CARDANO: TRAINING STREAMING VIDEO LANGUAGE MODELS TO KNOW WHEN THEY HAVE SEEN ENOUGH

Anonymous authors
Paper under double-blind review

ABSTRACT

Video language models (VLMs) answer a question about a video after processing all of it, so inference cost scales with video length rather than with the difficulty of the question. Processing the whole video is also not always beneficial, since additional video can overturn an answer that was already correct. The relevant question is therefore not how much video a model watches but when it should stop. We present CARDANO, a training formulation for streaming VLMs that learns to answer questions and to decide when further observation would not change the answer. Rather than predicting the content of unseen video, CARDANO predicts whether observing more video would change its current answer, which makes adaptive stopping a self-supervised problem: the target is obtained by comparing the model's answer on a partial video with its own answer on the full video, and requires no annotation or dataset-specific supervision. At inference the model reads the video window by window and, after each window, either commits to an answer or continues watching. A commit head of about forty parameters, fit once and deployed at its natural operating point with no threshold tuned on any dataset, outperforms a causal fixed budget at a matched number of frames on four of six benchmarks, with confidence intervals excluding zero, and transfers without refitting to a second architecture. Stopping preserves accuracy rather than improving it: with Qwen2.5-VL-7B, CARDANO is within 1.5 points of its whole-video accuracy or above it on all six benchmarks while processing between one third and four fifths of each video. On FunQA and Video-MME, where training the answerer on video prefixes raises accuracy, CARDANO exceeds the untrained backbone watching the whole video by 8.4 and 3.8 points. Code is available at `https://anonymous.4open.science/r/cardano-vlm-404A`.

1 INTRODUCTION

Figure 1 illustrates a simple but important failure case. Four windows into the video, both men are leaning toward the camera and concentrating, which is enough evidence to answer the question correctly. The video continues, the photograph is taken, and the men celebrate. By the end of the clip one of them is smiling, and a model that processes the entire video answers incorrectly that they were laughing. In this case additional video did not refine the prediction but overturned a correct one.

Today's video language models (VLMs) answer questions only after processing an entire video, so inference cost scales with video length rather than with question difficulty. Watching the whole video is also not always beneficial: predictions often stabilise long before the video ends, and additional video can turn a correct answer into an incorrect one. An oracle that answers from the best single window outperforms watching everything by 2.6 to 5.9 points on all six of our main benchmarks with Qwen2.5-VL-7B, after subtracting the gain expected from chance alternation between options (the same correction removes all of the headroom on TimeBlind, Section 4.3). Recent work has shown that current VLMs often under-utilise the visual information available in videos (Moezzi et al., 2026). Our observation is complementary: the question is not only how much video a model watches, but when it should stop.

The amount of video required to answer a question varies substantially across examples. Figure 2 shows that annotated evidence spans end anywhere from roughly one third of a video to nearly its

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.

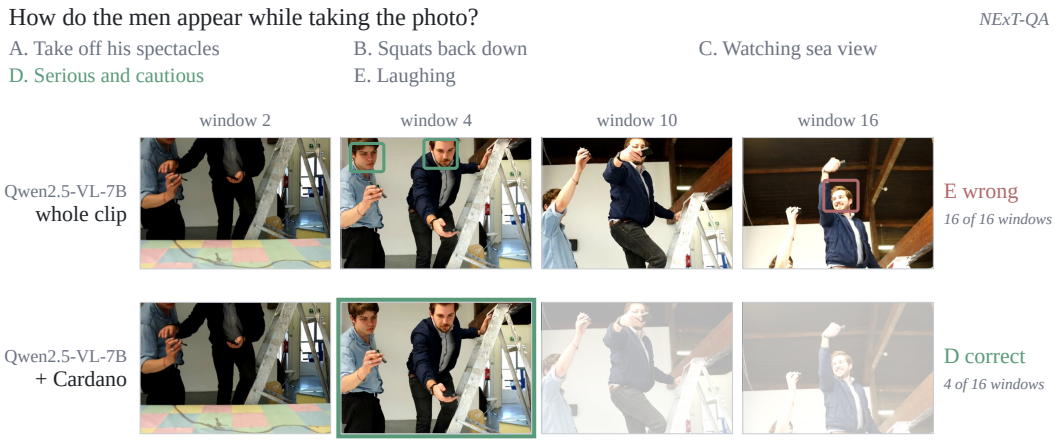


Figure 1: One question, one video, two inference strategies. Four windows in, the evidence is sufficient to answer correctly, and CARDANO commits. The same Qwen2.5-VL-7B backbone, processing the whole video, later sees one man smile and answers "laughing." Faded windows are skipped after the commit. The frames shown are representative snapshots; each window holds several frames and both models process all of them. Clip from NExT-QA (Xiao et al., 2021).

full duration. A fixed computation budget therefore wastes computation on easy questions while stopping too early on difficult ones. Non-causal frame selection uses the entire video to decide which frames to keep, which rules out streaming inference, and confidence-based stopping rules require thresholds tuned per dataset and estimate how confident the model is rather than whether more observation would change its answer.

We address this problem with CARDANO, a training formulation for streaming video language models in which the model processes a video sequentially and, after each window, predicts whether observing more video would change its current answer, committing without processing the remaining windows when it would not.

A natural alternative is to predict the unseen future and stop once it is unlikely to affect the answer. Section 3.2 shows that this is poorly matched to stopping: in the model's own representation, successive changes between windows have a cosine correlation of -0.43, close to the -0.5 expected of independent draws, so objectives that predict future content converge toward the average continuation, which carries little information about whether the answer will change.

CARDANO instead predicts answer stability directly. During training, each video is processed once to completion, and the model's prediction after every window is compared with its own prediction after the full video. This produces a binary target indicating whether the current answer has already stabilised, which requires no annotation, no temporal labels, and no dataset-specific supervision. A lightweight commit head is then fit on the trained model's own answer distribution, and the same policy is used on every dataset without tuning any stopping threshold.

Contributions.

1. We propose CARDANO, which learns when to stop watching by predicting whether more video would change the current answer. Its 40-parameter commit head is trained on the model's own predictions, with no annotation and no per-dataset threshold.
2. With Qwen2.5-VL-7B (Bai et al., 2025b), CARDANO outperforms a causal fixed budget by 1.3 to 3.7 points on four of six benchmarks and never falls more than 1.5 points below whole-video accuracy, while watching 31 to 80% of each video.
3. The head, fit once on Qwen2.5-VL-3B, transfers without refitting to Qwen2.5-VL-7B, to LLaVA-OneVision-7B ($+2.3$ to $+3.5$ points over the budget) and to three benchmarks it was not fit on.

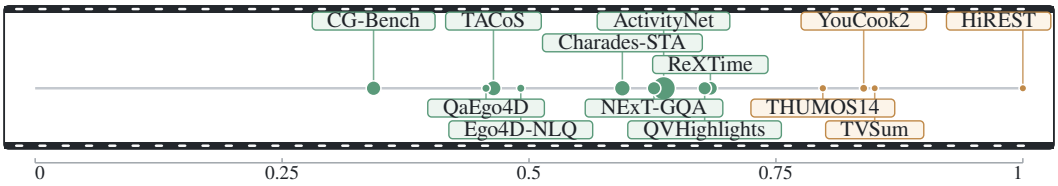


Figure 2: Where the annotated evidence ends, across 13 benchmarks with temporal annotations. Each pin marks the mean end of the evidence span as a fraction of video duration; bead size is the number of annotated questions. The variation across and within benchmarks is what a fixed budget cannot follow.

4. We evaluate on everyday activities, humour, long-form and expert video, physical reasoning (CLEVRER, +4.4 over the budget) and safety-critical dashcam footage (VRU-Accident, answered after 58% of the clip), and test the approach on robot and egocentric video (Appendices F and G).

2 Related Work

Streaming video language models. Streaming video language models process videos sequentially and respond while new observations continue to arrive. VideoLLM-Online (Chen et al., 2024a) learns to respond at appropriate moments in a continuous stream, Dispider (Qian et al., 2025) separates perception, decision, and reaction through an explicit triggering module, and StreamingBench (Lin et al., 2026) evaluates this interactive setting. These methods decide when to respond to an evolving stream; in our setting the question is fixed in advance, and the task is to decide whether the evidence seen so far suffices to answer it.

Frame selection for efficient video understanding. A large body of work reduces computation by selecting a subset of frames. SeViLA (Yu et al., 2023) localises question-relevant frames before reasoning, adaptive keyframe sampling (Tang et al., 2025) allocates frames according to estimated importance, and iterative methods (Zou et al., 2026b;a) progressively refine the selected subset. These approaches require the complete video before deciding which frames to keep and are therefore non-causal. CARDANO instead decides online whether more video needs to be processed at all, and we report non-causal selection only as a reference.

Adaptive computation and early exit. Adaptive inference allocates fewer operations to easier inputs. Early-exit networks attach intermediate classifiers (Teerapittayanon et al., 2016), learn exit policies for video (Ghodrati et al., 2021), stop once predictions stabilise (Zhou et al., 2020), or allocate decoding budgets in language models (Han et al., 2025; Schwartz et al., 2020). These methods reduce computation along the depth of a network with the input fixed; CARDANO allocates computation over the input itself, so the saving is video that is never encoded rather than layers that are never executed.

