ANNOTATION-GROUNDED EVALUATION OF SELF-GENERATED TEXT EFFECTS IN CLOSED-LOOP STREAMING VIDEO LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video LLMs can retain earlier generated text in the running context, allowing previous outputs to condition later predictions. We study this retained-output dependence as a distinct property of closed-loop streaming generation. Existing evaluations do not isolate it: supplied-history settings omit the model's own generated context, while end-to-end streaming metrics conflate retained-output effects with newly arriving visual evidence. We introduce an annotation-grounded protocol for evaluating retained-output effects during closed-loop generation. In-loop probing reads current-step predictions at annotated times, while narration judging evaluates generated output against the same timestamped reference timeline. Fixed-run reconstructions on COIN show that retained narration can reduce responsiveness to changing visual evidence, particularly across annotated step transitions. We further introduce Retained-Output Contrastive Decoding (ROCD), a training-free method that contrasts prefix-matched views with and without direct access to retained generated text. Across multiple streaming video LLMs, ROCD increases responsiveness to newly arriving visual evidence, with corresponding changes in both probe readouts and generated narration. Together, the protocol characterizes how retained output affects adaptation to changing visual evidence, while ROCD provides a training-free method for reducing this dependence during streaming inference.

1 INTRODUCTION

Streaming video LLMs must update their predictions as new visual content arrives while maintaining information across time. Existing systems manage temporal context through mechanisms such as recent-frame windows, eviction, or compressed visual memory (Xu et al., 2026; Zhang et al., 2025). We focus on systems in which previously generated text remains in the model context and is available when subsequent visual chunks are processed (Xu et al., 2026; Chen et al., 2025; 2024; Yan et al., 2026); we refer to this setting as *closed-loop generation*. Because retained generated text is part of the conditioning context for later predictions, its contribution is intertwined with that of newly arriving visual input and cannot be determined from the generated output alone. We therefore measure how later predictions change when the retained text is controlled while the visual input is held fixed.

Existing evaluations do not isolate this distinction. Benchmarks based on supplied human history or one-time video questions evaluate predictions in externally constructed contexts (Chen et al., 2025; Niu et al., 2025; Fu et al., 2025); they do not include the model's own preceding output as input to later predictions. Closed-loop benchmarks retain that output and score the resulting generation (Yan et al., 2026; Xu et al., 2026), but their end-to-end scores combine visual processing with the influence of retained text. To attribute a change in a later prediction to retained model output, the comparison must be made at the same video time while holding the visual and measurement inputs fixed. This retained-output attribution problem is the focus of our evaluation.

Timestamped step annotations in instructional video identify the active step at sampled times and define a reference timeline across the complete video. At annotated times, *in-loop probing* temporarily asks which step is taking place, reads an answer distribution over dataset-defined options

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.

with a unique correct answer, and restores the preceding context. *Narration judging* evaluates emitted output against the full timeline on temporal misalignment, topical drift, degeneration, coverage, and coherence, using human and time-shifted transcripts as within-video references. The probe measures elicited answers, while the judge measures output produced during closed-loop operation.

On COIN (Tang et al., 2019), retained model narration lowers annotated current-step recognition across the evaluated systems. Around natural step changes, it also weakens the update toward the newly active step. Together, these findings characterize the retained-output effect as reduced responsiveness to changing visual evidence. Retained-Output Contrastive Decoding (ROCD) provides a selective, training-free intervention for this effect. At each decoding step, ROCD compares the model's prediction with a matched view that excludes direct access to retained self-generated text, then uses the contrast to suppress probability increases associated with that access while preserving the immediate speak-or-silence probability. The retained-output-excluded view provides a reference for changes associated with direct access to retained output, allowing ROCD to target that influence while preserving the remaining running context.

Under closed-loop generation across three streaming video LLMs, ROCD increases responsiveness to newly arriving visual evidence, as reflected in annotated-step probe readouts and post-transition updating. Narration judging further shows preferred-direction mean changes in temporal misalignment, topical drift, degeneration, coverage, and coherence. Our contributions are threefold. 1) We formulate retained model output as a distinct source of context in closed-loop streaming video LLMs and introduce an annotation-grounded evaluation protocol. 2) We characterize how retained narration affects current-step predictions, particularly around step transitions. 3) We introduce ROCD, a targeted training-free decoding method for reducing retained-output influence during streaming inference.

2 Related work

2.1 Evaluating supplied contexts and closed-loop outputs

Some video LLM benchmarks supply the text or question used for prediction, while others score output from a running generation loop. LiveSports-3K-CC (Chen et al., 2025) supplies the human commentary preceding each clip, while OVO-Bench (Niu et al., 2025) and Video-MME (Fu et al., 2025) pose questions about the video. Proact-VL's Live Gaming Benchmark-Streaming (Yan et al., 2026) and StreamingVLM's Inf-Streams-Eval (Xu et al., 2026) instead run the generation loop, retain generated text in its running context, and score the resulting output. These scores reflect both the model's response to incoming frames and the influence of its earlier generated text. Attributing later predictions to the retained text therefore requires matched contexts at the same video time.

POPE (Li et al., 2023b) and HallusionBench (Guan et al., 2024) use closed-set questions with one correct answer, while LLM-as-a-judge evaluation uses criterion-wise rubric scoring (Zheng et al., 2023). Closed-set questions provide a uniquely scored answer; rubric scoring evaluates separate properties of generated text.

2.2 Retaining generated text as the video advances

Streaming video LLMs use different mechanisms to manage information accumulated over a stream. StreamingVLM (Xu et al., 2026) retains attention sinks (Xiao et al., 2024), a short window of recent frames, and a longer window of text, while other systems manage historical visual context through retrieval or memory compression (Di et al., 2025; Xiong et al., 2025). In systems that generate continuously, earlier model outputs can also remain available to later predictions: VideoLLM-online (Chen et al., 2024), LiveCC (Chen et al., 2025), Proact-VL (Yan et al., 2026), and FlowNar (Zhong et al., 2026) all retain generated language across streaming updates.

Existing inference-time controls do not directly separate the contribution of retained output from newly arriving visual evidence. Repetition penalties (Keskar et al., 2019) suppress repeated tokens independently of the current frames, while clearing generated text removes useful and erroneous content together. Related work on hallucination propagation has compared predictions with and without generated dialogue history (Zhong et al., 2024), but does not study the same interaction under continuously changing visual input.

2.3 CONTRASTIVE DECODING OVER VISUAL AND TEXTUAL CONTEXT

Contrastive decoding combines next-token distributions obtained under different contexts (Li et al., 2023a). In multimodal settings, VCD (Leng et al., 2024) and M3ID (Favero et al., 2024) alter or remove visual input to expose language-driven predictions, while Self-Aug (Im et al., 2026) adapts the contrastive filtering rule. CAD (Shi et al., 2024) instead contrasts predictions with and without supplied textual context. Previously generated hallucinations in dialogue context have been shown to affect subsequent multimodal predictions, with RVD proposed to strengthen reliance on visual evidence (Zhong et al., 2024). These methods define the contrast through modified visual or textual context. In streaming video, however, the relevant text is generated earlier in the same run and remains available as new frames arrive. Existing end-to-end streaming evaluations report the resulting behavior without separating the contribution of retained generated text from that of the incoming visual evidence.

