the remaining eight prefer the middle on all three backbones, while attribute perception and causal reasoning peak at the widest on the 7B. A window chosen for the split as a whole therefore gives something up whichever it is, and the split is scored as one number in every published table including ours.

B.6 AN *oracle* ROUTER GIVEN THE EVIDENCE DISTANCE

OVO-S-Bench records the interval each question's evidence lies in, so a router can be given what the wording cannot supply and asked to name the narrowest of the three windows that reaches back to it. Table 12 puts that router beside the deployed one. It is an *oracle* in the sense that it reads an annotation no deployed system has.

The deployed planner names the same window as the annotation on 74% of questions, and the gap between them is 1.91 points. Two things follow. Reading the question recovers most of what knowing the distance would give, which is the claim Sec. 4.1 rests on. And the remaining gap is not the probe's to close: the annotation sends 40% of questions to the widest window, where a verdict on 4 frames has nothing to say.

Is the label a distance in disguise? The planner's choices track the annotation closely enough to raise the question, since 80% of L4 goes to the widest window. Two measurements bracket the answer. Reading the label off the annotated distance, under the best pair of thresholds fitted to the planner's own output, reproduces it on 74% of questions against 61% for always naming its most common label, and inside a narrow band of distances the planner still splits: where the evidence

Table 13: **A paired bootstrap of the gap over the strongest fixed window.** 4000 resamples of the question set per setting, within scoring group. *Fixed* names the single window the routed system is measured against, which is the strongest of the three in that setting.

Benchmark & backbone	Q-SPAN	Fixed	Gap	95% interval
OVO-S-Bench 7B	46.81	widest	+1.19	[−0.26, +2.64]
OVO-S-Bench 8B	52.94	widest	**+3.09**	[+1.43, +4.75]
OVO-S-Bench 27B	60.05	widest	**+4.92**	[+3.31, +6.54]
OVO-Bench 7B	77.64	medium	−0.39	[−1.07, +0.24]
OVO-Bench 8B	82.38	short	+0.05	[−1.26, +1.39]
OVO-Bench 27B	84.52	short	−0.51	[−1.64, +0.55]
StreamingBench RT 7B	84.88	medium	+0.08	[−0.44, +0.60]
StreamingBench RT 8B	83.96	medium	+0.16	[−1.08, +1.36]
StreamingBench RT 27B	87.72	medium	+0.16	[−0.88, +1.20]

begins within 2 seconds it sends 7% of questions to the widest window, and where it begins beyond 2 minutes it keeps 22% on the short one. A duration bucketiser cannot do either. What the label is not is better than the distance: given the distance the router in this table gains 1.91 points over ours, while a planner asked for the distance itself loses 2.97 (Table 22). That is the claim of Sec. 4.1 in two numbers, and App. §B.20 asks the frames the same question instead. The quantity we read off the wording is worth less than knowing how far back the evidence lies and more than guessing it, because it is the one of the two the wording contains.

The router has to be defined over the annotation rather than the answers. Keeping whichever window answered each question is the obvious construction and measures something else: no policy can know which window was right, a quarter of these questions offer only two options so a wrong window is right half the time by chance, and the three windows are far from independent, reaching 67.5 between them against the 87.2 that independent errors at the same accuracies would give.

B.7 How much of the gap is noise

Every number in this paper is greedy, so a seed changes nothing and reporting a spread over seeds would be reporting zero. That leaves two sources of variance that are real, and they are separate questions. The pipeline makes two decisions by generation, and they could have come out otherwise; and the question set is a sample, so the gap between two policies has a sampling error of its own. This section measures both on the deployed system, without refitting anything.

Resampling the two decisions. The planner and the probe were rerun at temperature 0.7 under five seeds, paired seed by seed, with the answering runs left as they are: a seed moves a question between windows, so the routed score is a lookup and five draws cost two generations each. On OVO-S-Bench at 8B the five draws give 53.01 ± 0.21 against the greedy 52.94, a gain over the widest window of 3.16 ± 0.21 points that is positive in five of five draws. The frame count is 44.6 ± 0.2 and the probe escalates $7.2 \pm 0.7\%$ of questions. Sampling reaches the same window as the greedy run on 98% of questions, which is the reason the spread is this small: the planner's choice is not a close call on most questions.

Resampling the questions. Table 13 gives a paired bootstrap of routed minus the strongest fixed window, 4000 resamples per setting, a setting being one benchmark answered by one backbone, drawn within each scoring group so that the statistic resampled is the benchmark's own estimator and the two policies are compared on the same questions. The conclusion is the one Sec. 6 states, now with its uncertainty attached. Where the levels differ in how far back they reach, the gap is positive on all three backbones and clears zero in every resample on the two larger ones. Where one regime holds most of the questions, the interval straddles zero: those settings are ties, not small wins or small losses, and neither the +0.16 nor the −0.51 should be read as a direction.

B.8 Which fixed window, and who gets to choose it

Table 13 compares routing against the strongest of the three windows in each setting. A fourth candidate would not raise that maximum: the 256-frame window at 1 fps, the widest point of Figure 1,

Table 14: **Routing against each window held fixed in advance.** Gap in points with a 95% paired bootstrap interval, 4000 resamples, same estimator per benchmark as elsewhere. † marks the window that is strongest in that setting, which is the baseline Table 13 uses and the one only hindsight supplies. *Frames* is routing's mean per question.

Benchmark & backbone	Q-SPAN	Frames	against the short	against the medium	against the widest
OVO-S-Bench 7B	46.81	51.5	$+4.40$ $[+2.18, +6.54]$	$+2.67$ $[+1.07, +4.23]$	$+1.19$ $[-0.26, +2.64]^{\dagger}$
OVO-S-Bench 8B	52.94	44.6	$+5.20$ $[+3.24, +7.26]$	$+5.78$ $[+3.59, +8.03]$	$+3.09$ $[+1.43, +4.75]^{\dagger}$
OVO-S-Bench 27B	60.05	59.4	$+10.39$ $[+8.21, +12.66]$	$+8.76$ $[+6.36, +11.17]$	$+4.92$ $[+3.31, +6.54]^{\dagger}$
OVO-Bench 7B	77.64	30.4	-0.29 $[-2.05, +1.52]$	-0.39 $[-1.07, +0.24]^{\dagger}$	$+4.64$ $[+2.67, +6.71]$
OVO-Bench 8B	82.38	19.7	$+0.05$ $[-1.26, +1.39]^{\dagger}$	$+3.55$ $[+1.72, +5.44]$	$+9.49$ $[+6.88, +12.03]$
OVO-Bench 27B	84.52	19.5	-0.51 $[-1.64, +0.55]^{\dagger}$	$+4.44$ $[+2.64, +6.37]$	$+8.15$ $[+5.76, +10.62]$
StreamingBench RT 7B	84.88	34.6	$+4.28$ $[+3.08, +5.40]$	$+0.08$ $[-0.44, +0.60]^{\dagger}$	$+1.52$ $[+0.48, +2.60]$
StreamingBench RT 8B	83.96	20.1	$+3.00$ $[+2.12, +3.92]$	$+0.16$ $[-1.08, +1.36]^{\dagger}$	$+3.68$ $[+2.24, +5.08]$
StreamingBench RT 27B	87.72	22.4	$+3.12$ $[+2.16, +4.12]$	$+0.16$ $[-0.88, +1.20]^{\dagger}$	$+1.56$ $[+0.48, +2.64]$
mean over the nine settings			$+3.29$	$+2.80$	$+4.25$
worst of the nine			-0.51	-0.39	$+1.19$

wins L4 outright at 47.08 and loses the four-level average at 49.54 against the widest window's 49.85. That is the hardest comparison available and it is also one no deployment could run, because the label is read off the results afterwards. It is worth being explicit about how little that label transfers. The strongest window is the widest in all three OVO-S-Bench settings, the medium one in all three StreamingBench settings and in OVO-Bench at 7B, and the short one in OVO-Bench at 8B and 27B: it moves with the benchmark, and on OVO-Bench it moves between backbones answering the same questions. Committing to a window in advance means committing to being wrong about it somewhere.