Confidence-based and calibrated stopping. Confidence thresholding is the standard approach for deciding when to stop collecting evidence. Sequential analysis (Bartholomew & Ghosh, 1972) provides the classical formulation, while conformal prediction (Angelopoulos & Bates, 2023) and confident adaptive language modeling (Schuster et al., 2022) calibrate confidence scores into stopping rules with statistical guarantees. Confidence estimates how certain the model is about its current prediction, whereas CARDANO predicts whether additional observations would change it, with no threshold calibrated or tuned per dataset; we compare against tuned confidence thresholds throughout.

Embodied agents and vision-language-action models. Embodied agents face the same question. The closest precedent for our decision rule is Ren et al. (2024), who stop an embodied agent's exploration once a conformal prediction set over a vision language model's answers contains a single option, with the trigger calibrated to a coverage level. DeeR-VLA (Yue et al., 2024) exits a vision-language-action model early when consecutive exits agree, a consistency test along network

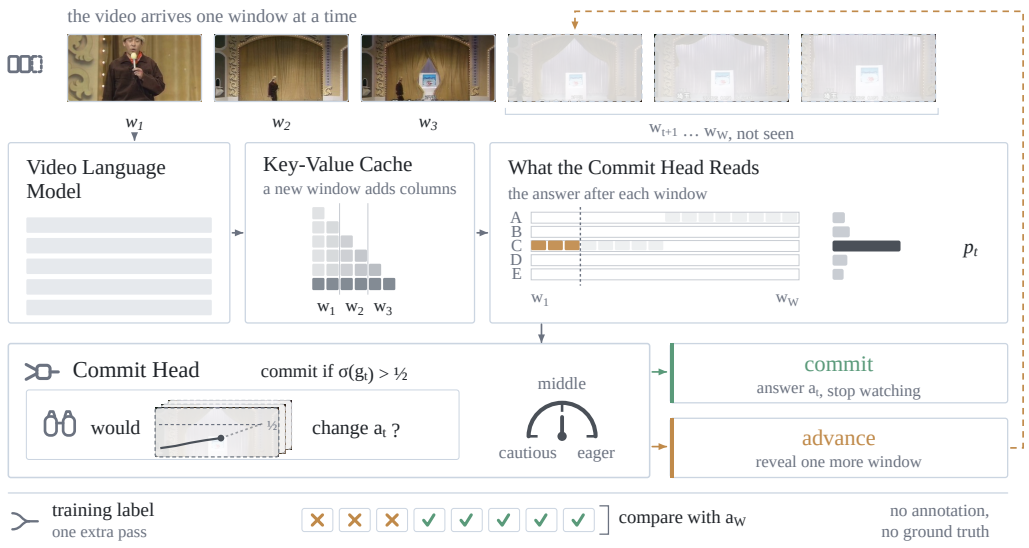


Figure 3: Overview of CARDANO. The model reads a video one window at a time and never re-encodes a frame. After each window the backbone produces an answer and the answer distribution, from which a lightweight commit head reads nine features. The head predicts whether more video would change the current answer, then commits or advances. Training supervision is generated automatically by comparing the current prediction with the prediction after the full video, which requires one additional pass and no human annotation.

depth rather than along the observation stream. We test our premise on DROID (Khazatsky et al., 2024) and BridgeData V2 (Walke et al., 2023) robot video and the decision on EgoPlan-Bench (Chen et al., 2023) and RoboFAC (Lu et al., 2025) (Appendices F and G).

Future prediction and anticipation. Predicting future events has long been studied in video understanding, including action anticipation, consequence reasoning in VisualCOMET (Park et al., 2020), and self-supervised predictive representations such as V-JEPA (Bardes et al., 2024). We show instead that the part of the future predictable from the current video contributes little to deciding whether the answer will change.

3 METHOD

3.1 STREAMING SETUP

Figure 3 gives an overview. A clip is divided into W consecutive windows of equal duration, each contributing a small fixed number of frames, so a window is a fixed share of the video rather than a fixed number of seconds, and watching every window means seeing the whole clip. At step t the model has seen windows $w_1, \ldots, w_t$ and produces a distribution $p_t \in \Delta^{C-1}$ over the C answer options, where Δ^{C-1} is the probability simplex and C varies by dataset. Its answer at that step is $a_t = \arg\max p_t$. Frames enter a key-value cache and are never re-encoded, so watching window t costs one encoding of that window's frames regardless of how many windows preceded it. We report accuracy at the window at which the method commits, and the *fraction watched*, t_{commit}/W.

3.2 WHY NOT PREDICT THE CONTENT OF THE REMAINING VIDEO

The direct approach to stopping is to predict the unseen remainder and check whether it matters. We first explain why this approach fails, since the reason motivates our target.

Let z_t be the pooled visual representation of window t and let $d_t = z_{t+1} - z_t$ be the change it produces. Across three backbones and five datasets we measure $\cos(d_t, d_{t+1}) \approx -0.43$. For a

sequence of independent draws around a fixed mean this correlation is exactly -0.5, so successive windows carry almost no information about one another. A predictor trained with any loss whose optimum is a conditional mean, which includes squared error and cosine objectives, therefore converges to the mean window of the clip. Splicing that mean into the stream in place of the real future changes the model's answer on essentially no questions, while a single real future window changes it on a fifth of them. Video is not unpredictable, but its predictable part is the part shared with the current window, which the model already has; the part that would change an answer is not predictable, and a better predictor does not remove this property of the data.

3.3 What the model predicts instead

A stopping decision does not require the content of the remaining video, only whether that content would change the answer, and we use this directly as the target. For each window t define

$$y_t \;=\; \mathbf{1}[a_t = a_W]\,, \tag{1}$$

where a_t is the model's answer after t windows and a_W is its answer after the full available stream. In words, y_t is one when committing now yields the same answer as watching everything, and zero when the remaining video would still move the answer. Figure 4 shows how a training pass produces these labels.

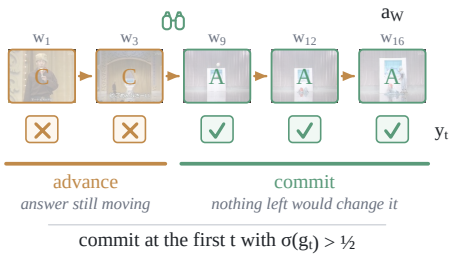


Figure 4: The self-supervised stopping target. In training the model answers after every observed window. A window is labelled commit if its answer already matches the answer after the full video, and advance otherwise; the labels require no human annotation. At inference only the observed prefix is available.

The target is a single number per window rather than a high-dimensional prediction, so it is not subject to the argument of Section 3.2. It is computed from the model's own behaviour in a pass that runs to the end of the stream, so it needs no annotation and no ground-truth answer. It is defined against the model's own final answer rather than the correct answer, which separates the two roles: the backbone is responsible for being right, and the head for recognising when the answer no longer depends on more video.

3.4 The commit head

The commit head reads a vector $\phi_t \in \mathbb{R}^9$ summarising the answer distribution and its recent movement: the two largest probabilities in p_t, the entropy of p_t normalised by $\log C$ so that it does not depend on the number of options, the probability the previous window assigned to the current answer, whether the answer changed from the previous window, how long the current answer has persisted, the position t/W, and the change in top probability and in total variation from the previous window. Every entry of ϕ_t is causal and none depends on C, which lets one head serve datasets with different numbers of options. Each feature is standardised with statistics computed once over the training corpus, giving $\tilde{\phi}_t$. The head is linear,

$$g_t^{(b)} \;=\; \mathbf{w}_b^{\top}\tilde{\phi}_t + c_b, \qquad b = 1,\dots,B, \tag{2}$$

where $B = 4$ is the number of budget heads described below, so the head has $B(9+1) = 40$ parameters. Because it reads only the answer distribution, it applies without modification to any backbone that produces a distribution over the options.

At test time the model commits at the first window where $g_t^{(b)}$ is positive, which is the natural operating point of a sigmoid at one half, and answers a_t. If no window triggers, it answers a_W. No threshold is selected on training data, on validation data, or per dataset.

3.5 Training the answerer and fitting the head

Training has two stages. The answerer is first fine-tuned to answer from a prefix of the stream rather than only from a complete clip. For a window k drawn uniformly from the first half of the stream, where the model will be asked to commit, the objective is

$$\mathcal{L}_{\text{ans}} \;=\; -\log p_{k+J}(y^{\star}) \;+\; \lambda\,\mathrm{KL}(p_k \,\|\, \mathrm{sg}[p_{k+J}])\,, \tag{3}$$

where $y^\star$ is the correct option, p_{k+J} is the distribution after the $J = 3$ windows that immediately follow k, sg stops the gradient, and $\lambda = 0.3$. The cross-entropy term is applied to the extended prefix rather than to p_k alone, which in our experiments gave a better answerer, and the consistency term pulls the prefix answer toward the answer the model gives once it has seen more of the clip.

The commit head is then fit once, offline, on the recorded per-window trajectories of the trained Qwen2.5-VL-3B answerer over the training splits of STAR, NExT-QA and FunQA, with the answerer frozen:

$$\mathscr{L}_{\text{head}} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{W}\sum_{t=1}^{W}\omega_b(y_t)\,\text{BCE}\Big(g_t^{(b)}, y_t\Big), \tag{4}$$

where BCE is binary cross entropy with logits and ω_b is the class weight defined below. This single head, fit on the outputs of the 3B model, is applied without refitting to every benchmark and to every backbone in Section 4, including Qwen2.5-VL-7B and LLaVA-OneVision-7B.