3 EVALUATION PROTOCOL:IN-LOOP PROBING AND NARRATION JUDGING

Our evaluation protocol combines two measurements of each closed-loop run. *In-loop probing* reads the model's answer distribution at annotated times while the run is in progress. *Narration judging* evaluates the output emitted across the run against the annotated video timeline. The video, annotations, and measurement procedure remain fixed within each paired comparison, while the experimental condition changes. Fixed-run reconstructions measure a context effect conditional on a saved run, whereas closed-loop generation comparisons include its propagation through subsequent retained contexts. The probe and judge show how the corresponding change appears in step answers and generated output, respectively.

3.1 HUMAN ANNOTATIONS AS A COMMON REFERENCE

Several instructional-video benchmarks provide temporally localized procedure steps together with textual step descriptions (Zhou et al., 2018; Tang et al., 2019; Zhukov et al., 2019). At a sampled video time, the boundaries identify the active step and its description. Across the complete video, the ordered annotations define a reference timeline independent of the model's generated text.

The probe uses the step descriptions as answer options and the active step as the correct answer. The judge uses the complete timeline as the reference for evaluating the generated output. The same annotations therefore support a uniquely scored answer at each probe time and a video-level reference for narration judging.

3.2 IN-LOOP PROBING

At an annotated time, the probe temporarily appends a multiple-choice question asking which step is taking place, reads $P(\text{step} \mid C_t)$ from the answer distribution, and restores the pre-query context before generation resumes. Matched reruns find no attributable effect of probing on subsequent narration (Appendix A.4).

The answer options are the video's annotated step descriptions, and the step active at the probed time is the unique correct answer. Probe accuracy is the share of probed times at which the active step receives the highest probability. The probability assigned to that step provides a second readout of the same answer distribution. Dataset-defined options give every evaluated system the same answer space, independent of how it phrases its generated output.

Using model likelihoods to score fixed candidate answers follows established multiple-choice evaluation practice (Lin et al., 2022). Treating the resulting answer probability as a continuous readout also has a precedent in work on model confidence (Steyvers et al., 2025).

3.3 NARRATION JUDGING AGAINST THE ANNOTATED TIMELINE

For each video, the narration judge receives generated outputs together with the annotated step timeline and scores each output on five criteria. *Temporal misalignment* asks whether the output describes the step annotated at each time. *Topical drift* asks whether its content belongs to the video's

instructional task, independent of timing. *Degeneration* scores repetition and looping. *Coverage* asks how much of the annotated timeline the output covers, and *coherence* scores its readability and logical flow. Temporal misalignment, topical drift, and coherence are scored from 0 to 3; degeneration and coverage are scored from 0 to 2. Higher scores for temporal misalignment, topical drift, and degeneration indicate more of the corresponding failure; higher coverage and coherence scores indicate more of the corresponding property. The criteria remain separate rather than forming a single score. LLM-as-a-judge provides a methodological precedent for evaluating open-ended generated outputs (Zheng et al., 2023), while rubric-based NLG evaluation provides a precedent for scoring distinct quality dimensions separately (Liu et al., 2023).

To make the criterion-wise scores interpretable within each comparison set, we include two transcript-derived anchors whose construction is described in Section 3.4. Scores are interpreted only within the same comparison set rather than pooled into a common scale.

The probe produces answer-level accuracy and probability readouts at sampled times, whereas the judge produces five criterion-wise scores for the generated output sequence. We report these results separately, preserving the distinction between answers elicited by the probe and output produced during closed-loop operation.

3.4 Instantiation and anchor construction

We instantiate both measurements on COIN (Tang et al., 2019). Within annotated segments, the probe asks the current-step question at regular intervals. Probe accuracy is computed within each video, and comparisons are paired at the video level. We report paired differences with 95% bootstrap intervals obtained by resampling videos; a difference whose interval contains zero is *within noise*. Further implementation details are provided in Appendix A.

For narration judging, each comparison set contains anonymized model and method outputs together with two anchors derived from the video's speech transcript. The *human-transcript anchor* retains the transcript's original words and timestamps, providing a within-video reference based on human narration. The *time-shifted transcript anchor* uses the same transcript but shifts every timestamp 60 s later. Because the words and ordering are unchanged, the pair provides a controlled manipulation of temporal alignment while preserving transcript content and form. Both anchors are scored with the same timeline, rubric, and judge prompt as the model outputs.

As a manipulation check, the timestamp shift sharply increases temporal misalignment while leaving topical drift, degeneration, coverage, and coherence unchanged within each comparison set. Appendix C reports the criterion-wise anchor scores.

4 Retained generated text affects annotated current-step recognition

4.1 Effect on annotated current-step recognition

At each probe time, we reconstruct paired contexts from the same saved run, holding the video time, visual input, and probe fixed while replacing prior model narration with a placeholder. The placeholder condition yields higher annotated-step accuracy in both evaluated systems (Table 1). These comparisons estimate the effect of retained text conditional on that run; they do not produce the counterfactual closed-loop run induced by a different retention policy.

Retained narration also weakens updates after annotated step changes. Across 105 transitions in 59 videos, the conditions are comparable before onset; while the new step is active, retained narration reduces the increase in its probability by 15.0 percentage points, with a 95% video-level bootstrap interval of $[10.4, 19.7]$ (Appendix B).

4.2 Effects across text amounts and video intervals

Matched amounts of model narration and human transcript produce different accuracy profiles. We reconstruct contexts with increasing amounts of text from video intervals preceding the same probe time. The *model-narration condition* uses the model's saved output, whereas the *human-transcript*

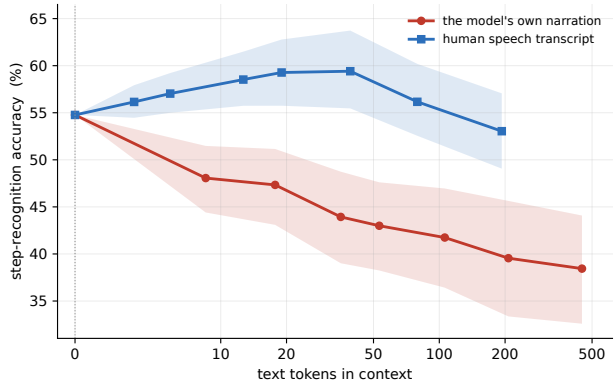


Figure 1: **Accuracy by retained-text amount.** Annotated-step accuracy for StreamingVLM on COIN ($n = 40$); shading shows paired 95% video-level bootstrap intervals relative to the placeholder.

Table 1: **Effect of retained model narration on annotated-step recognition.** Differences are placeholder minus retained-text accuracy; brackets are paired 95% video-level bootstrap intervals.