Table 14 therefore scores the policy a deployment can have. Held fixed across all nine settings, the short window loses 3.29 points to routing on average and the medium one 2.80; the widest is the only choice routing beats in every one of the nine settings, and it pays for that in frames, spending 128 a question against routing's 20 to 60. The ties in Table 13 survive here and sit where the benchmarks put them: where the questions concern the present moment, the window that serves the present is level with routing, and on OVO-Bench at 27B it is half a point ahead. What does not survive is that a deployment could have held that window without measuring which one it was. Both comparisons are worth stating, since one asks whether the method beats a policy that could have been deployed and the other whether it beats hindsight, and the answers differ. The mean over nine settings is itself an aggregate of the kind Sec. 2 warns about: it is carried by OVO-S-Bench, whose levels are close to even, and shrinks toward zero on the benchmarks whose questions concentrate on the present, exactly as the mixture sweep of App. §B.13 predicts.

B.9 A FIXED WINDOW TUNED ON HELD-OUT QUESTIONS

Two comparisons appear above and neither is the protocol a practitioner would follow. Table 13 gives the baseline the strongest window per setting, which is read off the results; Table 14 makes it commit to one window everywhere, which is unlucky rather than unfair. The protocol in between is to tune: split the questions in half, choose the single best window on the first half, and report both policies on the second. Table 15 does that over 200 random splits per setting.

Tuning does not rescue the fixed window. Averaged over the nine settings the routed system is 1.10 points ahead on held-out questions against 1.00 under the hindsight comparison, and it is ahead in eight of the nine settings. The reason is visible in the last column: in most settings the validation half picks the same window hindsight would, so the two comparisons coincide. Where it does not, on OVO-Bench at 7B, the tuning splits evenly between two windows and lands on the worse one half the time, which is what turns a hindsight deficit of 0.39 into a held-out advantage of 0.02. Whatever the routed system's advantage is, it is not that its baseline was denied a validation set.

B.10 THE SAME QUESTIONS, PARTITIONED BY THEIR OWN EVIDENCE DISTANCE

A router handed those distances is measured in App. §B.6. OVO-S-Bench's four levels have median evidence distances of 2, 37, 2 and 279 seconds, close enough to the widths this method deploys

Table 15: **Routing against a fixed window tuned on half the questions.** Mean and standard deviation over 200 random halves per setting, scored on the half not tuned on, with the same estimator per benchmark as elsewhere. *Hindsight* repeats the comparison of Table 13 for reference. *Worst split* is the worst of the halves and *Tuning picks* how often each window won the validation half.

Benchmark & backbone	Held out	Worst split	Hindsight	Tuning picks
OVO-S-Bench 7B	$+1.42 \pm 0.82$	-0.06	$+1.19$	widest 96%, medium 4%
OVO-S-Bench 8B	$+3.29 \pm 1.00$	$+0.67$	$+3.09$	widest 98%, short 2%
OVO-S-Bench 27B	$+5.02 \pm 0.96$	$+3.03$	$+4.92$	widest 100%
OVO-Bench 7B	$+0.02 \pm 0.58$	-1.06	-0.39	medium 50%, short 50%
OVO-Bench 8B	$+0.05 \pm 0.67$	-1.71	$+0.05$	short 100%
OVO-Bench 27B	-0.51 ± 0.62	-2.02	-0.51	short 100%
StreamingBench RT 7B	$+0.33 \pm 0.24$	-0.24	$+0.36$	medium 100%
StreamingBench RT 8B	$+0.13 \pm 0.59$	-1.52	$+0.16$	medium 100%
StreamingBench RT 27B	$+0.20 \pm 0.57$	-1.44	$+0.16$	medium 100%
mean over the nine settings	$+1.10$		$+1.00$	

Table 16: **Re-partitioned by annotated evidence distance.** OVO-S-Bench at 8B, the 1669 questions that carry an evidence interval, binned by how far back the interval begins. *Fixed* is the best of the three windows within the bin. The last row weights the five bins equally, which is the aggregation the four-level average performs on a different partition.

Evidence begins	n	Short	Medium	Widest	Q-SPAN	Fixed	Gain
within 1 s	20	55.00	50.00	45.00	55.00	short	$+0.00$
1 to 8 s	835	57.01	49.22	46.35	57.01	short	$+0.00$
8 to 64 s	234	41.03	60.68	55.13	58.97	medium	-1.71
64 to 512 s	434	40.09	40.09	54.84	50.23	widest	-4.61
beyond 512 s	146	47.95	47.95	58.90	56.16	widest	-2.74
the five bins weighted equally	1669	48.21	49.59	52.04	**55.48**	widest	$\mathbf{+3.43}$

to invite the reading that the widths were chosen for them. The predictions do not depend on that partition, but the reported gain does, since the four-level average weights it. Table 16 re-partitions the same 1669 questions that carry an evidence interval by the interval itself, on a grid of octaves that lines up with none of the three widths, and weights the bins equally. The gain survives: 3.43 points, against 2.75 on these questions under the benchmark's own levels, and 4.46 under a partition at the distance quartiles (1.0, 5.4 and 166 seconds). Whatever the widths were chosen for, the advantage is not an artefact of how this benchmark groups its questions.

The per-bin columns say where it comes from, and they are worth reading in the paper's own terms. Inside a bin routing is often behind the best window for that bin, by 4.61 points where the evidence lies between one and eight minutes back, because a bin of questions that all reach equally far back is exactly the homogeneous population a fixed window serves and a router can only spoil. The two nearest bins are zero rather than behind because the planner sends every one of those questions to the short window, which is the window those bins want, so the two policies are the same policy there. The gain appears when the bins are put back together, which is the claim of Sec. 2 restated on a partition the benchmark did not choose.

Scattered evidence, and what this benchmark can say about it. Sec. 6 names the case a contiguous window cannot serve: evidence in several places at once. OVO-S-Bench cannot measure it. Its annotation gives one interval per question, 1669 of them, with a median span of 2.0 seconds, so every annotated question has its evidence in a single short place and the scattered case is outside the benchmark rather than inside it. That is also why the comparison against retrievers (Table 5, and App. §B.2 at matched budgets) is the only evidence we have on the question: a retriever is the policy that can return several places, and on these benchmarks it does not beat returning one.