3.6 One knob for the whole trade-off

A stopping policy trades accuracy against cost, and the usual way to expose this choice is a threshold swept per dataset; we instead place the choice in the objective. Each budget head b has a weight κ_b on the positive class, $\omega_b(y) = \kappa_b$ if $y = 1$ and 1 otherwise, so a small κ_b penalises false commits and yields a cautious policy that watches more, while a large κ_b yields an eager one. The four heads are independent linear functions of the same standardised features and are fit together, so a single fit produces the entire trade-off curve rather than one point on it. Every head is deployed at the same natural operating point, and the same value of κ_b is used on every dataset.

3.7 What this costs

We measure cost as wall-clock time to answer one question, relative to a single pass over the whole clip. Frames are encoded once and windows after the commit are never encoded, but cost is close to linear in the windows watched and reading the answer distribution at every window adds overhead, so the fraction watched overstates the saving. On Qwen2.5-VL-3B, CARDANO costs 0.72 of a single pass on STAR and 0.44 on NExT-QA, 46 to 67% less than a streaming model that answers after every window but 29 to 35% more than a fixed budget watching the same windows; the case for deciding is accuracy at matched frames (Appendix J).

4 Experiments

4.1 Setup

The answerer is trained on STAR (Wu & Yu, 2024), NExT-QA (Xiao et al., 2021) and FunQA (Xie et al., 2025), and the commit head is fit on the training splits of the same three datasets. The resulting policy is evaluated unchanged on their evaluation splits, on MVBench (Li et al., 2023b), Video-MME (Fu et al., 2024) and Video-MMMU (Hu et al., 2026), and on four further benchmarks (Section 4.3). We apply the method to Qwen2.5-VL-3B, Qwen2.5-VL-7B (Bai et al., 2025b) and LLaVA-OneVision-7B (Li et al., 2025), each compared against its own backbone watching the whole clip. Every method except FrameExit, which was run externally, sees the same windows of the same clips. Baselines with a swept parameter, including patience, are read at the sweep point closest to CARDANO's fraction watched, so none gains by stopping later, while CARDANO is read at one global setting. Frame selection methods answer with our backbone and adapter, so only the choice of frames varies (Appendix H). Every dataset is screened for label-free shortcuts (Appendix A). CARDANO results are means over ten head-fit seeds with 95% item bootstrap intervals; further settings are in Appendix K.

4.2 Accuracy against video watched

Table 1 reports accuracy and the fraction of video each method watched on the three training benchmarks. At the same fraction, CARDANO outperforms the causal first-K budget on every dataset and backbone, with intervals excluding zero (+3.7, +1.3 and +1.9 points on Qwen2.5-VL-7B, +4.5, +1.6 and +1.6 on Qwen2.5-VL-3B), and remains within 1.1 points of whole-clip accuracy on STAR

Table 1: Accuracy and fraction of video watched on the three training benchmarks, grouped by backbone. All methods see the same frames, except FrameExit, which was run externally with its own learned gating module. Uniform sampling selects frames after seeing the whole clip and is a non-causal reference. Swept baselines are read at the sweep point closest to CARDANO's fraction and first-K at exactly that fraction; CARDANO uses one global setting, averaged over ten head-fit seeds (standard deviation at most 0.1 point). Other backbones watching the whole clip are listed for reference. Arrows give the direction of improvement; bold marks the best and underline the second-best accuracy among causal methods at a matched fraction within each backbone group; shaded rows are CARDANO.

	STAR		NExT-QA		FunQA	
Method	Acc ↑	Watched ↓	Acc ↑	Watched ↓	Acc ↑	Watched ↓
Qwen2.5-VL-3B (Bai et al., 2025b)						
whole clip	64.7	1.00	78.6	1.00	68.5	1.00
CARDANO	64.0	0.56	78.9	0.34	73.3	0.42
Qwen2.5-VL-7B (Bai et al., 2025b)						
whole clip	68.3	1.00	82.6	1.00	67.6	1.00
uniform sampling *(not causal)*	68.8	0.69	82.2	0.31	76.1	0.41
first-K windows	63.5	0.69	80.4	0.31	74.1	0.41
confidence threshold, tuned	<u>67.0</u>	0.68	<u>81.7</u>	0.31	<u>75.6</u>	0.42
CARDANO	**67.2**	0.69	**81.8**	0.31	**76.0**	0.41
LLaVA-OneVision-7B (Li et al., 2025)						
whole clip	51.6	1.00	80.0	1.00	65.8	1.00
first-K windows	<u>54.5</u>	0.65	<u>76.5</u>	0.22	<u>72.5</u>	0.37
CARDANO	**58.0**	0.65	**78.8**	0.22	**75.0**	0.37
Published methods, on Qwen2.5-VL-7B						
patience, tuned (Zhou et al., 2020)	63.7	0.62	80.8	0.34	74.8	0.43
SeViLA localizer, zero-shot (Yu et al., 2023)	67.7	0.69	81.9	0.31	75.3	0.41
adaptive keyframe sampling (Tang et al., 2025)	68.7	0.69	81.0	0.31	75.9	0.41
FrameExit (Ghodrati et al., 2021)	63.8	0.67	80.2	0.33	73.4	0.42
Other backbones, whole clip						
Qwen3-VL-8B (Bai et al., 2025a)	71.4	1.00	84.5	1.00	69.8	1.00
LLaVA-Video-7B (Zhang et al., 2024)	65.4	1.00	81.5	1.00	66.2	1.00
InternVL2.5-8B (Chen et al., 2024b)	69.2	1.00	83.1	1.00	68.0	1.00

and NExT-QA while watching 0.69 and 0.31 of the video with Qwen2.5-VL-7B. On FunQA it exceeds the untrained whole-clip backbone by 8.4 points; this gain comes from training the answerer on prefixes, and the stopping rule retains it at 0.41 of the video. CARDANO also matches or exceeds the per-dataset tuned confidence threshold and the patience rule (Zhou et al., 2020) without any tuning, and outperforms FrameExit (Ghodrati et al., 2021), run with its learned gating module, by 1.6 to 3.4 points at similar fractions. The non-causal frame selectors land within 1.1 points of one another (Appendix H).

Figure 5 shows the full trade-off; all four budget heads come from the same fit and are read at their natural operating point.

The head, fit on the outputs of Qwen2.5-VL-3B, transfers to a different architecture without refitting. On LLaVA-OneVision-7B it outperforms a matched first-K budget on all three training datasets with intervals excluding zero (+3.5 on STAR, +2.3 on NExT-QA and +2.4 on FunQA), and it exceeds LLaVA's own untrained backbone watching everything by 6.4 points on STAR and 9.1 on FunQA. The head is deliberately not refit per backbone, since the claim under test is transfer.

4.3 TRANSFER TO BENCHMARKS IT WAS NOT FIT ON

Table 2 applies the policy unchanged to three benchmarks that neither the answerer nor the head was trained on. On Qwen2.5-VL-7B, CARDANO is at or above whole-clip accuracy on all three while

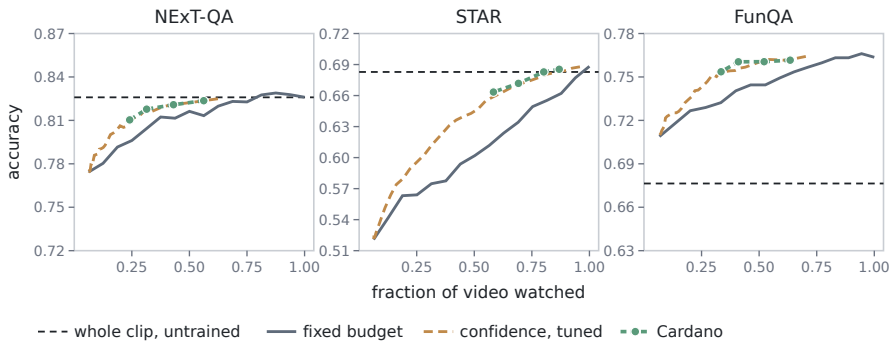


Figure 5: Accuracy against the fraction of video watched, for one learned policy. Each marker is a budget head at its natural operating point, with no per-dataset threshold. Grey is a fixed budget; orange is a confidence rule tuned per dataset. CARDANO lies above the fixed budget, and on FunQA above the untrained whole-video backbone on less than half of the video.

Table 2: The same policy on three benchmarks it was not fit on, with nothing refit or retuned, grouped by backbone, with arrows, bold, underline and shading as in Table 1.