Retained text	System	Videos	Difference (pp)
Full prior narration	StreamingVLM	80	16.3 [11.8, 20.8]
Full prior narration	LiveCC	80	11.0 [6.1, 15.9]
Excluding the latest 12 s	StreamingVLM	32	18.5 [8.7, 28.5]
Excluding the latest 24 s	StreamingVLM	32	16.8 [8.2, 25.4]
Latest 12 s only	StreamingVLM	32	8.9 [2.8, 15.2]

condition uses the video's speech transcript. At each tested amount, the two conditions are matched in token count, while their source and content remain coupled. Even the smallest tested amount of model narration yields lower accuracy than the placeholder condition, while the human-transcript condition remains near or above it over most of the tested range (Figure 1).

Within each saved narration run, we vary which portion of the preceding video timeline is retained while holding the current probe time fixed. Retaining either recent narration or narration from earlier video intervals yields lower accuracy than the placeholder condition (Table 1). Generated text from multiple portions of the video therefore contributes to the measured effect.

5 Method: Retained-Output Contrastive Decoding

At video time t, a streaming video LLM predicts from its running stream state, which may include self-generated text retained from earlier video times. Retained-Output Contrastive Decoding (ROCD) contrasts the closed-loop view with a prefix-matched retained-output-excluded view, suppresses token-probability increases associated with direct access to retained output, and preserves the immediate speak-or-silence probability mass.

5.1 Prefix-matched views with and without direct access to retained output

Let S_t denote the running stream state immediately before the current output, let $R_{<t}$ denote self-generated text retained from earlier video times, and let $u_{<k}$ denote the prefix currently being generated. The closed-loop view induces

$$p_{\mathrm{CL}}(y) = p_\theta(y \mid S_t, u_{<k}), \tag{1}$$

where the current prediction can directly access the entries in S_t assigned to $R_{<t}$. Let $\Pi_{\neg R}$ retain the addressable entries not assigned to $R_{<t}$. The prefix-matched retained-output-excluded view induces

$$p_{\mathrm{ROE}}(y) = p_\theta(y \mid \Pi_{\neg R}(S_t), u_{<k}). \tag{2}$$

Both views use the same model weights, current output prefix, and remaining running state. Their contrast therefore measures changes associated with direct access to $R_{<t}$ at the same autoregressive position. Because the retained-output-excluded view is formed from the same running state, it retains the cached visual and other non-output entries. This construction provides a matched reference for measuring how direct access to retained output changes the next-token distribution.

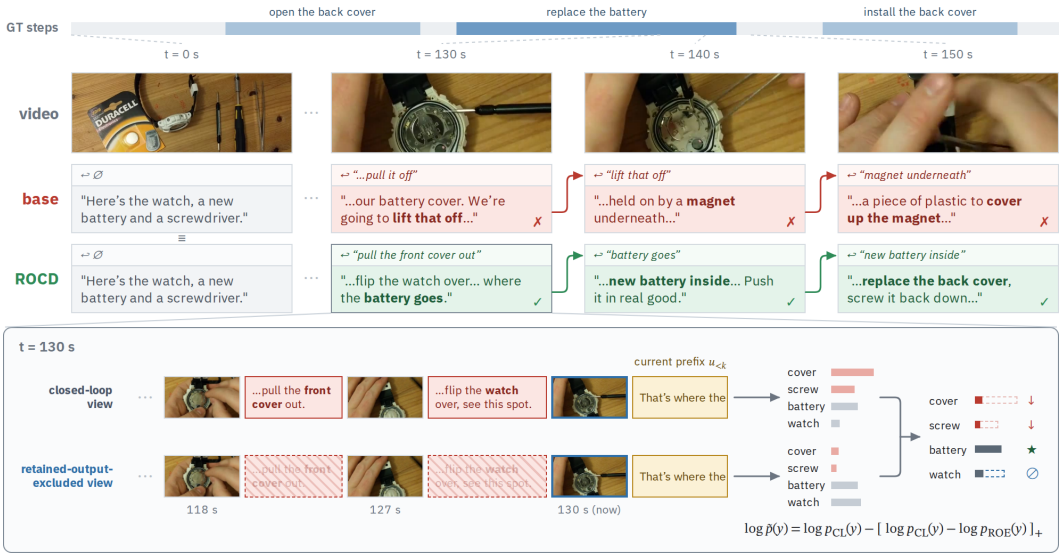


Figure 2: **ROCD overview.** Top: base continues prior cover-related narration, whereas ROCD follows the annotated battery-replacement step; arrows show outputs retained for later predictions. Bottom: the prefix-matched views share the visual state and current prefix $u_{<k}$, while red hatched blocks mark retained output excluded from direct access.

5.2 Suppressing increases associated with direct access to retained output

ROCD assigns each candidate token the unnormalized weight

$$w(y) = \min(p_{\mathrm{CL}}(y), p_{\mathrm{ROE}}(y)) \,. \tag{3}$$

Equivalently,

$$\log w(y) = \log p_{\mathrm{CL}}(y) - [\log p_{\mathrm{CL}}(y) - \log p_{\mathrm{ROE}}(y)]_+ \,. \tag{4}$$

If direct access to $R_{<t}$ raises a token's probability, ROCD caps its weight at its probability under the retained-output-excluded view; otherwise it retains the closed-loop probability. The contrast determines the suppression pointwise, without a strength coefficient. The retained-output-excluded distribution therefore supplies an upper bound rather than replacing the closed-loop distribution. After normalization, tokens whose weights were not reduced can gain relative probability.

5.3 Preserving the speak-or-silence decision

Applying Equation 3 over the full output vocabulary would couple what the model says with whether it speaks: probability removed from ordinary tokens could be reassigned to silence or termination tokens. ROCD instead fixes the closed-loop speak-or-silence probability and reweights only the conditional distribution over non-silence tokens.

Let $\mathcal{Y}_{\mathrm{sil}}$ denote the model's silence and termination tokens, and let $\mathcal{Y}_{\mathrm{sil}}^c$ denote their complement in the output vocabulary. Define

$$Z = \sum_{y' \in \mathcal{Y}_{\mathrm{sil}}^c} w(y'). \tag{5}$$

The ROCD distribution is

$$\tilde{p}(y) = \begin{cases} p_{\mathrm{CL}}(\mathcal{Y}_{\mathrm{sil}})\, p_{\mathrm{CL}}(y \mid \mathcal{Y}_{\mathrm{sil}}), & y \in \mathcal{Y}_{\mathrm{sil}}, \\ p_{\mathrm{CL}}(\mathcal{Y}_{\mathrm{sil}}^c)\, \dfrac{w(y)}{Z}, & y \in \mathcal{Y}_{\mathrm{sil}}^c. \end{cases} \tag{6}$$

Equation 6 preserves the silence mass and its internal distribution at the current decoding state while redistributing probability only within $\mathcal{Y}_{\mathrm{sil}}^c$. This local constraint does not fix the emission rate over the resulting closed-loop run.

Table 2: **In-loop probe results (%).** Δ is relative to base; † marks a paired 95% video-level bootstrap interval excluding zero. Bold marks each system's best mean.