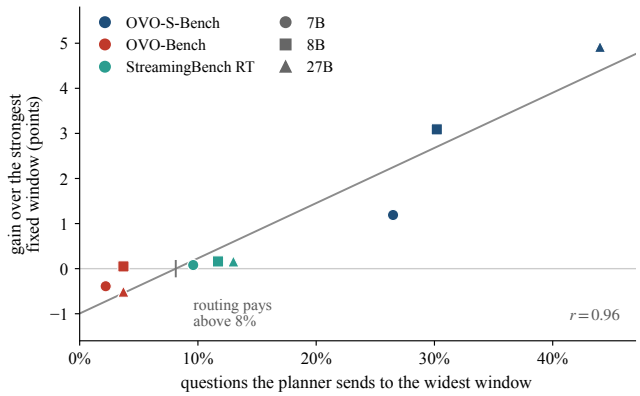

Figure 4: **What the planner's own traffic says about whether to route.** The share of questions the planner sends to the widest window, against what routing is worth in that setting. The share is the planner's output before the cap, so it is what a deployment reads off a sample of its questions with no video decoded and no answers needed; the gain is against the strongest fixed window, which only hindsight names (App. §B.8).

B.11 Deciding in advance whether to route at all

Sec. 6 says the gain depends on the mixture, which is only useful to a deployment that has already measured its mixture. There is a cheaper signal, and it needs no labels and no frames: run the planner over the questions and look at how much traffic it sends to the widest window. Figure 4 puts that share against the gain over the strongest fixed window on each of the nine settings of Table 14. The two track each other with a correlation of 0.96, and a line fitted to the nine points crosses zero at 8% and reaches a full point at 16. The share is a property of the benchmark rather than the backbone: the three sizes sit almost on top of one another within each colour.

The share is what the mechanism says it should be: a deployment whose questions are all about the present has nothing to route, and the planner reveals that by never asking for the widest window. What makes it usable is that the planner is a text call, so this diagnostic costs one generation per question with no video decoded and no answers needed, and it can be run on a sample of live traffic before anything is deployed. It is a rule of thumb fitted on nine points rather than a validated predictor, and the share and the gain come from the same runs, so it should be read as an ordering rather than as a forecast of a number. Nine points is few enough to ask which of them carries the line: refitting with each setting left out in turn moves the zero crossing between 7.4% and 8.8% and the one-point mark between 15.0 and 17.4, so no single setting is holding it up.

B.12 Streams of hours, and where a memory should win

The widest window is a frame budget rather than a rate, so the further into a stream a question arrives the thinner it spreads, and Sec. 6 names this as a limit. It does not have to be argued on a stream nobody has: OVO-S-Bench already asks questions four and a half hours into a recording, its longest prefix being 16182 seconds, with a 90th percentile of 1144. Table 17 bands the questions by how much stream precedes them, and puts our OASIS reproduction beside the three windows, since a memory built over the prefix is built for exactly this case and the long bands are where it should overtake a window chosen per question.

Two things follow. The router is ahead in every band, including the last, where the uniform pass has thinned to a frame every 52.7 seconds and falls to 38.46 while routing holds 53.85 by sending most of those questions to a window that does not thin at all. And the memory does not pull ahead with length: past thirty minutes OASIS is 14.6 points behind the router and behind the uniform pass. Whatever structuring the history recovers, it is not what a question needs when the history is long.

Three caveats qualify the table. The last two bands hold 41 and 26 questions, so they locate a trend rather than settle a number. The columns are micro-averages inside each band and so are not

Table 17: **Every policy against how much stream precedes the question.** OVO-S-Bench at Qwen3-VL-8B, micro-average within each band over all 1722 questions. The OASIS row is at the budget its own method specifies. The last two rows give the router's mean frame count and how far apart the widest window's frames fall in that band, 128 frames divided by the prefix they cover.

Policy	all	under 2 min	2 to 10 min	10 to 30 min	30 to 60 min	over 1 hour
One window, held fixed						
The short window	49.01	50.31	45.91	53.91	43.90	53.85
The medium window	47.85	44.99	46.68	53.91	48.78	46.15
The widest window	50.00	43.76	50.38	57.55	53.66	38.46
A memory built over the prefix						
OASIS [CVPR'26]	46.92	42.13	46.16	55.21	39.02	50.00
Q-SPAN	**54.59**	**53.99**	**52.43**	**59.90**	**53.66**	**53.85**
questions	1722	489	782	384	41	26
Q-SPAN frames	44.6	20.2	51.0	60.2	52.5	68.2
the widest window, seconds between frames	—	0.4	2.5	7.4	18.7	52.7

Table 18: **A widest window that grows with the stream, on the questions where length bites.** Accuracy (%) on OVO-S-Bench at 8B, restricted to the questions arriving more than ten minutes and more than an hour into the stream. *Frames* is what the policy spends on those questions; no accuracy is defined where the prompt outruns the context.

Widest window	Frames	past 10 min (451)	past 1 h (26)
Uniform, 128 frames	128	56.10	38.46
Uniform, 256 frames	256	58.54	46.15
Uniform, 512 frames	512	— (201 unanswered)	— (all 26)
Q-SPAN, the window chosen per question	≈ 68	**58.98**	**53.85**

comparable with the four-level metric the main tables report. And a band is a population as much as a duration, since a four-hour recording is not a random draw of question types.

Would a budget that grows with the stream fix it? The thinning is a consequence of fixing the budget, so the obvious repair is to let it grow. Table 18 doubles it twice. The first doubling recovers part of what thinning costs and does not overtake routing, which spends about 68 frames where the doubled pass spends 256. The second returns no number at all: at 512 frames the prompt is about 301 000 visual tokens against this backbone's 262 000 context, and 201 of the 451 questions produce no answer. So a growing budget is a real mitigation with a ceiling rather than a substitute. It buys back some of the resolution that length took away, at four times the frames, still loses to choosing which stretch to spend them on, and runs out of context one doubling later, which is the sense in which a budget cannot be the answer to a stream that keeps growing.

B.13 WHAT THE GAIN BECOMES UNDER A DIFFERENT QUESTION MIXTURE

Sec. 6 says the gain depends on how a deployment's questions divide among the three regimes, and a benchmark fixes that division: OVO-S-Bench weights its four levels equally, so half of its questions have evidence out of view. A deployment has its own share. Table 19, plotted as Figure 3, re-weights the measured per-level outcomes to sweep that share from none to all, and re-chooses the fixed window at every mixture rather than at the one the benchmark rewards, the harder comparison: a practitioner serving only present-moment questions would tune the short window and keep it.

Routing is ahead of the best single window at every mixture up to a distant share of 90%. It is worth most, 3.85 points, near a share of two fifths, which is where the single window a practitioner would pick flips from the short one to the widest: the gain peaks where the choice is genuinely uncertain, not where the questions are hardest. That the peak sits at the handover is a property of the envelope rather than a finding; what the sweep measures is where the handover falls, at two fifths, and how far above it routing runs there. This is why Figure 3 plots the three fixed windows and not the gain alone. A single gain curve would be measured against a baseline that changes identity mid-sweep, so the peak would look like a property of the questions rather than of the choice; drawn as an envelope, the handover is visible and the gain is the distance to it. At the two extremes the right single window is nearly as good, by construction, and the method's cost there is bounded: 0.90 points of gain left on

Table 19: **The gain against the share of questions whose evidence is out of view.** OVO-S-Bench at 8B, its four levels re-weighted so that the two out-of-view levels carry the stated share and the two current-view levels the rest. *Fixed* is the best single window *at that mixture*, so the comparison is against a window tuned to the deployment. Frames are per question, averaged the same way as the accuracies. Figure 3 plots these rows.