	MVBench		Video-MME		Video-MMMU	
Method	Acc ↑	Watched ↓	Acc ↑	Watched ↓	Acc ↑	Watched ↓
Qwen2.5-VL-3B						
whole clip	65.9	1.00	59.6	1.00	48.9	1.00
CARDANO	62.8	0.58	60.3	0.59	49.3	0.83
Qwen2.5-VL-7B						
whole clip	64.2	1.00	61.3	1.00	49.8	1.00
uniform sampling *(not causal)*	66.2	0.61	65.8	0.59	51.6	0.80
first-K windows	63.9	0.61	62.6	0.59	49.6	0.80
confidence threshold, tuned	65.2	0.62	63.2	0.58	**51.1**	0.81
CARDANO	**65.5**	0.61	**65.1**	0.59	51.0	0.80
LLaVA-OneVision-7B						
whole clip	58.1	1.00	60.8	1.00	33.7	1.00
first-K windows	53.3	0.62	55.4	0.56	33.7	0.83
CARDANO	**54.4**	0.62	**56.8**	0.56	**34.4**	0.83

watching 0.59 to 0.80 of the video, and outperforms the first-K budget on Video-MME by 2.5 points with an interval excluding zero. On LLaVA-OneVision-7B the gains over first-K remain positive but span zero, and CARDANO is 3.7 to 4.1 points below LLaVA's whole-clip accuracy on MVBench and Video-MME, so there the cross-architecture result concerns cost rather than accuracy.

We evaluate four further benchmarks: TimeBlind (Li et al., 2026), VRU-Accident (Kim et al., 2025), CLEVRER (Yi et al., 2019) and MotionBench (Hong et al., 2025) (Table 5 in Appendix C). On CLEVRER CARDANO outperforms the first-K budget by 4.4 points with Qwen2.5-VL-3B. On VRU-Accident, a safety-critical benchmark of dashcam collisions, it exceeds the untrained whole-clip backbone on all three backbones ($+5.0$, $+4.9$ and $+3.8$ points) while committing after 0.55 to 0.58 of the video (Figure 9). TimeBlind is a negative case: its oracle headroom equals the gain expected from chance alternation between its two options, and with Qwen2.5-VL-3B CARDANO matches the first-K budget ($+0.0$, $[-1.7, +1.7]$) while losing 3.7 points to the whole clip.

4.4 WHAT EACH PART CONTRIBUTES

We remove one design choice at a time (Table 6 in Appendix D). Each variant is scored against CARDANO at the variant's own fraction watched, obtained by sweeping CARDANO's threshold, so a variant that merely stops earlier gains nothing. Predicting the content of the remaining video instead

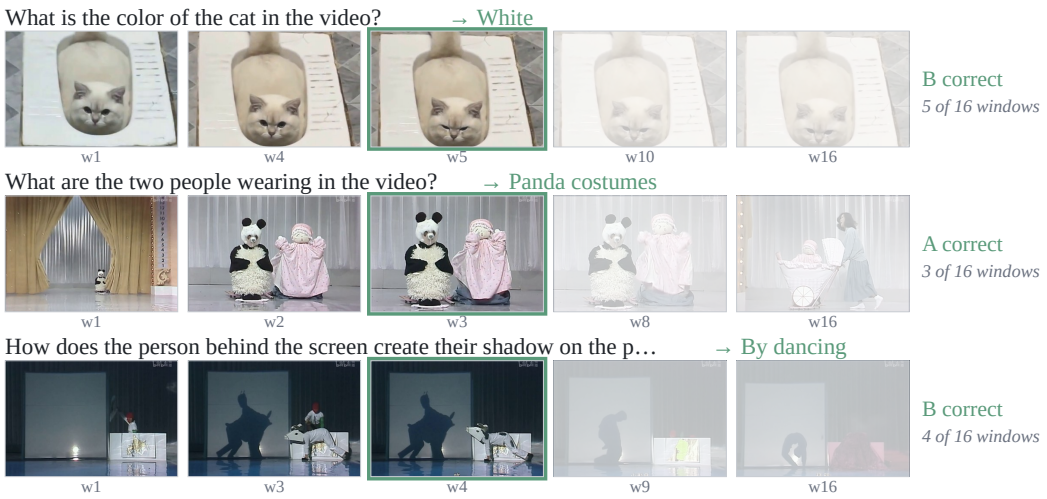


Figure 6: Adaptive stopping on individual clips. The box marks the window where CARDANO commits; faded windows are never processed. In each case the information the question requires is visible at the commit window, and the rest of the video does not change the answer.

of answer stability costs 4.8 points, as Section 3.2 predicts. Attaching the head to the untrained backbone instead of the trained answerer costs 1.9 points on STAR and 6.9 to 11.4 on FunQA, but nothing on NExT-QA at either scale, so how much the answerer benefits from training depends on the dataset. Feeding the head another clip's trajectory costs 4.0 points on STAR and is a loss with an interval excluding zero in all six dataset and backbone combinations, so the policy reads the clip at hand rather than applying a generic stopping habit.

Where it stops. The head typically commits after three to five of sixteen windows, once the content the question asks about is visible (Figure 6; more in Appendix E). The main failure is the converse: a stable answer that a later window would have changed, a case the self-agreement target cannot flag.

5 CONCLUSION

Predicting what unseen video will show does not support stopping, because its predictable part is the part the model already has. CARDANO instead asks whether the rest of the video would change the answer, a binary target computed from the model's own answers at no annotation cost. One untuned 40-parameter policy outperforms a causal fixed budget at matched frames on four of six benchmarks and on a second architecture, while accuracy remains near whole-clip performance.

The answerer was trained on a shorter window budget than it is evaluated on, a mismatch we have not removed, and because the commit target is agreement with the model's own final answer, watching everything is the best the head can match; a target defined against correctness is the natural next step.

A second direction is embodied agents and vision-language-action models, where committing early means acting sooner. Robot video satisfies the premise (Appendix F); on egocentric planning (Chen et al., 2023) the unrefit head transfers in direction but costs 2.7 points (Appendix G), and robot failure analysis (Lu et al., 2025) first requires an answerer trained on robot footage. Closed-loop control has no fixed final answer, so the target must be redefined, for example over a task-level question.

REPRODUCIBILITY STATEMENT

The code for the streaming loop, answerer training, trajectory recording and commit-head fitting, with the commands that reproduce the shipped configuration, is available at `https://anonymous.4open.science/r/cardano-vlm-404A`. The streaming setup and windowing are specified in Section 3.1 and Appendix K, the answerer objective and head fit in Equations 3 and 4, the baseline protocols in Appendix H, and the dataset screens in Appendix A. All benchmarks are public; we release no video data.

ETHICS STATEMENT

VRU-Accident contains real dashcam recordings of collisions; we use it only as recorded question answering data, and our results concern when an answer becomes available, not intervention in a deployed system. The EgoPlan clips derive from Ego4D and were obtained through a public redistribution that does not carry Ego4D's licence agreement; we do not redistribute them. We report the label-free shortcuts we found in FunQA and MotionBench (Appendix A).

AI USE STATEMENT

We used generative AI tools to edit the paper for readability and consistency of style, to format and compile the LaTeX source, including tables and figure placement, and to check and format bibliography entries against their published versions. We have not used generative AI tools to generate data sets, formulate claims, or propose hypotheses. All AI-assisted edits were reviewed by the authors, and every number in the paper was checked against the output of our own evaluation scripts. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Shuai Bai, Yuxuan Cai, Rui-Zhe Chen, Ke-qin Chen, Xiong-Hui Chen, Zesen Cheng, Liang-Hao Deng, Wei Ding, Rongyao Fang, Chang Gao, et al. Qwen3-VL technical report. *arXiv.org*, 2025a. doi: 10.48550/arXiv.2511.21631.

Shuai Bai, Ke-qin Chen, Xue-Jing Liu, Jia-Lin Wang, Wenbin Ge, Sibo Song, K. Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv.org*, 2025b. doi: 10.48550/arXiv.2502.13923.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Trans. Mach. Learn. Res.*, 2024. doi: 10.48550/arXiv.2404.08471.

D. Bartholomew and B. K. Ghosh. Sequential tests of statistical hypotheses. *International Statistical Review / Revue Internationale de Statistique*, 40(2):250, 1972. doi: 10.2307/2286949.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *Computer Vision and Pattern Recognition*, pp. 18407–18418. IEEE, 2024a. doi: 10.1109/CVPR52733.2024.01742.

Yi Chen, Yu-Ying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Rui-Lan Xu, Shan Ying, and Xihui Liu. EgoPlan-Bench: Benchmarking multimodal large language models for human-level planning. *International Journal of Computer Vision*, 134(3):118, 2023. doi: 10.1007/s11263-025-02676-0.

Zhe Chen, Weiyun Wang, Yue Cao, Yang-Zhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv.org*, 2024b. doi: 10.48550/arXiv.2412.05271.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Computer Vision and Pattern Recognition*, pp. 24108–24118. IEEE, 2024. doi: 10.1109/CVPR52734.2025.02245.

Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. FrameExit: Conditional early exiting for efficient video recognition. In *Computer Vision and Pattern Recognition*, pp. 15603–15613. IEEE, 2021. doi: 10.1109/CVPR46437.2021.01535.

K. Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Computer Vision and Pattern Recognition*, pp. 18973–18990. IEEE, 2021. doi: 10.1109/CVPR52688.2022.01842.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1274.

Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. MotionBench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *Computer Vision and Pattern Recognition*, pp. 8450–8460. IEEE, 2025. doi: 10.1109/CVPR52734.2025.00791.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Xiang Yue, Bo Li, Yuanhan Zhang, and Ziwei Liu. Video-MMMU: Evaluating knowledge acquisition from multidisciplinary professional videos. In *Annual Meeting of the Association for Computational Linguistics*, pp. 27798–27828. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1281.

Alexander Khazatsky, Karl Pertsch, S. Nair, Ashwin Balakrishna, S. Dasari, Siddharth Karamcheti, Soroush Nasiriany, M. K. Srirama, L. Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, 2024. doi: 10.15607/RSS.2024.XX.120.