System	Method	Top-1 ↑	Δ	$P(\text{step})$ ↑	Δ
StreamingVLM	Base	40.5	–	36.4	–
	VCD	43.5	$+2.9$	39.6	$+3.3^{\dagger}$
	ROCD (ours)	**50.7**	$\mathbf{+10.2}^{\dagger}$	**43.0**	$\mathbf{+6.6}^{\dagger}$
LiveCC	Base	38.3	–	36.8	–
	VCD	40.8	$+2.5$	39.2	$+2.4^{\dagger}$
	ROCD (ours)	**46.4**	$\mathbf{+8.1}^{\dagger}$	**42.0**	$\mathbf{+5.2}^{\dagger}$
Proact-VL	Base	49.6	–	43.4	–
	VCD	50.5	$+0.9$	47.9	$+4.5^{\dagger}$
	ROCD (ours)	**54.6**	$\mathbf{+5.0}^{\dagger}$	**49.1**	$\mathbf{+5.7}^{\dagger}$

5.4 Constructing the retained-output-excluded view during streaming inference

Streaming systems represent S_t with a running key–value cache. We associate each cache position with its provenance—prompt, visual input, or self-generated text—and implement $\Pi_{\neg R}$ by retaining prompt and visual positions while omitting positions assigned to $R_{<t}$. The current prefix $u_{<k}$ is appended to both views. We call this operation *cache selection*; it constructs ROCD's retained-output-excluded view directly from the running state.

Cache selection reuses encoded visual inputs and accumulated non-output entries, avoiding stream replay, visual re-encoding, and a new context prefill at each video update. ROCD still evaluates an additional retained-output-excluded branch and retains its key–value cache; the two views share model weights but not cache state. Its computational benefit is therefore avoidance of context reconstruction, not elimination of contrastive inference or key–value memory.

6 Experiments

We evaluate ROCD on COIN under closed-loop generation using the released checkpoints and configurations of StreamingVLM (Xu et al., 2026), LiveCC (Chen et al., 2025), and Proact-VL (Yan et al., 2026), without updating model weights. In-loop probing uses a fixed 80-video evaluation set, while narration judging uses 101 videos per system.

For probing, we compare ROCD against the original deployment configuration of each model (*base*) and a streaming adaptation of VCD, whose contrastive branch preserves retained text while applying diffusion noise to the visual input. ROCD and VCD are applied during both generation and probe readout, with all comparisons paired by video and probe state. Narration judging additionally includes *text clearing*, which removes completed sentences while retaining an unfinished trailing sentence. We report top-1 annotated-step accuracy as the primary probe metric and annotated-step probability as a secondary readout, with paired 95% video-level bootstrap intervals. Judge criteria are reported separately as within-system descriptive means; cross-system comparisons are limited to the direction of within-system differences. Implementation and evaluation details are provided in Appendices A and A.1.

6.1 ROCD improves recognition and strengthens updates after step changes

Across all three systems, ROCD improves both the Top-1 decision and the probability assigned to the annotated step (Table 2); all six ROCD–base intervals exclude zero (Appendix A.3).VCD changes the answer distribution but yields smaller observed Top-1 gains than ROCD. On StreamingVLM, ROCD adds 7.2 points over VCD in the paired Top-1 comparison, with an interval that excludes zero (Appendix A.3). Thus the cross-system decision gain is not reproduced by an input-side visual contrast. Step-aligned results further locate the gain: the pre-onset difference is within noise, whereas after onset ROCD increases the update toward the newly active step by 6.5–10.2 percentage points across the three systems, with all intervals excluding zero (Figure 3). This temporal asymme-

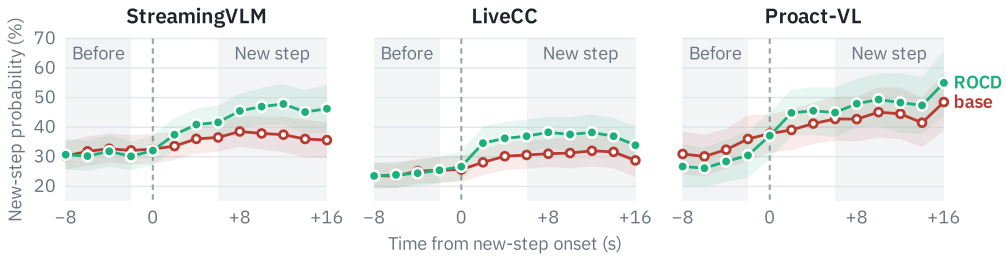


Figure 3: **ROCD strengthens updates after step transitions.** Curves show mean new-step probability at two-second offsets; bands are 95% video-level bootstrap intervals. Gray regions denote the analysis windows.

Table 3: **Narration judging against the annotated timeline.** Anchors are cross-set ranges; methods are within-system means. Bold text indicates the best generated output for each system, while the second-best is underlined.

System	Output	Misalign. ↓	Drift ↓	Degen. ↓	Cover. ↑	Coher. ↑
Anchors	Human transcript	0.19–0.31	0.04–0.06	0.00–0.01	1.91–1.92	2.83
	Shifted transcript	2.97–2.98	0.04–0.06	0.00–0.01	1.91–1.92	2.83
StreamingVLM	Base	1.86	1.67	0.11	1.43	1.90
	Text clearing	**1.30**	**1.17**	1.06	**1.72**	0.80
	VCD	1.80	1.62	0.24	1.50	**1.96**
	ROCD (ours)	1.40	1.26	**0.03**	**1.72**	1.94
LiveCC	Base	2.26	2.01	1.43	1.39	1.46
	Text clearing	2.00	1.81	1.63	1.50	1.23
	VCD	1.50	1.46	0.96	1.72	**1.66**
	ROCD (ours)	**1.14**	**1.11**	**0.67**	**1.79**	1.62
Proact-VL	Base	2.16	1.58	1.52	1.12	1.34
	Text clearing	2.42	**1.04**	1.16	0.38	0.19
	VCD	1.81	1.45	1.44	1.41	1.38
	ROCD (ours)	**1.37**	1.07	**0.82**	**1.65**	**1.59**

try mirrors Section 4: retained narration dampens adaptation when the visual state changes, whereas ROCD restores responsiveness to new evidence rather than imposing a resolved global confidence offset. Appendix A.2 gives transition eligibility rules and sample counts.

6.2 Probe gains coincide with preferred narration changes

We use the judge results to examine whether the probe gains come from making the narration more conservative at the cost of informativeness. We do not observe such a tradeoff. Compared with base, ROCD reduces temporal misalignment, topical drift, and degeneration, while improving coverage and coherence across all three systems (Table 3). The gains in step recognition therefore do not come with lower coverage or coherence in the generated narration.

The baselines show less consistent behavior. Text clearing improves temporal misalignment or topical drift in some systems, but can also increase degeneration or reduce coverage and coherence, while VCD avoids the most severe degradations. ROCD is the only intervention whose mean score moves in the preferred direction on all five criteria for all three systems. We interpret these rubric scores as evidence of directional consistency under the fixed judge configuration, rather than as resolved pairwise superiority.