Out-of-view share (%)	Short	Medium	Widest	Q-SPAN	Fixed	Gain	Frames
0	56.31	50.81	50.29	57.21	short	+0.90	22.6
25	52.02	48.98	50.07	55.07	short	+3.05	37.6
40	49.45	47.89	49.94	53.79	widest	**+3.85**	46.6
50 (the benchmark)	47.74	47.16	49.85	52.94	widest	+3.09	52.7
60	46.02	46.43	49.76	52.08	widest	+2.32	58.7
75	43.45	45.34	49.63	50.80	widest	+1.17	67.7
100	39.16	43.51	49.40	48.66	widest	−0.74	82.7

the table when every question concerns the present, 0.74 points of loss when every question reaches back, plus the two extra generations. The reading for a deployment is therefore not that a balanced mixture is required but that a lopsided one is cheap to be wrong about, and that the frame budget scales with the mixture rather than with the policy: the same router spends 23 frames a question at one extreme and 83 at the other.

B.14 WHERE THE THREE WIDTHS COME FROM, AND WHY NOT A NUMBER OF SECONDS

The three windows were taken from prior work rather than fitted here, and the text call names one of three rather than a duration. Both choices are arguments in Sec. 4.1; this section measures them. Six widths were run on each benchmark under the recipe every other run uses, so they differ in reach and in nothing else, and every cell below is the full question set.

Tables 20 and 21 say the same thing from two directions. Accuracy moves monotonically with reach in opposite directions on the two kinds of question, and it does so smoothly: the six widths trace one trade-off rather than mark three separable regimes. Within that trade-off the endpoints are not arbitrary. 4 frames is the only width that peaks on OVO-S-Bench's nearest level, and widening one step to 8 costs 6.04 points there; the uniform pass over the prefix is the only width that peaks on the two levels whose evidence lies furthest back, by 3.89 and 4.02 over 64 frames. The middle width has its evidence on OVO-Bench, where 32 frames peaks on action-sequence identification at 75.68 and loses both ways, 6.76 at 16 frames and 4.06 at 64.

A larger action space was also measured, since three is a choice and not a deduction. A six-way planner was written in the same terms, each read named by its own length, the five rules pointing at a band rather than at a single read so the model picks the rung inside it, and the eleven examples kept. Given six rungs it uses three: 56% of questions go to 4 frames and 34% to the uniform pass, while 8, 16, 32 and 64 frames take 2%, 0%, 5% and 4%, and on 87% of questions it lands on the rung the three-way planner would have chosen. It also scores lower for more frames, 52.38 with the probe at 49.0 frames a question against 52.94 at 44.6, and 51.83 without it against 52.38. The finer space is therefore not free and not used: the same wording that names one of three does not name one of six any better, which is the reading Sec. 4.1 gives of what the question carries.

The widths can also be rewired under the prompt rather than beside it. The planner names one of three reads and which window each name fetches is a deployment choice, so the ladder was rebuilt four ways and the routed score recomputed with the plan and the probe untouched: 8/64/128 gains 0.73, 4/16/64 gains 2.02 at 23.7 frames a question, 16/64/128 gains 0.54 and 16/128/128 gains 1.01, against 3.09 for the deployed 4/32/128. The prompt therefore transfers to ladders it was not written for, in the sense that routing is still ahead of the best fixed window on all four, and the deployed rung spacing is nonetheless the one that pays: every ladder that widens the shortest rung loses most of the gain, which is what the prompt's own wording predicts, since it describes that rung as the current instant and sixteen seconds is not one.

What the widths do not show is sensitivity to the middle boundary. On OVO-S-Bench the four-level averages at 16, 32 and 64 frames are 45.98, 47.16 and 47.98, a range of two points, so moving that boundary within a factor of four changes little. That is the honest reading of taking the value

Table 20: **Six window widths on OVO-S-Bench.** Accuracy (%) per level and their average, over all 1722 questions at each width. Widths are N frames at 1 fps ending at the query, except 128, which spreads them uniformly over the prefix. The method deploys 4, 32 and 128.

Frames	L1	L2	L3	L4	Average
4	**59.94**	39.63	52.69	38.69	47.74
8	53.90	38.52	51.61	36.13	45.04
16	50.08	43.15	50.90	39.78	45.98
32	49.28	48.70	52.33	38.32	47.16
64	48.33	51.85	52.69	39.05	47.98
128 uniform	46.10	**55.74**	**54.48**	**43.07**	**49.85**

Table 21: **Six window widths on OVO-Bench.** Accuracy (%) on the real-time macro and on the two backward-tracing subsets, over all 1282 questions at each width, under the same recipe as Table 20.

Frames	Real-time	EPM	ASI	Backward average
4	**82.34**	52.86	58.11	55.49
8	80.72	54.21	62.84	58.52
16	80.74	61.28	68.92	65.10
32	78.83	64.65	**75.68**	**70.16**
64	78.17	**68.01**	71.62	69.82
128 uniform	72.90	64.98	69.59	67.29

from prior work: the endpoints matter and are supported, the middle one is loose, and nothing here depends on having tuned it.

A planner that names seconds. Sec. 4.1 argues that a question fixes which regime it belongs to but not how far back its evidence sits, so a planner asked for a duration would be reporting how long ago things of that kind tend to happen rather than reading the question. Table 22 tests that on the same backbone, under the same system message, with the label space changed from three names to a number of seconds; the prediction is then served by the narrowest measured width that reaches at least that far, so the comparison needs no further answering runs.

It is worse on every level, by 2.92 points overall. The interesting part is how it fails: it predicts a median of five seconds and lands within 0.44 points of simply always using the widest window, so almost none of what it does is question-dependent. It also spends fewer frames than the three-way planner, 20.1 against 42.7, which makes it cheaper and worse rather than a point on a trade-off.

B.15 Reach against the frame count

Figure 1 widens the window by adding frames, so each curve in it moves two quantities together and the fall at wide windows could belong to either: the window reaching past the evidence, or the frames it now spends filling the context. The two separate cleanly, because the recent window takes a frame count and a rate and spans $(N-1)$ divided by that rate, so a target span and a count fix the rate between them. Table 23 holds 8 and 32 frames against spans of 8, 16, 32 and 64 seconds. Presentation, pixels per frame, decode and backbone are those of every other run, and the cell at 32 frames over 32 seconds reproduces the medium window's 47.16, which is that arm under another name.

A row holds the frame count and the tokens it costs and lengthens the window alone. Along the 8-frame row the nearest level loses 8.26 points while the level whose evidence sits half a minute back gains 7.96; along the 32-frame row they move by -2.86 and $+10.18$. Two populations move in opposite directions by eight to ten points with the budget held fixed, which is the claim of Sec. 2 with the quantity that confounded it no longer varying.

A column holds the window and quadruples the frame count. The four-level average moves by -0.07, $+0.16$, $+1.12$ and $+1.48$ across the four spans, so four times the visual tokens is worth at most a point and a half where lengthening the window at a fixed count moves a single level by eight. Frames are not free of effect: they help most where the window is widest and the sampling is

Table 22: **Naming seconds against naming one of three reads**, on OVO-S-Bench. A predicted duration is served by the narrowest width in Table 20 reaching at least that far. *Frames* is the mean per question.

Planner	L1	L2	L3	L4	Average	Frames
Names a number of seconds	55.96	48.15	54.12	39.42	49.41	20.1
Names one of three reads	**59.94**	51.85	**54.84**	42.70	52.33	42.7
and the probe behind it	**59.94**	**54.26**	54.48	**43.07**	**52.94**	44.6
Always the widest window	46.10	55.74	54.48	43.07	49.85	128.0

Table 23: **Reach and the frame count, varied one at a time.** OVO-S-Bench at 8B, per level and four-level average. The window's length is set by the sampling rate rather than by the frame count, so a row holds the budget and lengthens the window while a column holds the window and quadruples the budget. The cell at 32 frames over 32 seconds is the medium window.