Younggun Kim, Ahmed S. Abdelrahman, and Mohamed Abdel-Aty. VRU-Accident: A vision-language benchmark for video question answering and dense captioning for accident scene understanding. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 772–782. IEEE, 2025. doi: 10.1109/ICCVW69036.2025.00085.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In *Neural Information Processing Systems*, volume 34, pp. 11846–11858, 2021.

Baiqi Li, Kangyi Zhao, Ce Zhang, Chancharik Mitra, Jean de Dieu Nyandwi, and Gedas Bertasius. TimeBlind: A spatio-temporal compositionality benchmark for video LLMs. *arXiv.org*, 2026. doi: 10.48550/arXiv.2602.00288.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025. doi: 10.48550/arXiv.2408.03326.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023a. doi: 10.48550/arXiv.2301.12597.

Kunchang Li, Ya-Li Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Computer Vision and Pattern Recognition*, pp. 22195–22206. IEEE, 2023b. doi: 10.1109/CVPR52733.2024.02095.

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. *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151, 2026. doi: 10.1109/icassp55912.2026.11463959.

Weifeng Lu, Minghao Ye, Zewei Ye, Tao Lin, Shuo Yang, and Bo Zhao. RoboFAC: A comprehensive framework for robotic failure analysis and correction. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.12224.

Shayda Moezzi, Umer Saleem, Andong Deng, Chen Chen, and Sarah Ostadabbas. Position: Video LLMs must not ignore the pixel dynamics in plain sight. In *Forty-third International Conference on Machine Learning Position Paper Track*, 2026.

Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. VisualCOMET: Reasoning about the dynamic context of a still image. In *European Conference on Computer Vision*, pp. 508–524. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58558-7_30.

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 *Computer Vision and Pattern Recognition*, pp. 24045–24055. IEEE, 2025. doi: 10.1109/CVPR52734.2025.02239.

Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, 2024. doi: 10.15607/rss.2024.xx.089.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems 35*, pp. 17456–17472. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-1269.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. The right tool for the job: Matching model and instance complexities. In *Annual Meeting of the Association for Computational Linguistics*, pp. 6640–6651. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.593.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Computer Vision and Pattern Recognition*, pp. 29118–29128. IEEE, 2025. doi: 10.1109/CVPR52734.2025.02711.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, pp. 2464–2469. IEEE, 2016. doi: 10.1109/ICPR.2016.7900006.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 2023. doi: 10.48550/arXiv.2308.12952.

Bo Wu and Shoubin Yu. STAR: A benchmark for situated reasoning in real-world videos. In *NeurIPS Datasets and Benchmarks*, 2024. doi: 10.48550/arXiv.2405.09711.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *Computer Vision and Pattern Recognition*, pp. 9772–9781. IEEE, 2021. doi: 10.1109/CVPR46437.2021.00965.

Binzhu Xie, Sicheng Zhang, Zitang Zhou, Bo Li, Yuanhan Zhang, Jack Hessel, Jingkang Yang, and Ziwei Liu. FunQA: Towards surprising video comprehension. In *Lecture Notes in Computer Science*, pp. 39–57. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-73232-4_3.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2019.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *Advances in Neural Information Processing Systems 36*, volume 36, pp. 76749–76771. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-3354.

Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. DeeR-VLA: Dynamic inference of multimodal large language models for efficient robot execution. In *Advances in Neural Information Processing Systems 37*, pp. 56619–56643. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-1803.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Trans. Mach. Learn. Res.*, 2024.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT loses patience: Fast and robust inference with early exit. In *arXiv (Cornell University)*, volume 33, pp. 18330–18341. Cornell University, 2020. doi: 10.48550/arxiv.2006.04152.

Junbo Zou, Ziheng Huang, Shengjie Zhang, Liwen Zhang, and Weining Shen. VideoBrain: Learning adaptive frame sampling for long video understanding. *arXiv.org*, 2026a. doi: 10.48550/arXiv.2602.04094.

Yuanhao Zou, Shengji Jin, Andong Deng, Youpeng Zhao, Jun Wang, and Chen Chen. A.I.R.: Enabling adaptive, iterative, and reasoning-based frame selection for video question answering. In *arXiv.org*, 2026b. doi: 10.48550/arXiv.2510.04428.

A SHORTCUT SCREEN

Before using a dataset we check whether its questions can be answered without the video. Table 3 scores three rules that use no model and no frames: choosing the most frequent option letter, the longest option, or the shortest option. If one of these approaches the accuracy we report, the dataset does not measure what we claim. The screen also changed a decision in this work: it disqualified a candidate training corpus whose correct answer was the longest option two thirds of the time.

Table 3: Accuracy of three label-free rules that use no model and no frames, against per-item chance. On FunQA and MotionBench the longest option is correct about 15.5 points above chance.

Dataset	Chance	Letter	Long	Short
STAR	25.0	35.1	27.9	24.6
NExT-QA	20.0	20.6	20.6	22.1
FunQA	20.8	27.8	36.5	13.8
MVBench	28.4	29.2	29.4	24.6
Video-MME	25.0	28.7	27.5	23.2
Video-MMMU	11.4	17.9	17.0	10.8
TimeBlind	50.0	50.0	50.0	50.0
CLEVRER	50.0	52.1	47.3	52.7
MotionBench	25.5	27.7	41.0	17.5
VRU-Accident	25.0	28.1	24.0	23.0

The screen flags two of our benchmarks. On FunQA the longest option is correct 36.5% of the time against a per-item chance rate of 20.8%, a margin of 15.7 points and the largest in the set; MotionBench shows a similar margin of 15.5 points. This does not affect the comparisons we draw, because every method answers the same questions from the same options and our FunQA result is measured against the same backbone watching the whole clip rather than against chance, so the shortcut is available to both sides of every comparison. It does mean that absolute accuracy on FunQA and MotionBench overstates how much of the task requires the video.

B CHOOSING A SAFETY-CRITICAL BENCHMARK

The motivation for stopping early is strongest where waiting has a cost, so we searched for a benchmark in that setting. Table 4 lists the candidates. Two requirements did most of the filtering. The method reads the model's distribution over answer options after each window, so a benchmark must consist of video with a fixed option set, and every candidate is screened for label-free shortcuts before we report accuracy on it. A third requirement is that some current model scores above chance, since a benchmark on which no model does so provides no information about stopping.

Table 4: Benchmarks considered for a safety-critical evaluation. The method requires video and a fixed option set, since the head reads the distribution over answer options. We also screen for label-free shortcuts and require that some model answers the benchmark above chance.

Benchmark	Modality	Format	Labelled n	Outcome
VRU-Accident	video	4-way MCQ	1076	used, after curation
SUTD-TrafficQA	video	2–4-way MCQ	30092	clean, not yet staged
SurgMLLMBench	frames	open-ended + seg.	0	no video, no MCQ
MedStreamBench	video	mixed	182	media not redistributed
Safe/Unsafe Behav.	video	8-class	691	leak $+.179$, no stopping signal
AccidentBench	video	2–12-way MCQ	17069	published SOTA below chance

C ADDITIONAL BENCHMARKS

Table 5 reports four benchmarks beyond the six in the main text, evaluated with the same policy and nothing refit. VRU-Accident and CLEVRER both clear per-item chance with an interval excluding zero, which we require before reporting a benchmark. TimeBlind is included as a negative case: on a two-option task, a model whose answer alternates between the options is correct by chance about half of the time, and its oracle-over-windows headroom is entirely accounted for by this effect. Section 4.3 summarises the results.

Table 5: Four further benchmarks. VRU-Accident contains real dashcam recordings of collisions involving vulnerable road users, a setting where a late answer has little value. TimeBlind has no exploitable stopping signal: its oracle headroom (+10.4) equals its chance-alternation null (+10.4). The first-K row is read at the Qwen2.5-VL-3B CARDANO fraction; bold, underline and shading are as in Table 1, applied to the two Qwen2.5-VL-3B methods. On CLEVRER CARDANO is +4.4 over the first-K budget with an interval excluding zero, but 9.1 points below the whole clip. On VRU-Accident it is above the untrained whole-clip backbone on all three backbones (+5.0 on Qwen2.5-VL-3B with an interval excluding zero), a gain attributable to answerer training, while the 3B difference from the first-K budget is within noise. On TimeBlind and MotionBench both differences from the first-K budget are within noise.

	TimeBlind		VRU-Accident		CLEVRER		MotionBench	
Method	Acc ↑	Watched ↓	Acc ↑	Watched ↓	Acc ↑	Watched ↓	Acc ↑	Watched ↓
Evaluated n	740		331		407		1220	
Qwen2.5-VL-3B, whole clip	58.4	1.00	54.9	1.00	87.6	1.00	52.4	1.00
Qwen2.5-VL-7B, whole clip	62.3	1.00	58.2	1.00	91.4	1.00	57.5	1.00
LLaVA-OneVision-7B, whole clip	53.8	1.00	52.4	1.00	84.1	1.00	51.2	1.00
First-K on Qwen2.5-VL-3B	**54.7**	0.60	<u>59.0</u>	0.58	<u>74.1</u>	0.36	<u>54.0</u>	0.70
CARDANO on Qwen2.5-VL-3B	**54.7**	0.60	**59.9**	0.58	**78.5**	0.36	**54.4**	0.70
CARDANO on Qwen2.5-VL-7B	56.8	0.65	63.1	0.55	84.5	0.38	58.0	0.68
CARDANO on LLaVA-OneVision-7B	51.2	0.62	56.2	0.56	77.8	0.39	52.1	0.68

Table 6: Ablations on STAR with Qwen2.5-VL-7B. Δ is against CARDANO at the variant's fraction watched. *Content target* predicts the remaining video's content; *frozen backbone* uses the untrained model; *foreign state* reads another clip's trajectory. Bold marks the best and underline the second-best variant.