6.3 Attribution and implementation checks

First, we test whether the probe gain depends on directly changing the immediate speak-or-silence decision. Unconstrained ROCD applies Equation 3 over the full vocabulary, allowing the contrast to change both emitted content and silence mass. The control isolates ROCD's intervention at a fixed decoding state. ROCD changes no measured silence mass, yet its recognition gain is comparable to that of unconstrained ROCD, which changes the silence mass at 54.0% of states. The gain therefore

Table 4: **Speak-or-silence control on a 40-video subset of the 80-video probe set (%).** Changed states have silence-mass changes above 10^{-3}.

Variant	Changed states ↓	Top-1 ↑	$P(\text{step})$ ↑
Base	–	38.1	34.1
Unconstrained ROCD	54.0	48.2	40.7
ROCD (ours)	0.0	50.2	39.7

does not require directly shifting the immediate speak-or-silence decision; it can arise from redistributing probability within the non-silence tokens. Changed content can still alter later states and the emission rate over the resulting closed-loop run (Appendix A.5).

We next compare cache selection with an independently reconstructed reference without retained output. This reference tests a different context construction, not an exact version of ROCD. The two constructions select the same reweighted token at 82.3% of intervention states, showing that they are related but not interchangeable contrasts. Their recognition difference is unresolved in the 12-video, three-seed comparison; the current evidence therefore does not favor either construction for recognition. Appendix A.6 reports the distributional and downstream generation results.

Finally, cache selection makes the contrast practical by avoiding reconstruction of the visual and textual input pipeline. It reduces branch preparation from 1,238 to 39 ms per video chunk and keeps ROCD below real time (RTF 0.434), whereas the reconstructed reference and VCD both exceed an RTF of 1. This is a latency benefit rather than a memory optimization: ROCD still retains an additional key–value cache. Appendix A.7 reports the complete measurements.

7 Limitations

The probe measures recognition within a dataset-defined answer space and does not establish open-vocabulary understanding outside that annotation ontology. Narration-judge scores are likewise relative to the specified criteria and comparison set rather than an absolute scale transferable across datasets or judge configurations. We interpret their means descriptively under the fixed judge configuration: video-level resampling would quantify variation across cases but would not capture uncertainty from the judge, prompt, or rubric. Streams with overlapping events, weak temporal boundaries, or no suitable annotations require different probe and judging constructions.

The fixed-run reconstructions in Section 4 and closed-loop generation experiments in Section 6 estimate different effects: the former condition on a saved run, whereas the latter include propagation through changed future contexts. Their effect magnitudes are therefore not directly comparable.

ROCD reweights according to changes associated with direct access to retained self-generated text rather than according to correctness, so it can suppress useful as well as erroneous influences. It also cannot introduce a correct description that the closed-loop distribution does not already propose. Finally, excluding direct access to self-generated text does not remove indirect dependence preserved in the running state, and the current comparison with independent reconstruction cannot separate this dependence from differences between the input pipelines.

8 Conclusion

We introduced in-loop probing and narration judging to measure how retained self-generated text affects current-step recognition and emitted narration in closed-loop streaming video LLMs. Fixed-run reconstructions show that retained narration weakens recognition, including adaptation after step transitions, and that matched amounts of model narration and human transcript yield different accuracy profiles. ROCD suppresses token-probability increases associated with direct access to retained output while preserving the immediate speak-or-silence probability. Across three systems, ROCD improves both probe readouts and post-transition updates; narration judging shows preferred-direction changes on all five criteria. The protocol and ROCD make self-generated context measurable and actionable during streaming inference.

AI USE STATEMENT

We used generative AI tools for language editing and polishing, retrieval and discovery of related work, research ideation and execution, and drafting portions of the manuscript. All AI-assisted text, technical content, and references were reviewed and verified by the authors. We did not use generative AI tools to generate synthetic datasets or to prove mathematical claims. We take full responsibility for the final content of this work.

REFERENCES

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

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

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *International Conference on Learning Representations (ICLR)*, 2025.

Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14303–14312, June 2024.

Chaoyou Fu, Yuhan Dai, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2405.21075.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Eun Woo Im, Muhammad Kashif Ali, and Vivek Gupta. Self-aug: Query and entropy adaptive decoding for large vision-language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023b.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023.

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

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.naacl-short.69/.

Mark Steyvers, Heliodoro Tejeda, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas W. Mayer, and Padhraic Smyth. What large language models know and what people think they know. *Nature Machine Intelligence*, 7:221–231, 2025.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.17453.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. In *International Conference on Learning Representations (ICLR)*, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61463–61475, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6445dd88ebb9a6a3afa0b126ad87fe41-Paper-Conference.pdf.

Weicai Yan, Yuhong Dai, Qi Ran, Haodong Li, Wang Lin, Tao Jin, Xing Xie, Hao Liao, and Jianxun Lian. Proact-VL: A proactive videoLLM for real-time AI companions. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=k9PKgV0L4C.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Weihong Zhong, Xiaocheng Feng, Liang Zhao, Qiming Li, Lei Huang, Yuxuan Gu, Weitao Ma, Yuan Xu, and Bing Qin. Investigating and mitigating the multimodal hallucination snowballing in large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.648/.

Zeyun Zhong, Manuel Martin, Chengzhi Wu, David Schneider, Frederik Diederichs, Juergen Gall, and Juergen Beyerer. FlowNar: Scalable streaming narration for long-form videos. In *International Conference on Machine Learning*. PMLR, 2026. Accepted, to appear.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7590–7598, 2018.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

A MEASUREMENT IMPLEMENTATION DETAILS

At each probing time, we obtain $P(\text{step} \mid C_t)$ from the next-token probabilities assigned to the option letters. We then trim the cache to its pre-question state so that the probe does not alter the subsequent generated output.

Narration-judging scores are compared only within the same comparison set. Prompt versions and judge models are kept separate rather than pooled into a common scale. We report criterion-wise means as descriptive summaries under each fixed judge configuration; they are not used for resolved pairwise method claims.

Across the 80 probe videos, StreamingVLM and LiveCC each provide 3,459 two-second probe times within annotated steps, and Proact-VL provides 3,568. Within each paired comparison, the video, visual input, probe time, question, option order, and model weights are identical. Results are aggregated within videos, with 95% intervals from 10,000 video-level bootstrap samples.

A.1 BASELINE IMPLEMENTATIONS

Both baselines use the same released checkpoints, prompts, and decoding configurations as the unmodified closed loop, and neither updates model weights.

Text clearing. Text clearing removes completed generated text from the running context after each output while keeping the visual inputs, the task prompt, and the user inputs. Generated text is divided into sentences at periods, exclamation marks, and question marks, excluding ellipses, decimal points, and periods inside tokens such as URLs. After each output, the key–value entries of every completed sentence are removed; an unfinished trailing sentence is kept so that the model can complete it at the next output. Only content tokens are removed: assistant-turn delimiters remain, and no silence or placeholder text replaces the removed sentences. Each system uses its own cache mechanics. On StreamingVLM, generated text enters the model's native previous-text block at every chunk, and completed sentences are pruned from that block with the pruning operation of its sliding window. On LiveCC, removed entries are cut from the middle of the cache and the rotary position offset is advanced by the number of removed tokens, so later tokens continue the original position numbering. On Proact-VL, the model's position bookkeeping is updated without decreasing its maximum position index.