Frames	Span	L1	L2	L3	L4	Average
8	8 s	**54.21**	40.00	50.54	34.67	44.86
8	16 s	50.72	44.81	50.18	**37.59**	45.83
8	32 s	48.97	48.89	51.25	35.04	46.04
8	64 s	45.95	47.96	**52.69**	35.77	45.59
32	8 s	51.19	40.93	51.25	35.77	44.79
32	16 s	49.76	45.37	**53.41**	35.40	45.99
32	32 s	48.17	49.44	52.33	**38.69**	**47.16**
32	64 s	48.33	**51.11**	51.97	36.86	47.07

therefore thinnest, and at the shortest span the extra frames cost the nearest level 3.02 points. That is sampling density, and neither it nor the budget is what the curves are read for.

The same experiment on OVO-Bench at 7B says it again in one line. Holding 8 frames and lengthening the window from 8 to 64 seconds costs 7.67 points of the six-subset macro, 79.11 against 71.44, and the subsets that ask about the present pay most, the worst losing 16.78 and the next 9.78. Holding the span instead and going from 8 frames to 32 changes that macro by -1.05 at 8 seconds and $+4.05$ at 64, negative where the window is already dense and positive where it is thin.

B.16 Why the probe escalates to the medium window

The probe only decides that the short window is insufficient; where an escalated question then goes is a separate choice. Table 24 replays the same planner and the same probe firings with that choice varied, and additionally removes the medium window altogether by sending both the planner's medium questions and the probe's escalations to the widest window. Escalating to the medium window is better than escalating to the widest in four of the five settings, and removing the medium window is

Table 24: **Where the probe should escalate to.** Accuracy under the benchmark's own metric, and the mean frames per question, with the planner and the probe's firing decisions held fixed. *No medium window at all* routes the planner's medium questions to the widest window as well. **Bold** marks the best of the three single windows on each benchmark, which is the baseline Sec. 6 compares routing against, and the best of the three escalation targets beneath them.

	OVO-S 8B		OVO-Bench 8B		OVO-Bench 27B		SB 8B		SB 27B	
Escalation target	Score	#Fr.	Score	#Fr.	Score	#Fr.	Score	#Fr.	Score	#Fr.
the short window alone	47.74	4.0	**82.34**	4.0	**85.03**	4.0	80.96	4.0	84.60	4.0
the medium window alone	47.16	32.0	78.83	32.0	80.08	32.0	**83.80**	32.0	**87.56**	32.0
the widest window alone	**49.85**	128.0	72.90	128.0	76.37	128.0	80.28	128.0	86.16	128.0
planner only, no probe	52.33	42.7	82.78	18.0	84.96	18.3	82.84	18.9	86.88	21.0
the medium window (ours)	52.94	44.6	**82.38**	19.7	**84.52**	19.5	**83.96**	20.1	**87.72**	22.4
the widest window	53.05	50.3	81.78	25.1	84.24	23.4	83.48	23.4	87.48	26.5
no medium window at all	**53.32**	58.9	80.88	54.6	84.36	53.6	82.88	32.5	87.48	37.7

Table 25: **What the probe adds, with the planner held fixed.** A paired bootstrap of routed minus the same plan without the probe, 4000 resamples of the question set per setting, same estimator per benchmark as elsewhere. The probe is worth most where the planner sends most questions to the short window and the evidence sometimes is not there; on OVO-Bench, where the questions are about the present and the short window is already the strongest single one, it is a tie.

Benchmark & backbone	Planner alone	+Probe	Gap	95% interval
OVO-S-Bench 7B	46.37	46.81	+0.45	[−0.50, +1.35]
OVO-S-Bench 8B	52.33	52.94	**+0.60**	[+0.05, +1.18]
OVO-S-Bench 27B	59.22	60.05	**+0.83**	[+0.44, +1.27]
OVO-Bench 7B	77.36	77.64	+0.28	[−0.99, +1.58]
OVO-Bench 8B	82.78	82.38	−0.40	[−1.11, +0.29]
OVO-Bench 27B	84.96	84.52	−0.44	[−1.12, +0.13]
StreamingBench RT 7B	82.96	84.88	**+1.92**	[+1.04, +2.84]
StreamingBench RT 8B	82.84	83.96	**+1.12**	[+0.64, +1.64]
StreamingBench RT 27B	86.88	87.72	**+0.84**	[+0.44, +1.28]

worse on four of the five while spending 70 to 200% more frames. OVO-S-Bench is the exception in both comparisons, which is consistent with its question mix: the planner sends 29% of its questions to the widest window there, against 9% on StreamingBench (Table 7), so on that benchmark pushing a borderline question further back is usually right.

What the probe is worth, with the planner's choices held fixed, is in Table 25: five of the nine settings clear zero, the three StreamingBench ones by 0.84 to 1.92, and the two OVO-Bench cells where it reads negative straddle it.

The one-step limit has a failure mode, and it is small. Counting the escalations the probe makes where the medium window is wrong and the widest one is right bounds what the limit costs directly: 15 questions of 1722 on OVO-S-Bench, 6 and 5 of 1282 on OVO-Bench at 8B and 27B, 3 and 2 of 2500 on StreamingBench, so between 0.08 and 0.87 points. Counting every short-routed question that only the widest window answers bounds the short window's whole share of the design rather than the probe's, and that is between 0.80 and 1.92 points. Both are smaller than what escalating to the widest window instead costs elsewhere in this table.

Escalating on the answer's own confidence instead. The probe asks a separate question about the frames. The obvious alternative asks the answering call itself: answer at the short window, and if the letter distribution is not confident enough, ask again at the medium one, then at the widest. That cascade needs no planner and no probe, so it is the adaptive baseline this design should be measured against. The letter distribution was recorded at each of the three windows on OVO-S-Bench at 8B, which makes the cascade a lookup at any threshold, and Table 26 sweeps it.

It does not reach the routed system at any threshold, and it does not reach a fixed window either. Its best setting scores 49.43 while spending 80 frames a question, against 49.85 for always taking the widest window at 128 and 52.94 for routing at 44.6; at a budget matched to routing it scores 47.9. The reason is visible in the last column: a threshold strict enough to catch the questions that need more history sends most questions all the way to the widest window rather than one step, since low confidence says the answer is uncertain and not how far back to look. Confidence is a signal about the answer; reach is a property of the recording, and the two are not the same quantity.

Answering at all three windows and combining afterwards does not rescue it either. A majority vote over the three answers reaches 47.06 while spending all 164 frames, below every single window, because two windows that are wrong outvote the one that is right; the policy that counts a question correct whenever any of the three answers it reaches 65.88, which bounds what choosing among these windows could ever be worth and is not a policy at all.

The operating point, as accuracy against frames. Read as a trade-off, the eight policies in this section and Table 24 trace a frontier on OVO-S-Bench at 8B: 47.74 at 4 frames, 52.33 at 42.7, 52.94 at 44.6, 53.50 at 49.6. The deployed system is on that frontier, and so is exactly one policy above it, the second probe at the medium window, which buys 0.56 points for five more frames. The two remaining variants are dominated: escalating to the widest window instead of the medium one reaches 53.05 at 50.3 frames and removing the medium window altogether reaches 53.32 at 58.9,

Table 26: **A confidence cascade over the same three windows.** OVO-S-Bench at 8B, four-level average, all 1722 questions. The answering call runs at the short window and is repeated at the next window whenever the letter distribution falls below the threshold, either on the top letter's probability or on its margin over the runner-up. *Frames* is the mean per question and the last column counts how many questions were asked again once and twice. The last block answers at all three windows and combines afterwards, so it spends every frame; *any window right* is not a policy but a bound on what choosing among these three could be worth.