	Variant	Acc ↑	Watched ↓	Δ ↑
	CARDANO	**67.2**	0.69	**0.0**
	content target	61.8	0.58	−4.8
	frozen backbone	62.4	0.47	<u>-1.9</u>
	foreign state	<u>63.5</u>	0.69	−4.0

D FURTHER ABLATIONS

Table 6 summarises the ablations on STAR with Qwen2.5-VL-7B, and Table 7 reports the two measured ablations on every dataset and backbone.

E FURTHER QUALITATIVE EPISODES AND FAILURE CASES

Figure 7 shows one episode from each of the six main benchmarks, read as in Figure 6, and Figure 8 shows questions on which committing cost the correct answer. The failures are not spread evenly over question types. They concentrate on counting, locating, ordering and comparing across time, that is, on questions whose answer depends on the whole clip, where an answer that has been stable for several windows is consistently wrong rather than settled. The self-supervised target cannot flag this case, since it asks only whether the current answer matches the final one.

Figure 9 shows a safety-critical episode. On a dashcam clip from VRU-Accident the hazard is visible in the first window and CARDANO commits at the third of sixteen windows, so the answer is available while the clip is still playing. This concerns when the decision becomes available, not intervention in a deployed system.

Table 7: The two measured ablations on all six dataset and backbone combinations. Δ is against CARDANO's own accuracy at the variant's fraction watched, with CARDANO's commit threshold swept to spend that fraction; values are means over ten head-fit seeds, and the parentheses give the number of seeds whose 95% item bootstrap interval excludes zero. Foreign state is a loss with an interval excluding zero in every cell on all seeds. Frozen backbone is such a loss on all seeds in three cells, on half of the seeds on Qwen2.5-VL-7B STAR, and null on NExT-QA at both scales, where the untrained model is already close to its own ceiling.

Backbone	Dataset	Acc ↑	Watched ↓	Frozen backbone Δ	Foreign state Δ
Qwen2.5-VL-3B	STAR	64.1	0.56	−3.4 (10/10)	−4.5 (10/10)
Qwen2.5-VL-3B	NExT-QA	78.8	0.34	−0.9 (0/10)	−2.7 (10/10)
Qwen2.5-VL-3B	FunQA	73.4	0.42	−6.9 (10/10)	−2.7 (10/10)
Qwen2.5-VL-7B	STAR	67.2	0.69	−1.9 (5/10)	−4.0 (10/10)
Qwen2.5-VL-7B	NExT-QA	81.7	0.32	−0.1 (0/10)	−2.2 (10/10)
Qwen2.5-VL-7B	FunQA	76.1	0.41	−11.4 (10/10)	−3.2 (10/10)

F WINDOW INDEPENDENCE BEYOND THIRD-PERSON VIDEO

Section 3.2 rests on the measurement that successive windows of a video are close to independent draws, so that a predictor trained with a mean-seeking loss converges to the average window. That measurement was taken on third-person web video. Embodied observation streams are smoother and driven by the agent's own motion, and if they behaved differently the argument would not apply to them.

We therefore repeated the measurement on robot video. On the DROID wrist and exterior cameras (Khazatsky et al., 2024) and on BridgeData V2 (Walke et al., 2023) (98 to 100 episodes each), consecutive-window displacement correlations remain strongly negative. Three of five valid streams satisfy the pre-registered criterion $\cos(d_t, d_{t+1}) \leq -0.40$ (DROID wrist -0.41; Bridge -0.41 at both 2 fps and its native 5 Hz), the two DROID exterior cameras are inconclusive at -0.38, and none approaches the threshold of -0.25 at which content prediction would become viable. This includes the wrist camera, where ego-motion dominates and a positive correlation was most plausible. The correlation is more negative in high-motion windows, so the result is not an artefact of a stationary arm. Cells at the native 15 Hz rate are excluded because they fail the pre-registered representation-health bound: $\cos(z_t, z_{t+1})$ is 0.86 to 0.93 against a limit of 0.85, since two-frame windows 0.13 s apart are near duplicates. Values are for Qwen2.5-VL-7B; the 3B model agrees within 0.01, and the positive control reproduces -0.427 on STAR against the -0.43 reported in Section 3.2.

The premise of Section 3.2 is therefore not specific to third-person video. This does not imply that CARDANO transfers to robot control: the commit target is agreement with the answer after the full stream, and a control policy has no such fixed answer, since its action is meant to change with the scene. The natural extension is to embodied tasks with a terminal answer.

G TRANSFER TO EGOCENTRIC PLANNING

Appendix F shows that the premise behind the method holds beyond third-person video. A separate question is whether the commit decision itself transfers. We evaluated it on EgoPlan-Bench (Chen et al., 2023), 2,116 egocentric clips that ask what a person should do next to reach a stated goal, with four options each. The head was not retrained; the same pooled head is applied unrefit at the same global operating point.

This benchmark is adversarial for a stopping rule by construction. The answer is the next action given the current state, so the decisive evidence is in the last frame: the frozen backbone scores 20.3 on the first window against a chance rate of 25, and 36.1 on the whole clip, so a rule that stops early is expected to lose accuracy.

The head largely defers its decision on this benchmark. It watches 0.894 of the stream, and 64.6% of items do not commit before the final window, which is the behaviour the benchmark calls for. It is 1.4 points above a matched causal budget, though this interval ($[-0.0, +2.8]$) does not exclude zero,

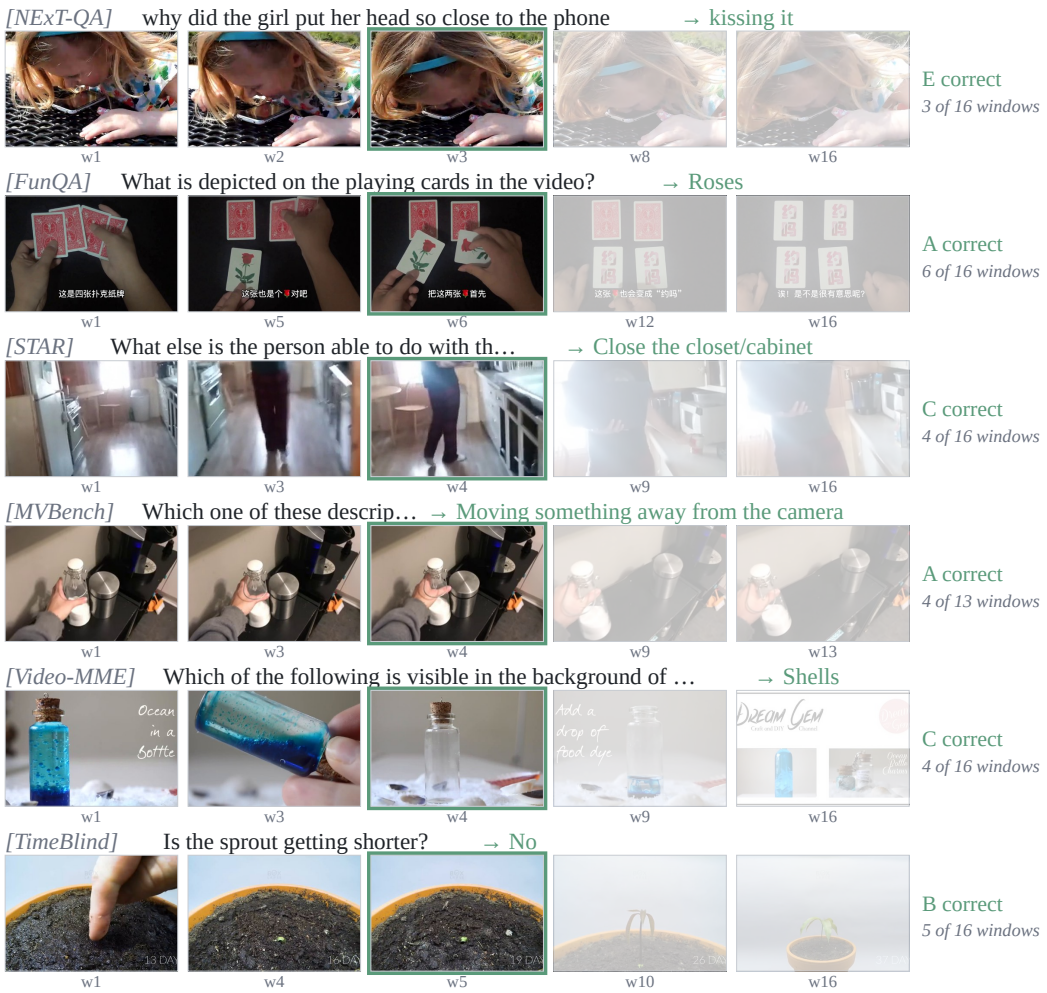


Figure 7: One episode from each benchmark, read as in Figure 6: the box marks the commit window and faded windows are never processed. The arrow gives the correct option. In each case the evidence the question asks about is on screen at the commit window, and the rest of the video does not overturn it.

and the tenth of the stream it does save costs 2.7 points against watching everything ($[-3.6, -1.8]$). We searched for a subset on which it matches whole-clip accuracy while still saving computation, pre-registering the subsets (benchmark level, effective frame rate and clip-duration terciles) and the criteria before computing any of them, and found none; the two subsets that satisfy the criteria do so only where the answerer is already at chance. A sweep over the budget weight shows a monotone trade-off with no operating point that escapes it: the setting closest to matching whole-clip accuracy watches 99.3% of the video.