VCD. VCD (Leng et al., 2024) contrasts the closed-loop prediction with a branch whose visual input is distorted and whose text is unchanged. With ℓ_{CL} and ℓ_{noise} denoting the next-token logits of the two branches,

$$\ell_{\mathrm{VCD}}(y) = (1+\alpha)\,\ell_{\mathrm{CL}}(y) - \alpha\,\ell_{\mathrm{noise}}(y), \tag{7}$$

evaluated only for candidates with $p_{\mathrm{CL}}(y) \geq \beta \max_{y'} p_{\mathrm{CL}}(y')$; all other candidates receive zero probability. We use the original values $\alpha = 1$ and $\beta = 0.1$. The distorted input applies forward diffusion at step 500 of a 1,000-step schedule with noise variances increasing linearly from 10^{-4} to 0.02, added to the preprocessed pixel tensors as in the original implementation. The noised branch receives the same retained generated text and the same current prefix as the closed loop, so the visual input is the only manipulated variable. On StreamingVLM, the noised branch is rebuilt at each video chunk from the current visual window and the generated text so far, and is then advanced with the tokens emitted by the closed loop. On LiveCC and Proact-VL, it is a persistent stream with its own key–value cache that receives the same inputs with noised pixels, together with the emitted text, at every chunk. As in the original method, VCD reweights the full output vocabulary; it does not preserve the speak-or-silence probability, which is a component of ROCD (Section 5.3). At probe

times, the same rule combines the option-letter distributions of the closed-loop and noised contexts for the same question.

A.2 ROCD TRANSITION ANALYSIS

The before window averages the four two-second offsets from -8 to -2 s. The new-step window averages the available offsets from $+6$ to $+16$ s for which the newly annotated step remains active, with at least two such offsets required per transition. This yields 105 transitions across 59 videos for StreamingVLM and LiveCC and 71 transitions across 45 videos for Proact-VL.

A.3 PAIRED INTERVALS FOR PROBE RESULTS

Unless the contrast names two methods, differences are ROCD minus base.

Table 5: **Paired intervals for the probe results.** Differences and intervals are in percentage points.

Analysis	System and readout	Difference	95% interval
Overall	StreamingVLM, Top-1	$+10.2$	$[6.2, 14.2]$
	StreamingVLM, annotated-step P	$+6.6$	$[4.6, 8.7]$
	LiveCC, Top-1	$+8.1$	$[3.6, 12.6]$
	LiveCC, annotated-step P	$+5.2$	$[3.2, 7.1]$
	Proact-VL, Top-1	$+5.0$	$[1.6, 8.5]$
	Proact-VL, annotated-step P	$+5.7$	$[3.5, 7.9]$
Step transition	StreamingVLM, increase	$+10.2$	$[6.1, 14.2]$
	StreamingVLM, new-step level	$+9.2$	$[5.4, 12.9]$
	LiveCC, increase	$+6.5$	$[3.7, 9.3]$
	LiveCC, new-step level	$+6.4$	$[3.1, 9.5]$
	Proact-VL, increase	$+8.6$	$[4.2, 13.3]$
	Proact-VL, new-step level	$+4.2$	$[-3.6, 11.0]$
Direct comparison	StreamingVLM, ROCD–VCD Top-1	$+7.2$	$[3.1, 11.4]$
	StreamingVLM, ROCD–VCD annotated-step P	$+3.4$	$[1.2, 5.6]$

A.4 PROBE NON-INTERFERENCE CHECK

We run the same 80 StreamingVLM videos with and without in-loop probing under the same decoding configuration and random seed. Base narration is character-identical in all 80 paired runs. ROCD narration is identical in 79 of 80 direct comparisons. In the remaining video, the stored probed run first differs from the unprobed run at 103 s; repeated current runs with and without probes are character-identical to one another and diverge together from the stored run at that time. Instrumented probe calls preserve the random-number-generator state and cached keys and values bit-for-bit, although position-index metadata is not bitwise identical after trimming. The matched current reruns attribute the recorded output mismatch to run reproducibility rather than the probe intervention.

A.5 EMISSION RATES DURING CLOSED-LOOP GENERATION

Across 42,139 reweighted decoding states, ROCD preserves $p_{\mathrm{CL}}(\mathcal{Y}_{\mathrm{sil}})$ with a maximum silence-mass change of 6×10^{-7}. Unconstrained ROCD changes the silence mass by more than 10^{-3} at 54.0% of states, always increasing it.

Emission rate is the percentage of streaming steps at which the system emits text. On the 40-video StreamingVLM subset, the observed rates are 84.3% for base, 64.3% for unconstrained ROCD, and 79.4% for ROCD. Because changed tokens alter later closed-loop states, these run-level rates do not test the immediate invariance in Table 4.

A.6 COMPARISON WITH INDEPENDENT RECONSTRUCTION

Let I_t contain the task prompt, timestamps, and visual inputs specified at video time t. We independently reconstruct a context without retained self-generated text and define the reconstructed

reference

$$p_{\mathrm{recon}}(y) = p_\theta(y \mid I_t, u_{<k}). \tag{8}$$

This reference rebuilds its input state, whereas p_{ROE} excludes direct access to $R_{<t}$ within the retained running state. The two distributions therefore arise from different context constructions; neither is defined as an approximation of the other.

We compare p_{ROE} with p_{recon} at matched states during closed-loop generation with ROCD. The analysis contains 16 videos, 16,241 decoding states, and 12,570 states at which the reweighting rule changes at least one token weight. Distribution-level measures precede reweighting; decision-level measures follow Equation 6. Results are first aggregated within each video.

Table 6: **Retained-output-excluded and independently reconstructed contrasts.** Intervals are 95% video-level bootstrap intervals. CL-to-contrast JS reports $p_{\mathrm{recon}}/p_{\mathrm{ROE}}$ in that order.

Measure	All states	Intervention states
JS$(p_{\mathrm{recon}}, p_{\mathrm{ROE}})$	$0.066\ [0.058, 0.073]$	$0.078\ [0.069, 0.087]$
Contrast top-1 agreement	0.748	0.698
Reweighted-distribution JS	0.028	0.036
Reweighted-token agreement	$0.864\ [0.847, 0.881]$	$0.823\ [0.803, 0.844]$
CL-to-contrast JS	$0.125/0.082$	$0.140/0.086$

The prespecified agreement analysis required reweighted-token agreement of at least 0.95 and a median relative-JS ratio no greater than 0.25. At intervention states, agreement is 0.823 and the per-video median ratio is 0.38 (0.39 when states are pooled), so neither threshold is met. The retained-output-excluded distribution is also closer to the closed-loop distribution than the reconstructed reference, but this difference cannot be assigned solely to indirect dependence on $R_{<t}$: a discrepancy is already present early in the videos and at short accumulated contexts, and the constructions use different visual-input pipelines.

Closed-loop generation comparison. We separately compare downstream outcomes on 12 videos and average each video over decoding seeds 7, 11, and 23.

Table 7: **Closed-loop generation comparison with the reconstructed reference.** Differences are in percentage points and intervals are paired 95% video-level bootstrap intervals.