Rule	Threshold	Score	Frames	Asked again once / twice
One window, held fixed				
The short window	—	47.74	4.0	—
The medium window	—	47.16	32.0	—
The widest window	—	49.85	128.0	—
Cascade on the top letter's probability				
top	0.50	47.63	8.1	74 / 40
top	0.80	47.69	34.8	156 / 392
top	0.90	47.83	46.1	163 / 548
top	0.99	49.00	74.5	140 / 947
Cascade on the margin over the runner-up				
margin	0.50	47.11	26.7	154 / 280
margin	0.80	47.92	45.0	165 / 532
margin	0.90	48.66	54.9	161 / 670
margin	0.99	49.43	80.1	134 / 1027
All three windows, answered and then combined				
majority vote	—	47.06	164.0	—
any window right	—	65.88	164.0	—
Q-SPAN, the window chosen per question	—	**52.94**	44.6	—

both paying more frames than the second probe for less accuracy. So the one-step design is a point on a frontier rather than a heuristic with nothing behind it, and the paper reports the cheaper of the two undominated points. A cascade that climbed further, one rung at a time through six widths, would be climbing rungs the planner does not ask for: given six it sends 2%, 0%, 5% and 4% of questions to the four it does not deploy (App. §B.14), so the steps such a cascade would add are steps between windows nothing selects.

A second probe, at the medium window. The design is asymmetric: the probe runs only on the short window, so a question the planner sent to the medium window is never reconsidered. Nothing about a two-option verdict forbids asking it again over the 32 frames the medium window supplies, and that arm was measured. It helps, by less than the first probe and for more frames: routed accuracy goes from 52.94 to 53.12 when the second verdict is asked only where the planner itself chose medium, and to 53.50 when it is asked wherever the pipeline holds a medium window, at 49.6 frames a question against the deployed 44.6. Of the 273 questions holding a medium window it calls 95 insufficient, and among those escalations 21 turn a wrong answer right against 9 the other way; a verdict that never erred would have 35 to find there. So the one-step design leaves about half a point on the table, which is the same order as the bound above and is bought at a tenth more frames. We report the one-step system because it is the one whose cost we characterise elsewhere in the paper, not because a second step is unavailable.

B.17 WHETHER THE PROMPT WAS FITTED TO WHAT IT IS TESTED ON

The planner is a prompt, so whether it was fitted to what it is tested on is the right question to ask, and the prompt has two kinds of content to ask it of. The rules and the examples are written about what a question refers to, in generic phrasings: they name no benchmark and no task label, and the phrasings they quote are ordinary English rather than any benchmark's templates, which can be checked against the prompt itself in App. §D.1. The three widths are not ours either, two of them being the recent window and the uniform pass prior systems converged on, with the middle one between them.

Table 27: **One wording, many recordings, and where its evidence sits.** Groups of identical wording among the 1669 OVO-S-Bench questions that carry an evidence interval, asked of at least five recordings, with every question's evidence at least 5 s back. *Spread* is the ratio of the largest distance in the group to the smallest.

Wording	Questions	Recordings	Nearest	Median	Farthest
at the current moment has this place appeared earlier in the video	62	34	9 s	114 s	706 s
at the current moment has this object appeared earlier in the video	19	14	7 s	202 s	450 s
compared with the last time you passed through this scene…	11	11	16 s	56 s	200 s
which option was shown for a longer duration in the video…	9	8	39 s	108 s	464 s
assuming the car is facing north at the beginning of the video…	16	16	112 s	243 s	603 s
assuming the car is initially facing north, what is the order…	8	8	223 s	317 s	1117 s
assuming the entrance faces due north, what is the order…	6	5	128 s	327 s	464 s
compared with the last time you passed through this scene…	9	9	72 s	100 s	252 s
assuming the car is initially facing north, what is the order…	7	7	172 s	252 s	396 s

One prompt, unchanged, serves all three benchmarks and all three backbones, and it is the wording in App. §D.1 in every run this paper reports. Wording that carried one benchmark's phrasing would predict a fall on the others; there is none. The routed system beats the published training-free systems on all three, and against the strongest single window it lands within half a point in either direction on the two whose questions concentrate on the present (Table 14). What there is instead of a fall is a tie, and Sec. 6 gives its reason, which is the share of questions on each side rather than the wording.

Three further pieces of evidence point the same way and are reported above: the decision survives being re-partitioned by evidence distance rather than by the benchmark's own levels (App. §B.10), it survives rewiring the three labels to other widths (App. §B.14), and it survives reversing the option order and deleting every demonstration (below). None of this rules out that a fourth benchmark would behave differently. It does rule out the mechanism the concern names, since the rules name none of the phrasings a benchmark would supply.

How much of the decision is the prompt's surface. The same question can be put to the wording rather than the content: do the planner's choices follow the question or the way the three reads are written down? Three variants were run on OVO-S-Bench at 8B, each changing one thing and nothing else: the three reads in reverse order, widest first, with the demonstrations reordered to match; the 11 demonstrations removed; and each read described in one clause instead of a sentence listing cases. The routed score is 53.11, 52.36 and 52.75 against the deployed 52.94, and the plans agree with the deployed one on 96, 92 and 95% of questions. Reversing the order helps, by 0.17, the opposite of what a positional preference would produce, and removing every demonstration costs 0.58. The decision is therefore not an artefact of option order or the examples; the examples buy about half a point.

B.18 The same wording, across the recordings it is asked of

Sec. 4.1 argues that a question fixes which kind of evidence it refers to but not how far back that evidence sits. That is a claim about a distribution, and OVO-S-Bench can be asked for it directly: group the 1669 questions that carry an evidence interval by their wording, keep the groups asked of several recordings, and read off the spread of distance inside each group. Table 27 gives the groups whose wording is identical, which are asked of at least five recordings, and whose evidence is at least 5 s back throughout, so that no group owes its spread to the annotation's floor.

Nine groups qualify and the spread inside one is 5× at the median. The largest is a sentence asked of 34 recordings whose evidence sits between 9 and 706 seconds back. A planner asked for seconds would have to answer that sentence with one number; a planner asked which of three reads it wants answers the part the sentence settles, and the probe is what looks at the rest.

B.19 What reading the question does that matching it does not

Table 2 replaces the planner with literal patterns transcribed from the rules its own prompt states, and that baseline recovers about a quarter of the distance from a shuffled plan to the planner. This section says where the rest of the distance is. On OVO-S-Bench at 8B the two agree on 1104 of

Table 28: **Naming the window from 4 frames, against naming it from the question.** OVO-S-Bench at 8B, all 1722 questions, four-level average. The router receives the short window's 4 frames and the question text with no answer options, and answers with one letter.

Policy	Average	Frames
The planner alone, from the wording	52.33	42.7
The planner and the probe, as deployed	**52.94**	44.6
The router alone, from 4 frames	50.39	32.9
The router, with the planner deciding where it says these frames suffice	50.81	42.8
The planner, with the router allowed to widen its choice	52.22	51.0

1722 questions and score identically there, 51.90 either way, so the entire gap lives in the 618 they disagree about: the planner reaches 57.44 on those and the patterns 45.63.