Figure 10 shows three episodes in which the head stops early and is correct. In each, the commit coincides with the moment the scene determines the next action; for example, *pour meat on plate* is answered as the plate reaches the pan, five windows before the clip ends.

The decision therefore transfers in direction but not without cost, subject to two limits. The fine-tuned answerer does not transfer, scoring 29.4 against a chance rate of 25, so the numbers above use the frozen backbone with the head fit on frozen trajectories. In addition, this is human egocentric video with a planning question rather than embodied control. The setting we would test next is one in which the answer becomes determined partway through an episode, such as robot failure identification.

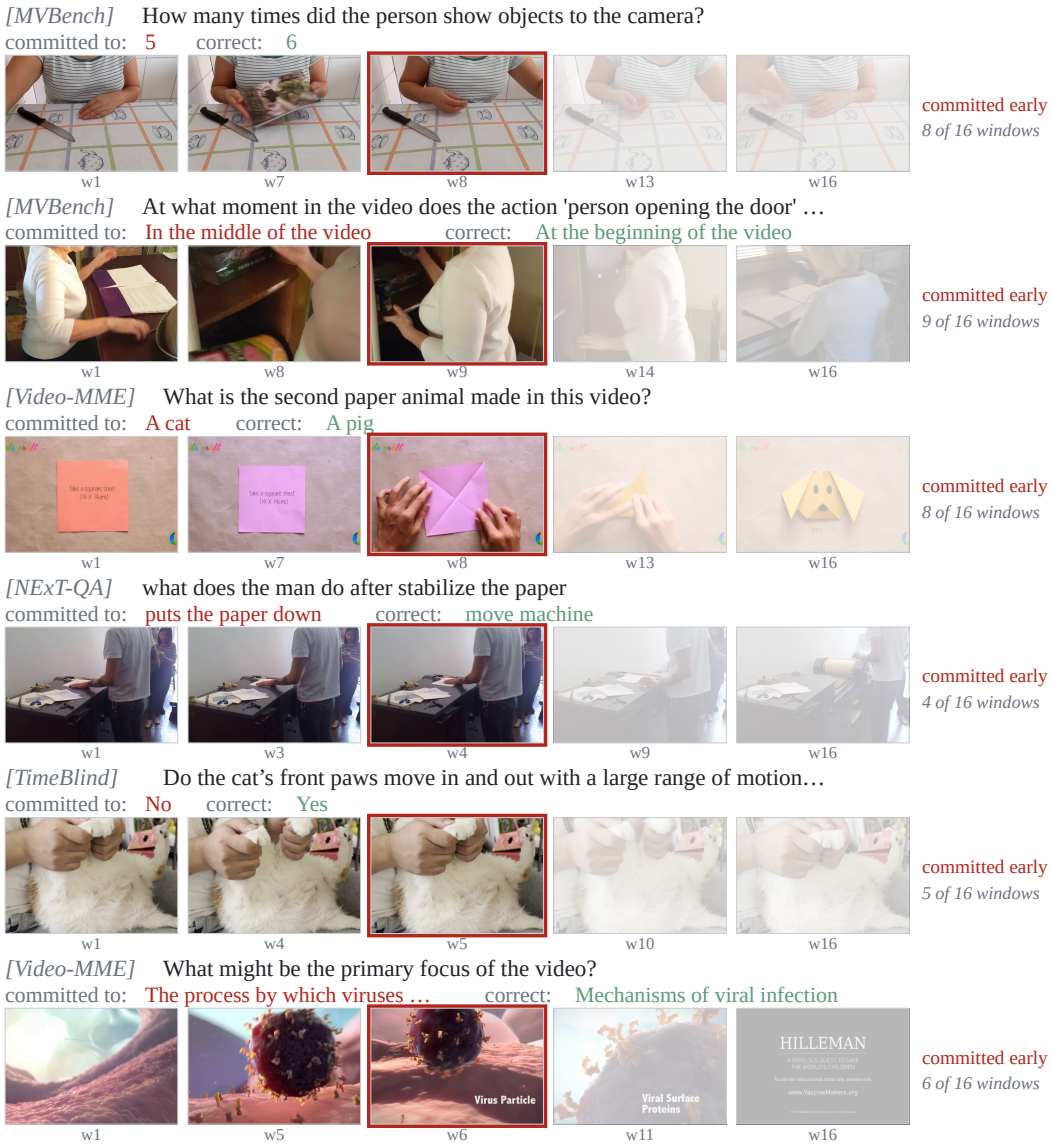


Figure 8: Failure cases, selected to illustrate that they are systematic. Each is the mode of Section 4.4: the answer had been stable for several windows, the head read that as settled and committed, and the rest of the video would have corrected it. Committing costs the answer on 1.3–5.2% of questions depending on the benchmark. The six share a kind of question: counting (*how many times*), locating (*at what moment*, *during which part*), ordering (*the second* paper animal, *after* the paper is stabilised) and comparing across time (*at first … and later*). For these the answer is a function of the whole clip, so early stability does not imply sufficiency: the model is consistently wrong. A target defined against the final answer cannot identify this case.

We did not find a benchmark for that setting that supports the test without first fine-tuning an answerer on robot footage. On RoboFAC (Lu et al., 2025), the two tasks whose evidence would lie mid-stream fail label-free screens. Failure *locating* draws its distractors from other tasks, so choosing an option from the video's own task vocabulary scores 0.82 on simulation and 0.89 on real footage without any view of the failure, and failure *detection* is answered correctly 80 to 90% of the time by replying "no". This leaves failure identification, which we class-balanced on the gold label ($n = 474$, fewer than the 606 the annotations describe, because the official test split references videos



Figure 9: A safety-critical episode from VRU-Accident, read as in Figure 6. A pedestrian is already crossing in the first window; CARDANO commits at the third of sixteen, marked with the hazard sign, and answers correctly. The thirteen faded windows are never encoded. Beyond the computation saved, the answer is available while the clip is still playing rather than after it ends. Committing early costs the answer on 0.6% of VRU-Accident questions, the lowest rate of any benchmark we measured. We measure question answering on recorded dashcam video, not intervention latency in a deployed system, so this illustrates decision timing rather than collision avoidance.

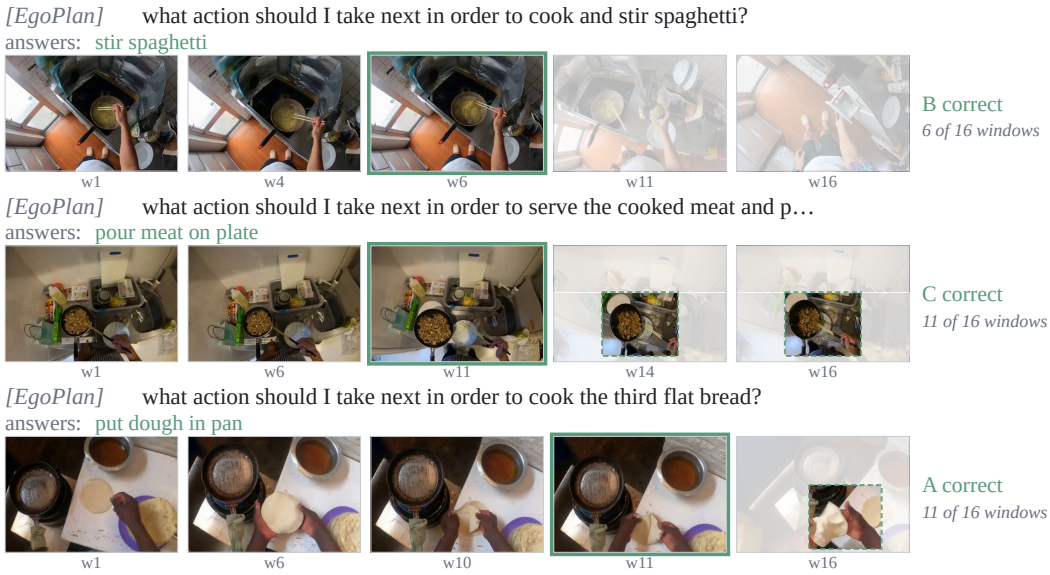


Figure 10: Three EgoPlan episodes, read as in Figure 6: the head commits at the boxed window and the faded windows are never encoded. In each case the commit lands at the moment the scene makes the next action determined. In the middle row the model answers *pour meat on plate* at window 11, as the plate is carried to the pan and five windows before the clip ends; in the bottom row it commits once the dough has been worked into a round ready for the pan. The dashed regions show what those skipped windows contained (the pour, the dough going in), which is the event the answer had already named. The top row is the same behaviour in a different guise: the model commits at window 7 while still at the pot, and the last quarter of that clip leaves the stove altogether for another appliance, so the nine windows it does not watch carry no evidence about the question. This is the benchmark whose design works hardest against stopping, since the question asks what to do *next*; aggregate results are reported in Appendix G.

absent from the public repository, concentrated in one simulation class). On this task Qwen2.5-VL-7B is at chance at every window (33.6 against a chance rate of 33.3 on the real three-way split, from the first window to the sixteenth), so the head's early stop costs nothing only because the model has no information, and the comparison is uninformative. A second screen shows why balancing is not sufficient: error type is largely predictable from the task alone, with a task-conditional majority rule scoring 0.63 and 0.53 against chance rates of 0.17 and 0.33, so identification inherits a milder form

of the shortcut that disqualifies locating. A meaningful test would require training an answerer on robot footage and balancing within task as well as within class. We also note that RoboFAC's published training and test splits are disjoint in questions but not in videos: 3,872 of the 6,152 videos carrying training identification items also appear in the test annotations, so fine-tuning on the official training split contaminates the official test set. A video-disjoint split leaves roughly 2,257 training items, all from simulation.