Contrast	Top-1	$P(\mathrm{step})$	Emission rate
ROCD − base	$+10.3\ [6.4, 15.1]$	$+7.2\ [4.9, 9.8]$	-4.7 (within noise)
Reconstructed reference − base	$+11.7\ [6.3, 17.2]$	$+7.7\ [4.1, 11.7]$	$+4.4$ (within noise)
Reconstructed reference − ROCD	$+1.4\ [-3.9, 6.1]$	$+0.5\ [-2.5, 3.2]$	$+9.1\ [4.4, 14.3]$

The recognition difference between the two context constructions is unresolved, whereas their emission rates differ. Both preserve $p_{\mathrm{CL}}(\mathcal{Y}_{\mathrm{sil}})$ at each reweighted state, so the emission-rate difference arises after their closed-loop runs diverge; the present experiment does not isolate its mechanism.

A.7 Computational cost

Timing measurements use six videos with three measured repetitions per method on an otherwise unoccupied GPU. RTF is elapsed time divided by video duration; relative time is paired to base. Peak memory is the increase over memory allocated before each run.

The retained-output-excluded branch and reconstructed reference retain similar contrastive caches (approximately 0.28 GB each), so cache selection does not provide a key–value memory saving. It avoids a second model instance but still incurs an additional token-level forward pass.

As a post-hoc analysis that excludes the first 16 seconds, during which the visual window is still growing, the per-video preparation-time slope for ROCD is -0.9 ms per 1,000 accumulated tokens, with a 95% interval of $[-3.9, 0.9]$. The corresponding slopes are $+29.2\ [20.3, 38.1]$ for the reconstructed reference and $+71.2\ [63.4, 79.0]$ for VCD. Because this window was selected after

Table 8: **Computational cost.** ROCD prepares its retained-output-excluded branch by cache selection; the reconstructed reference independently rebuilds a context without retained output. Preparation is the corresponding per-chunk work before token decoding.

Method	RTF ↓	Relative time ↓	Preparation (ms) ↓	Peak increase (GB) ↓
Base	0.261	1.00×	–	0.53
ROCD	0.434	1.66×	39	1.20
Reconstructed reference	1.581	6.05×	1,238	2.56
VCD	1.594	6.11×	1,272	2.86

inspecting the full-video slopes, these values characterize the observed scaling behavior rather than a prespecified confirmatory test.

B Phenomenon details

B.1 Annotated step transitions

We align the probe measurements to natural COIN step boundaries in the same 80-video set used for the main probe results. The baseline range covers eight to two seconds before each boundary. The response range covers probe times from six to sixteen seconds after the boundary while the newly annotated step remains active, with at least two such probe times per transition. This yields 105 transitions across 59 videos. Before the boundary, the probability assigned to the newly active step is 30.1% without narration and 31.7% with the model's narration retained; their paired difference is within noise. During the new step, the corresponding probabilities are 50.6% and 37.3%. The increase is 20.6 percentage points without narration and 5.5 percentage points with narration retained, giving a paired difference in the increase of 15.0 percentage points with a 95% video-level bootstrap interval of $[10.4, 19.7]$.

As a sensitivity analysis, we include every probe time from six to sixteen seconds after each boundary, including times after short annotated steps have ended. This fixed-window analysis contains 151 transitions across 68 videos. The probability assigned to the new step changes from 29.6% to 50.0% without narration and from 31.7% to 37.9% with narration retained. The paired difference between these increases is 14.2 percentage points, with a 95% video-level bootstrap interval of $[9.8, 18.8]$.

B.2 Text amount and retention analyses

The dose analysis probes 40 videos at four-second intervals within annotated segments. At each probe time, the compared text conditions share the same reconstructed visual and probe inputs.

At equal amounts of narration, windows ending twelve or twenty-four seconds before the probe and windows ending at the probed time all lower accuracy relative to the no-narration condition. The paired differences between recent and earlier windows are within noise at the tested amounts. Shuffling the order of the saved narration likewise leaves its difference from the unshuffled condition within noise. These comparisons do not establish equivalence; they report differences that the current experiments do not resolve. Deploying a different retention policy would produce a different closed-loop run; the reported comparisons instead reconstruct contexts from the saved run.

B.3 Running-loop rewrite

The running-loop study crosses two factors: which sentence the context holds and which independently generated continuation supplies the later narration. At one moment, we replace three seconds of generated narration with a template sentence that names either the correct step or the wrong step the model had named. We then rerun both rewrites while forcing their later narration to match the continuation generated after either sentence. Holding the later narration fixed identifies the contribution of the sentence retained in context; holding the sentence fixed identifies the contribution transmitted through later narration. The probe reads the running cache by appending the question, reading the probability of the correct step, and trimming the cache back to its previous state.

From four to sixteen seconds after the rewrite, the correct sentence raises the probability of the correct step relative to the wrong sentence, and most of the measured difference is associated with the sentence retained in context. The contribution transmitted through later narration is within noise in this small study. The second prespecified range, twenty-four to sixty-four seconds after the rewrite, has complete readings for too few videos to estimate the effect. The study measures step recognition on StreamingVLM, and applying its result to naturally generated errors assumes that later narration responds to the model's own wrong sentences as it responds to the wrong-step template sentence.

C NARRATION JUDGE: PROMPT CALIBRATION AND MANIPULATION CHECK

Narration judging is performed with Claude Opus 5. For each COIN video, we first extract the spoken narration using `openai/whisper-base.en`. Videos are included in narration judging only when the resulting ASR transcript contains at least 100 words, ensuring that the speech-transcript anchor contains sufficient narrated content for comparison.

For each included video, the judge receives the task category, the ground-truth step timeline, and six anonymized transcripts corresponding to the same video. The transcripts include the evaluated system outputs together with the ASR-derived speech-transcript anchor and its time-shifted counterpart. Transcript lines and timeline segments use the same video clock. System identities are hidden, and the judge is instructed not to infer transcript provenance and to score each transcript independently.

The judge assigns five scores to every transcript: temporal alignment, topical drift, degeneration, coverage, and coherence. Temporal alignment is scored from 0 to 3 against the annotated step timeline, with lower values indicating closer correspondence between the narrated content and the step active at the same timestamp. Drift is scored from 0 to 3 and measures whether the narrated content remains within the instructional task irrespective of timing. Degeneration is scored from 0 to 2 for repetition and looping, coverage from 0 to 2 for how much of the annotated task is narrated, and coherence from 0 to 3 for readability and logical flow. The dimensions are scored independently.

To calibrate the temporal-alignment scale, the prompt includes three fixed hand-written anchors: an on-task and on-time example with alignment 0, an on-task narration shifted by approximately one step with sustained temporal mismatch and alignment 3, and an off-task narration with alignment 3 and drift 3. The complete judge prompt is reproduced below. Shifting the human transcript by 60 seconds raises temporal misalignment from 0.22 to 2.97 on the StreamingVLM comparison set, from 0.19 to 2.98 on the LiveCC set, and from 0.31 to 2.98 on the Proact-VL set. Because the shift preserves the transcript words, its scores for topical drift, degeneration, coverage, and coherence remain unchanged within each set. In this manipulation, timestamp displacement therefore changes only the temporal-misalignment score.