The disagreements are lopsided and their cause is legible. On 305 questions the patterns ask for the medium window and the planner keeps the short one, worth 61.64 against 48.20, and 234 of those 305 are fired by one branch: the words *left*, *put* and *place*, which the rules include for *where did I put X* and which also match *which objects are strictly to the left of the bin*. A pattern cannot tell a direction from a past tense; the planner is not matching the word but reading what the question refers to. On another 222 the patterns keep a question short and the planner sends it to the widest window, worth 53.60 against 41.89, and those carry no temporal keyword at all: *what is the background colour of the license plate on the parked black car that passes to the right* names no time, and its answer is nonetheless behind. The reverse direction is rare and is where the planner loses: on 15 questions the patterns choose the widest window and the planner the shortest, and there the patterns are ahead, 40.00 against 20.00.

B.20 A VERDICT THAT NAMES THE WINDOW, NOT A PLANNER THAT READS THE QUESTION

The probe is binary: it says whether the short window suffices, never how far back to go. The obvious stronger version asks the same frames to name one of the three windows outright, which would make the text planner unnecessary. Table 28 measures it. The prompt is the probe's, with the three reads described in the planner's own words and in the planner's order, and the frames are the same four the probe sees.

It is worse than reading the question, by 1.94 points as a replacement for the planner and 2.55 below the deployed system, while spending ten fewer frames. Neither composition rescues it: letting the planner decide the questions the router calls present-moment recovers little, and letting the router widen the planner's choice loses ground while spending six more frames a question. The two disagree on 35% of questions, and the disagreement is one-sided in an informative way. Where the planner sends a question to the widest window, the router calls the 4 recent frames sufficient a quarter of the time; the router sends 18% of questions to the widest window against the planner's 29%. Four recent frames show what is in them, not what is missing from them, so a verdict on them is well posed for sufficiency and badly posed for reach. That is the reason the probe is binary and the reach is decided from the wording.

C FULL COMPARISON TABLES

Each benchmark's published field by subtask, with the frame count each row uses: Table 29 for OVO-Bench and Table 30 for StreamingBench. OVO-S-Bench reports four levels rather than named subtasks, so its full field is already in the main text (Table 1a).

On OVO-Bench, hallucination detection scores whether a model declines an option the video does not support, which is robustness to unsupported claims rather than recall of what was seen (Shen et al., 2026), so the backward-tracing columns here are episodic memory and action-sequence identification only. It is also the one subset where reasoning inside the answer prompt helps rather than hurts, and it falls monotonically as the window widens (Table 10), where EPM and ASI move the other way, since a model shown more of the video is readier to name something it saw than to decline. Frame counts are not comparable across benchmarks: OVO-S-Bench's own protocol

gives every system 128 frames sampled from the prefix, while the OVO-Bench and StreamingBench collations report rates such as 1 fps, so that column is read within a table and never across them.

Table 29: **OVO-Bench.** Accuracy (%) on the perception and backward-tracing subsets; "—" is unreported. #Fr. is the query-averaged frame count. **Bold** is the best and underline the second best among open-weight systems.

			real-time visual perception							backward tracing	
Method	Params	#Fr.	OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI
Proprietary											
Gemini 1.5 Pro	—	1 fps	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.4
GPT-4o	—	64	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7
Open-weight offline video LLMs											
LLaVA-Video [TMLR'25]	7B	64	69.8	59.6	66.4	50.6	72.3	61.4	63.3	51.2	64.2
LLaVA-OneVision [TMLR'25]	7B	32	66.4	57.8	73.3	53.4	71.3	62.0	64.0	54.2	55.4
Qwen2-VL	7B	64	69.1	53.2	63.8	50.6	66.3	60.9	60.7	44.4	66.9
InternVL2	8B	64	68.5	58.7	69.0	44.9	67.3	56.0	60.7	43.1	61.5
LongVU [ICML'25]	7B	1 fps	55.7	49.5	59.5	48.3	68.3	63.0	57.4	43.1	66.2
Open-weight streaming, trained											
VideoLLM-online [CVPR'24]	8B	2 fps	8.1	23.9	12.1	14.0	45.5	21.2	20.8	22.2	18.8
Flash-VStream [ICCV'25]	7B	1 fps	25.5	32.1	29.3	33.7	29.7	28.8	29.9	36.4	33.8
Dispider [CVPR'25]	7B	1 fps	57.7	49.5	62.1	44.9	61.4	51.6	54.6	48.5	55.4
TimeChat-Online [ACM MM'25]	7B	1 fps	75.2	46.8	70.7	47.8	69.3	61.4	61.9	55.9	59.5
ThinkStream	3B	1 fps	85.2	64.2	69.8	49.4	69.3	64.1	67.0	53.9	59.5
StreamForest [NeurIPS'25]	7B	1 fps	68.5	53.2	71.6	47.8	65.4	60.9	61.2	58.9	64.9
StreamChat [arXiv'24]	7B	15 fps	—	—	—	—	—	—	50.1	50.1	54.0
StreamingVLM [ICLR'26]	7B	—	—	—	—	—	—	—	61.9	—	—
VST [ECCV'26]	7B	2 fps	80.5	55.1	72.4	55.1	76.2	64.1	67.2	56.9	64.9
Training-free, frozen Qwen2.5-VL-7B											
[Base] *Qwen2.5-VL*	7B	1 fps	83.2	52.3	70.7	51.1	69.3	60.9	64.6	50.8	56.8
InfiniPot-V [NeurIPS'25]	7B	—	—	—	—	—	—	—	65.9	—	—
FluxMem [CVPR'26]	7B	1 fps	—	—	—	—	—	—	67.2	—	—
SAVEMem [arXiv'26]	7B	1 fps	92.0	68.8	81.0	65.7	71.7	70.3	74.9	50.8	62.8
OASIS [CVPR'26]	7B	—	85.2	72.5	66.4	52.3	67.3	64.7	67.3	51.9	58.8
Q-SPAN (ours)	7B	30	94.0	71.6	80.2	64.0	76.2	79.9	77.6	57.9	62.8
Training-free, frozen Qwen3-VL-8B											
[Base] *Qwen3-VL*	8B	1 fps	79.9	61.5	71.6	57.3	69.3	62.0	66.9	58.6	70.3
OASIS [CVPR'26]	8B	—	92.0	80.7	81.0	67.4	67.3	79.9	78.1	62.0	60.1
Q-SPAN (ours)	8B	20	94.6	84.4	83.6	71.9	78.2	81.5	82.4	62.3	75.7
Training-free, frozen Qwen3.8-27B											
[Base] *Qwen3.8*	27B	1 fps	83.9	72.5	72.4	59.0	75.3	63.0	71.0	64.3	66.9
Q-SPAN (ours)	27B	20	**97.3**	**89.9**	**84.5**	**74.2**	**79.2**	**82.1**	**84.5**	**65.0**	**77.0**

D PROMPTS

Both prompts are given to the same backbone that answers the question. The planner receives the question text and no frames; the probe receives the frames the short window supplies and no answer options. The blocks below give the prompt text, re-flowed to this page width, since the prompts are sent as plain paragraphs and no line break in them is load-bearing.