The clips are from Ego4D (Grauman et al., 2021), obtained through a public redistribution that does not carry Ego4D's own licence agreement. The benchmark supplies the current observation as a separate image, which we omit because in a stream the final window already contains that frame. Sampling is sparse on the long tail, with 16% of clips below 0.5 effective frames per second, which partly explains the whole-clip accuracy of 36.1.

H FRAME SELECTION BASELINES

The FrameExit row uses its learned gating module (Ghodrati et al., 2021), which decides after each window whether to stop observing. Three rows in Table 1 choose frames after seeing the whole clip: uniform sampling, adaptive keyframe sampling and the SeViLA localizer. Each receives the same thirty-two frames that every other row sees, keeps K of them, and answers in a single pass through the same backbone and adapter, so the only difference from CARDANO is which frames are kept. Each is read at CARDANO's fraction on that dataset, interpolating linearly between the two nearest budgets.

Because a budget is undefined for a clip with fewer than K frames, the selection rows are computed on the clips long enough for every budget: 2,342 of 2,400 on STAR, all 2,400 on NExT-QA and 2,123 of 2,400 on FunQA. On these same clips, CARDANO scores 67.4, 81.7 and 75.8 with Qwen2.5-VL-7B, and the first-K budget at the same fraction scores 63.8, 80.4 and 73.6, so the comparison in Table 1 is unchanged on the common subset. Before any of these numbers is used it must pass an identity check: at $K = 32$ a selector returns every frame in temporal order, so its answer must match the streaming whole-clip answer item by item. Agreement is 99.96 to 100.00% on all nine dataset and method combinations.

Three details concern the SeViLA row. Only the QVHighlights-pretrained localizer (Lei et al., 2021) is public; the self-refined per-dataset localizers of Yu et al. (2023) were not released, so the row uses the zero-shot localizer and is labelled as such. Its candidates are our thirty-two stream frames rather than its own decoding of the video. It is re-implemented on BLIP-2 (Li et al., 2023a) without the LAVIS library; the weight mapping is verified by tensor identity against the checkpoint's own answerer, and the published prompt, truncation, frame prefix, yes-logit and top-k-then-sort selection are reproduced.

The three selectors land within 1.1 points of one another on every dataset: choosing frames by relevance does not outperform spacing them evenly, so what the non-causal group gains is coverage rather than selection.

I SUPERVISING THE ANSWER AND THE COMMIT DECISION FOR ACCURACY

A natural question is whether training the answerer and the commit decision for accuracy, rather than for self-agreement, would improve accuracy as well as cost. We ran this experiment on Qwen2.5-VL-3B; the result is inconclusive.

The treatment supervises the answer at every window with the gold label, changes the commit target from stability to correctness, and lets the commit loss reach the backbone. The control is the recipe of Section 3. The two differ in three settings at once, so the contrast tests the combination rather than any single change. Both were trained on the same 3,000 items with identical segment boundaries, two seeds each, and evaluated on $n = 2{,}400$ items per dataset, with the frozen trajectories verified to be bit-identical across arms. Table 8 reports the accuracy difference, treatment minus control.

Signs agree across seeds on all three datasets, but every effect is smaller than the difference between two identically configured runs in this setup, which is 0.8 to 0.9 points. Only two of the six cells exclude zero, and they point in opposite directions. The experiment is therefore underpowered:

Table 8: Accuracy difference (points) between accuracy-supervised training and the recipe of Section 3, with 95% intervals, for two training seeds.

Dataset	Seed 0	Seed 1
STAR	-0.98 $[-2.04, -0.29]$	-0.17 $[-0.77, +0.75]$
NExT-QA	$+0.47$ $[-0.21, +1.00]$	$+0.67$ $[+0.17, +1.46]$
FunQA	$+0.73$ $[-0.36, +1.12]$	$+0.28$ $[-0.46, +0.96]$

Table 9: Cost of answering one question on STAR, relative to one pass over the whole clip, for Qwen2.5-VL-3B. A wall-clock profile measured on 120 timed STAR items is applied to the stopping distribution of the same policy; $\rho = 0.56$ is the fraction of windows watched. The answer pass was timed on four-option items, so on five-option datasets the cost of CARDANO is slightly understated. Bold marks the lowest and underline the second-lowest cost.

	Method	Frames encoded	Extra parameters	Cost ↓
	whole clip, one pass	W	none	1.00
	whole clip, streaming	W	none	1.34
	frame selection	$W + K$	a selector	1.46
	first-K windows	ρW	none	**0.54**
⏱	CARDANO	ρW	one head	<u>0.72</u>
	CARDANO, LLaVA-OV-7B	ρW	one head	0.81

separating a half-point effect from a 0.85-point run-to-run spread would require on the order of ten seeds. Accordingly, the accuracy claim in this paper is that accuracy is preserved, which we measure, and not that it is improved.

J COMPUTE ACCOUNTING

Table 9 applies a wall-clock timing profile, measured on STAR, to the stopping distribution of the Qwen2.5-VL-3B policy, so every row rests on one model and one set of stopping points. Each frame is encoded once into the key-value cache and windows after the commit are never encoded, but the head must read the answer distribution at every window it watches; one answer pass costs 39% of one advance pass. Because the cost of a window grows only mildly with the context already in the cache, cost is close to linear in the number of windows watched, which is why the fraction watched is not the cost ratio.

Applied to each dataset's own stopping points, the same profile gives a cost relative to one whole-clip pass of 0.72 on STAR, 0.44 on NExT-QA, 0.52 on FunQA, 0.77 on Video-MME, 0.74 on MVBench and 1.09 on Video-MMMU, where the head commits after four fifths of the windows and the per-window overhead exceeds the cost of the skipped windows. On STAR, CARDANO on LLaVA-OneVision-7B costs 0.81, and frame selection, which must encode the whole clip before selecting, costs 1.46. Against a model that must be able to answer at any moment, and so runs the answer pass at every window, committing early is 46% cheaper on STAR and 67% cheaper on NExT-QA. Against a fixed budget that watches the same number of windows and answers only at the last one, CARDANO is 29 to 35% more expensive, because that baseline makes no decision. All figures in this appendix are for Qwen2.5-VL-3B.

K TRAINING DETAILS

Windowing. Every clip is divided into $W = 16$ equal windows of two frames each, or fewer for clips shorter than about four seconds, where the frame rate is capped. A method that watches the whole clip therefore sees thirty-two frames, and one window is one sixteenth of the video. Frames are resized to a fixed maximum pixel budget before encoding. The same windowing is used for CARDANO, for every baseline and for the whole-clip reference, so the rows of Table 1 differ only in which windows were consumed.

Budget weights. The positive-class weight κ_b of Section 3 takes four values, 0.01, 0.02, 0.05 and 0.10, giving the four points of Figure 5. Every number in the main text is read at $\kappa = 0.05$ and at the head's natural operating point, $\sigma(g_t) > 0.5$; no threshold is selected on any dataset, and the four heads are fit together in one run rather than swept.

The commit head. The head is linear on the nine features of Section 3, with 40 parameters in total. Each feature is standardised with statistics computed once over the training corpus, and the bias is initialised from the base rate of the target so that the head starts at a sensible operating point instead of saturating. The head is fit on the recorded trajectories of the trained Qwen2.5-VL-3B answerer, separately from the answerer, with learning rate 0.02 decayed linearly to zero. We report means over ten fitting seeds.

The answerer. The backbone is adapted with LoRA (Hu et al., 2021) using the objective of Equation 3. During answerer training no gradient from any commit objective reaches the backbone; the deployed head is fit afterwards on the trained answerer's recorded trajectories.

Ties and intervals. Log-probabilities are stored at fixed precision, and this rounding makes about 2% of items exactly tied at the final window. Ties are broken uniformly at random, every reported quantity is averaged over five tie-breaking seeds, and each quantity uses its own random stream, so that a number depends only on the quantity it measures. Intervals are 95% item bootstrap intervals over 10,000 resamples.

A mismatch between training and evaluation. The answerers were trained with a shorter window budget than they are evaluated with: training drew its supervision from a prefix of each clip, while every evaluation in this paper spans the whole clip. The direction of the effect is not obvious, since training on prefixes may help by matching the deployed regime or hurt by never exposing the model to the late windows on which it is scored. We have not run a matched control.