VIDEO-RSI: RECURSIVE SELF-IMPROVEMENT OF VIDEO UNDERSTANDING AGENTS VIA HARNESS EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video understanding agents acquire evidence through an executable harness that controls what they observe and how they use those observations. However, execution traces contain only the evidence acquired by the current harness, leaving competing explanations for failure unresolved and limiting the basis for self-improvement. We introduce Video-RSI, a framework for recursive self-improvement in which a video understanding agent uses its own language model to revise its harness. Through active video investigation, the model revisits the original training videos to test competing failure explanations with additional observations, grounding proposed changes in evidence beyond the existing trace. Cost-aware harness evolution turns these diagnoses into reusable revisions and determines which revisions to retain by considering both answer accuracy and visual cost. Across our evaluation settings on video understanding benchmarks, the evolved agent improves accuracy while processing fewer frames and achieves competitive accuracy-efficiency trade-offs against existing video understanding agents. These results demonstrate the potential for video understanding agents to improve their own evidence acquisition and use through harness evolution.

1 INTRODUCTION

Video understanding agents must decide what to observe before they can decide how to answer. In long videos, relevant events may be brief, temporally distant, or surrounded by visually similar distractions. Recent agents address this challenge through iterative retrieval, temporal navigation, and tool-guided observation (Wang et al., 2024; Zhang et al., 2025b; Lin et al., 2026). At each step, the agent must choose where to look, which details to inspect, and whether the evidence is sufficient to answer. These decisions are governed by an executable *harness* that controls evidence acquisition, processes observations, and supports subsequent reasoning. The harness shapes both what the model sees and how it uses that evidence, linking answer accuracy to the cost of visual observation. Improving this executable layer is therefore an opportunity to strengthen video understanding without changing model weights.

Automated agent design uses feedback to improve agent programs (Hu et al., 2025; Lee et al., 2026), and recent work extends code-level evolution to video understanding (Xu & Chen, 2026). Self-Harness further shows that a frozen model can revise its own operating harness in terminal environments (Zhang et al., 2026a). We study this form of *harness self-improvement* for video understanding. The language model that answers video questions also investigates failures and revises the code governing its behavior. As shown in Figure 1, adaptive evidence gathering changes what an agent observes within a question, while harness self-improvement changes the procedure it uses across subsequent questions. This makes execution experience a basis for revising how evidence is acquired and organized, allowing useful changes to accumulate in the program. Our objective is to turn execution experience into reusable improvements that increase answer accuracy while reducing visual observation, with the underlying models fixed.

This objective requires both evidence for a revision and a criterion for deciding whether to retain it. An execution trace contains only the observations acquired by the current harness. A missing event may therefore reflect a sampling gap, a perceptual error, or a failure to use available evidence, each

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.



(a) MLLMs (b) Video agents (c) Video-RSI

Figure 1: From answering video questions to improving the harness. (a) Direct MLLM inference uses supplied observations. (b) Video agents adapt evidence acquisition within a question. (c) Video-RSI uses the same frozen language model to investigate execution failures, revisit training videos, and revise the harness offline. Revisions are selected by accuracy and visual cost, producing a reusable harness for subsequent questions.

requiring a different modification. For instance, an event absent from the recorded observations does not establish whether the agent failed to inspect it or misinterpreted what it saw. Revisiting the video can help distinguish these explanations before the code is changed. Yet correcting a failure does not necessarily produce a better harness. Denser sampling may improve an answer while increasing visual cost across many questions, whereas better evidence reuse may avoid redundant inspection. Useful self-improvement must connect a supported failure diagnosis to an assessment of the revised harness's accuracy and visual cost over complete executions.

We introduce **Video-RSI**, a framework in which the same frozen language model serves as both Solver and Editor to recursively improve a video agent's executable harness. *Active Video Investigation* revisits training videos to test failure explanations and inform code revisions. *Cost-Aware Harness Evolution* evaluates the resulting candidates by answer accuracy and visual cost, retaining revisions that become the starting point for further improvement. These mechanisms link evidence for what to change with criteria for which changes should accumulate. The resulting loop can revise observation strategies and evidence-processing routines together, assessing their combined effect on the behavior of the complete agent. Evolution takes place offline. The selected harness is then frozen for evaluation on new questions. Across four video understanding benchmarks, the evolved harness improves accuracy while processing fewer frames than its initial version and achieves competitive accuracy–efficiency trade-offs against existing video agents. On MLVU, active investigation also yields higher final accuracy than trace-only revision at similar inference-time frame usage.

Our contributions are threefold:

- We introduce Video-RSI, a framework for video harness self-improvement in which the same language model solves video questions, investigates failures, and revises its executable harness while model weights remain fixed.
- We develop an improvement loop that links active video investigation to diagnosis-guided code revision and accuracy–cost selection, addressing both the evidence for a modification and its value for subsequent executions.
- We evaluate Video-RSI across four video understanding benchmarks and analyze investigation and evolution dynamics, showing improvements over the initial harness in both accuracy and frame usage and the benefit of investigation over trace-only revision in the MLVU comparison.

2 Related Work

2.1 Agentic Video Understanding

Video agents address long videos by deciding which evidence to acquire and how to organize it for reasoning. Iterative retrieval and hierarchical representations allow VideoAgent and VideoTree to concentrate on question-relevant content (Wang et al., 2024; 2025b), while VCA guides exploration with intrinsic rewards and a bounded visual memory (Yang et al., 2025). Beyond frame selection, document retrieval (Ma et al., 2025) and multi-granular tool use (Zhang et al., 2025b; Lin et al., 2026) connect video observations to successive reasoning steps. DVD combines global browsing, clip search, and direct frame inspection. VideoSeek provides complementary overview, skim, and focus operations. MR.Video uses MapReduce to reconcile entities and aggregate question-specific analyses across segments (Pang & Wang, 2025), while LVAgent combines the observations and reasoning of multiple agents through dynamic collaboration (Chen et al., 2025).