Table 9: **Transcript anchors for narration judging.** Arrows indicate the preferred direction.

System	Anchor	Misalign. ↓	Drift ↓	Degen. ↓	Cover. ↑	Coher. ↑
StreamingVLM	Human transcript	0.22	0.06	0.01	1.92	2.83
	Shifted transcript	2.97	0.06	0.01	1.92	2.83
LiveCC	Human transcript	0.19	0.04	0.00	1.92	2.83
	Shifted transcript	2.98	0.04	0.00	1.92	2.83
Proact-VL	Human transcript	0.31	0.05	0.00	1.91	2.83
	Shifted transcript	2.98	0.05	0.00	1.91	2.83

C.1 NARRATION JUDGE PROMPT

For each video, the narration judge receives the task category, the ground-truth step timeline, and six anonymized narration transcripts. Transcript lines and timeline segments share the same video clock. The judge scores every transcript independently on five criteria: temporal misalignment, topical drift, degeneration, coverage, and coherence. The prompt explicitly separates temporal alignment from topical relevance so that an on-task narration spoken at the wrong time can receive high misalignment but low drift. Three fixed calibration examples anchor the temporal-misalignment

scale, including an on-time narration, a consistently time-shifted but otherwise on-task narration, and an off-task narration. The complete judge prompt is reproduced below.

You are rating transcripts of real-time video narration. For each video you are given the task category (`video_class`), a **ground-truth step timeline** (`timeline`) listing what the person is actually doing during each time span, and six unlabeled transcripts (A–F) of narration for that same video. The transcripts come from different sources; do not attempt to identify or guess the sources, and rate each transcript independently on its own merits.

Each transcript line is prefixed with a timestamp [Ns] indicating when it was spoken. The timeline uses the same clock.

Rate EVERY transcript on ALL five dimensions:

alignment (0–3, lower is better) — primary dimension

At each moment, is the narration describing the step that the timeline says is actually happening at that moment?

Match by MEANING, not wording: a narration may paraphrase a step, describe it in finer detail, or name the tools/objects involved—all of these count as describing that step. Explanations, tips and asides that accompany the current step count as describing it.

A transcript can be fully on-task yet misaligned (it describes real steps of this task, but at the wrong times)—that is alignment 2–3 with drift 0.

- 0 = well aligned: throughout the video, what is narrated at time t corresponds to the step the timeline gives for time t.
- 1 = mostly aligned: brief lags or anticipations only; the narration generally moves through the steps in step with the timeline.
- 2 = misaligned in places, but the narration still tracks the timeline's progression with only a modest offset.
- 3 = a SUSTAINED, whole-video temporal mismatch: the narration marches through real steps but on its own clock, so at essentially every moment it is describing a different step than the one happening. A large, CONSISTENT lag or lead across the whole video counts as 3, because the narration never actually coincides with the current step even though the steps themselves are real. A short or occasional lag is 1–2; a whole-video offset is 3.

Judge alignment ONLY. Conversely a transcript can be aligned in its opening and then depart from the task entirely—that is bad on both alignment and drift.

drift (0–3, lower is better)

How far the narration departs from what a video of this task would plausibly show, regardless of timing.

- 0 = fully anchored: content stays within this task.
- 1 = mostly anchored: occasional generic filler or brief off-task remarks.
- 2 = substantially detached: extended stretches describe things a video of this task would not show, or the narration follows its own storyline.
- 3 = largely detached: driven by its own momentum (invented events, wrong activity, endless meta-talk such as sign-offs) rather than by the video.

degen (0–2, lower is better)

Repetition and looping degradation.

- 0 = no meaningful repetition.
- 1 = noticeable repeated phrases or restarts, but new content continues.
- 2 = heavy looping: long spans are near-verbatim repeats with little or no new content.

coverage (0–2, higher is better)

How much of the task's content, as listed in the timeline, gets narrated at all (at any time).

- 0 = almost nothing substantive is narrated.
- 1 = some steps are narrated, with large gaps or missing phases.
- 2 = most steps in the timeline are narrated.

coherence (0–3, higher is better)

Readability and logical flow of the narration as a text.

- 0 = incoherent fragments.
- 1 = locally readable but disorganized or self-contradictory.
- 2 = mostly coherent with minor breaks.
- 3 = coherent throughout.

Dimensions are independent. In particular alignment and drift are separable: on-task-but-mistimed is alignment-bad / drift-good; fluent invention is drift-bad and usually alignment-bad too. Score each dimension on its own.

Calibration anchors (frozen, inline)

Three hand-written prototypes pin the alignment scale. Use the FULL range—do not hedge toward the middle. Each is on a generic task with a stated timeline.

ANCHOR 1 — alignment 0 (on-task AND on-time)

`video_class`: Make a Cup of Tea

`timeline`: 0–14 boil water / 15–29 pour over tea bag / 30–49 steep / 50–60 add milk and stir

`SCORES`: alignment=0, drift=0, degen=0, coverage=2, coherence=3

```
[1] We fill the kettle and switch it on to bring the water to a boil.
[16] The water is ready, so we pour it over the tea bag in the mug.
[32] We let the tea steep for a couple of minutes to draw out the
colour.
[51] Finally we add a splash of milk and give it a gentle stir.
```

Each line names exactly the step the timeline places at that moment. alignment 0.

ANCHOR 2 — alignment 3 (on-task but a SUSTAINED whole-video lag)

`video_class`: Make a Cup of Tea

Same timeline.

`SCORES`: alignment=3, drift=0, degen=0, coverage=2, coherence=3

```
[16] We fill the kettle and switch it on to bring the water to a boil.
[34] Now the water is ready, so we pour it over the tea bag.
[52] We let the tea steep for a couple of minutes.
[60] And now we add a splash of milk and give it a stir.
```

Every line is a real tea step (drift 0), but each is spoken a full step-length AFTER it actually happens: boiling is narrated during the pour, pouring during the steep, and so on across the WHOLE video. The narration never coincides with the current step—a sustained, whole-video temporal mismatch. This is alignment 3, NOT 2: a large consistent lag over the entire clip means the narration tracks its own clock, not the timeline.

ANCHOR 3 — alignment 3 (off-task)

video_class: Replace Filter For Air Purifier

SCORES: alignment=3, drift=3, degen=0, coverage=0, coherence=3

```
[0] You,
[4] I am Akshay, and today we will be reviewing
[7] the Maruti Suzuki Swift 2021 petrol model.  We got the sporty
[11] SX variant.  It's a 1st Gen Swift which is now
[15] discontinued in India.  The new generation swift is coming with
```

Video shows air-filter replacement, but narration is a self-driven CAR REVIEW describing an activity the video never shows: drift 3, alignment 3, coverage 0.

Output

Return ONLY a JSON list, one object per video in the pack, with integer scores for alignment, drift, degen, coverage, and coherence for transcripts A–F. No prose or explanations are permitted.

Pack

{{PACK}}