Planner. The prompt names the three windows in physical terms rather than in frame counts, states the criterion (cover the evidence as narrowly as possible, since a window that is too wide spreads the same frame budget thinner), and gives five rules and eleven examples written in generic phrasings rather than any benchmark's templates.

Probe. The two-option form below is used throughout. A free-form "answer YES or NO" makes backbones that reason unprompted deliberate for hundreds of tokens and return nothing parseable, whereas the lettered form is the exact shape the answering prompt already emits a single token for. The question's own answer options are deliberately withheld, so that the probe cannot become a disguised answer attempt.

Table 30: **StreamingBench.** Accuracy (%) on real-time visual understanding (RT) and the three contextual-understanding tasks, anomaly (ACU), misleading (MCU) and sequential question answering (SQA). Their macro, CU-Avg, is quoted in the text rather than carried as a column. The proprietary and open-weight video LLM rows are quoted from Liang et al. (2026) and Shen et al. (2026); the streaming and training-free rows are quoted from each system's own paper, which is why several report real-time understanding alone, and InfiniPot-V reports it at the 4K cache budget it operates under. Our row covers the full subsets (2500 RT and 750 CU questions); its two frame counts are the means on RT and on the three CU tasks, in that order. **Bold** is the best and underline the second best over all systems. App. §B.5 breaks the real-time column out by its 10 tasks.

Model	Params	#Fr.	RT	ACU	MCU	SQA
Proprietary						
GPT-4o	—	64	73.3	41.2	38.4	32.8
Gemini 1.5 Pro	—	1 fps	75.7	51.4	40.7	**54.8**
Open-weight video LLMs						
Video-LLaMA2	7B	32	49.5	24.8	26.8	18.7
InternVL2	8B	16	63.7	32.0	31.2	32.3
Qwen2-VL	7B	64	69.0	31.2	26.0	39.6
LLaVA-OneVision [TMLR'25]	7B	32	71.1	35.6	36.0	27.3
Qwen2.5-VL	7B	0.5fps	66.0	33.2	30.1	34.0
Qwen3-VL	8B	0.5fps	72.8	35.6	35.7	43.9
Open-weight streaming, trained						
VideoLLM-online [CVPR'24]	8B	—	36.0	—	—	—
Flash-VStream [ICCV'25]	7B	—	23.2	—	—	—
StreamChat [arXiv'24]	7B	15 fps	60.1	28.8	22.4	32.0
Dispider [CVPR'25]	7B	—	67.6	—	—	—
TimeChat-Online [ACM MM'25]	7B	—	75.8	—	—	—
StreamForest [NeurIPS'25]	7B	—	77.2	—	—	—
VST [ECCV'26]	7B	2 fps	79.5	—	—	—
Training-free, frozen Qwen2.5-VL-7B						
[Base] *Qwen2.5-VL*	7B	1 fps	75.4	31.2	25.2	45.6
InfiniPot-V [NeurIPS'25]	7B	—	76.4	—	—	—
FluxMem [CVPR'26]	7B	1 fps	76.4	—	—	—
SimpleStream [arXiv'26]	7B	4	78.5	—	—	—
OASIS [CVPR'26]	7B	—	70.6	38.8	36.0	46.0
Q-SPAN (ours)	7B	35 / 12	<u>84.9</u>	61.2	<u>61.2</u>	52.0
Training-free, frozen Qwen3-VL-8B						
[Base] *Qwen3-VL*	8B	1 fps	75.4	41.2	29.2	48.4
OASIS [CVPR'26]	8B	—	78.2	42.7	49.6	48.4
SimpleStream [arXiv'26]	8B	4	80.6	—	—	—
Q-SPAN (ours)	8B	20 / 13	84.0	<u>64.0</u>	59.6	<u>53.6</u>
Training-free, frozen Qwen3.8-27B						
[Base] *Qwen3.8*	27B	1 fps	81.7	50.8	31.6	43.6
Q-SPAN (ours)	27B	22 / 22	**87.7**	**70.4**	**66.8**	50.8

D.1 PLANNER PROMPT

System message: *"You schedule the video frames a streaming assistant is allowed to look at. You see only the question, never the video. You answer with JSON and nothing else."*

```
A streaming video assistant has been watching a video and is at the moment
called NOW. It must answer the question below. Before it looks at any frames,
you choose which of three reads of the video it will receive. Every read ends
at NOW; they differ in how far back they reach and how densely they sample.

Pick the read that most narrowly covers the visual evidence needed. A read
that is too narrow misses the evidence; a read that is too wide spreads the
same number of frames thinner and loses detail.

The three reads:
```

```
- "short": a few frames of the last couple of seconds, at full detail. Use it
  when the evidence is the current instant: what is on screen now, where
  objects are right now, what something looks like, text visible now, where
  the assistant itself is, an action that is just finishing, something that
  just moved or just changed, or a short event that just played out.
- "medium": a few tens of frames covering the last tens of seconds. Use it
  when the evidence is in the recent past rather than at the current instant:
  something that just happened, the first time something appeared, or a
  comparison against an earlier state.
- "full": frames spread over the ENTIRE video so far, each one coarse. Use it
  when the evidence is spread across all of it, or lies further back than the
  recent past. Counting how many times something happened, or summarising
  the whole video, is of this kind: it needs every occurrence visited rather
  than one moment found.

Decide by asking WHICH MOMENT the question refers to, not when it is asked.
Five rules decide most cases:

R1. The question describes what is on screen at this instant, and the thing it
asks about is visible now -> "short".
R2. The question uses another event as its reference point ("after X", "before
X", "when X happened", "since X"). The answer lies around that event, not now
-> "medium". This holds whether the reference event belongs to the same ongoing
activity as now (a step in a task, one action in a sequence), or is an isolated
past occurrence, or is the FIRST time something happened: in every case the
evidence sits in the recent past, and a wider read would spread the frames too
thin to see it.
R3. The question asks what will happen next, or what someone is about to do.
Predicting requires seeing the activity that is under way, which takes time to
establish -> "medium". The current instant alone is not enough.
R4. The question asks where a specific object is, or what state it is in, and
that object may well NOT be on screen now ("where did I put X", "where is my
X"). Then the evidence is the last time the object was seen -> "medium".
Only use "short" if the question clearly concerns something
currently in view.
R5. The question asks about the LAYOUT of the space rather than about a
moment: the relative positions of places, or how they are oriented with
respect to one another, the order in which places or things were
encountered, the route taken, or anything needing a mental map of
everywhere that has been travelled through -> "full".

Examples:
Question: What colour is the mug on the desk?
{"read": "short"}
Question: Why did the person just stop walking?
{"read": "short"}
Question: How many times has the door been opened so far?
{"read": "full"}
Question: When the machine was switched on for the first time, what was next
to it?
{"read": "medium"}
Question: What does the person do after taking the lid off the pot?
{"read": "medium"}
Question: What happened just before the person picked up the drill?
{"read": "medium"}
Question: What is this person about to do?
{"read": "medium"}
Question: Where did I leave the scissors?
{"read": "medium"}
Question: Which way is the corridor turning right now?
{"read": "short"}
Question: Is the kitchen to the left or the right of the room I started in?
{"read": "full"}
Question: In what order did I pass the three shops on this street?
{"read": "full"}
```

```
Question: {question}
```

D.2 PROBE PROMPT

```
Look at the frames.

Question: {question}

Which is true?
A. These frames contain the visual evidence needed to answer the question.
B. Answering would require frames from earlier in the video.

Respond only with the letter corresponding to your chosen option (e.g., A, B).
```