Another line of work learns observation policies through model training. FrameThinker uses supervised fine-tuning and reinforcement learning to coordinate reasoning with additional frame acquisition (He et al., 2026). EVA further learns to allocate temporal windows, frame counts, and spatial resolution, and uses failure cases to guide the generation of additional training questions (Zhang et al., 2026d). Video-RSI instead keeps the underlying models fixed and evolves the executable harness that governs evidence acquisition and use. Its improvement loop revisits training videos to diagnose failures before making program changes that apply across questions.

2.2 Automated Agent Design and Self-Improvement

Automated agent design treats the programs surrounding language models as an optimization space. ADAS searches over agents expressed in code, while AFlow searches executable workflows (Hu et al., 2025; Zhang et al., 2025a). These formulations allow changes to the composition and control of model calls beyond isolated prompt edits. Meta-Harness extends this direction by using historical code, evaluation scores, and execution traces to optimize harnesses around a fixed model (Lee et al., 2026). This program-level view motivates our choice of an executable harness as the object of improvement, encompassing both evidence-processing tools and the logic that uses their outputs.

Self-improvement also concerns which system proposes the changes. STOP studies an improvement program that modifies itself (Zelikman et al., 2024), and the Darwin Gödel Machine combines self-modification with exploration over an archive of agents (Zhang et al., 2026b). Self-Harness uses the same frozen model to propose harness revisions and validates candidates through regression testing (Zhang et al., 2026a). Video-RSI builds on this code-level self-improvement setting, with the same language model serving as Solver and Editor. Our focus is the evidence available to that Editor. An execution trace records only what the current harness observed, so investigating the original video can distinguish failure explanations that would otherwise motivate different revisions.

2.3 Self-Evolving Video Agents

Video self-improvement has been explored through the generation and refinement of training signals. EvoGround develops a proposer–solver learning loop for temporal grounding, Video-Zero organizes questioner–solver co-evolution around local temporal evidence, and EvoVid emphasizes temporal information in self-evolution (Jung et al., 2026; Zhang et al., 2026c; Huang et al., 2026). These methods improve video capabilities through model training. Video-RSI instead accumulates improvements in executable code while keeping model weights fixed.

VideoHarness-RSI is a closely related approach to video harness evolution (Xu & Chen, 2026). It evolves context-construction programs that can coordinate retrieval, evidence representations, and auxiliary model calls around frozen vision–language models, and promotes candidates that improve validation accuracy. Video-RSI shares this program-level search setting and focuses on how the Editor obtains evidence for a revision. Active investigation of the original videos tests failure explanations before code is edited. Our selection rule also accounts for visual cost, admitting accuracy gains with bounded cost growth or cost reductions with bounded accuracy loss. Together, investigation and selection connect reusable harness changes to their effects on answer quality and observation requirements.

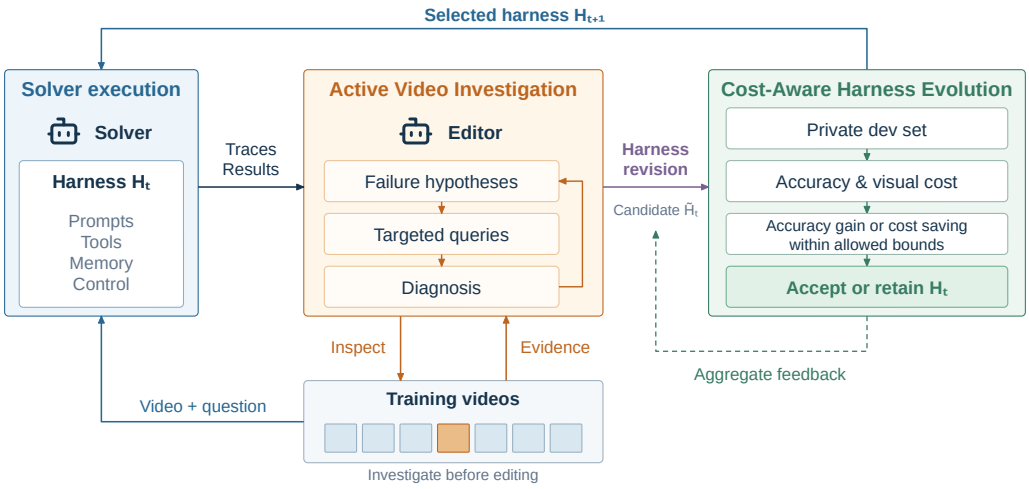


Figure 2: Overview of Video-RSI. The Solver and Editor share a frozen language model. Active video investigation revisits training videos to diagnose failures and guide harness revisions. Cost-aware harness evolution selects candidates using accuracy and visual cost on a private development set, carrying accepted revisions into subsequent iterations.

3 VIDEO-RSI

Video-RSI evolves the executable harness of a video question-answering agent while keeping the underlying models fixed. An Editor revisits the original video to investigate failures, uses the resulting diagnosis to revise the harness, and submits the candidate to a performance–cost gate. Accepted revisions become the starting point for the next evolution attempt. Figure 2 provides an overview.

3.1 PROBLEM FORMULATION AND EVOLVABLE HARNESS

A harness H governs how an agent acquires and uses evidence to answer a video question. Given a video V, a question q with answer choices, and fixed model services $\mathcal{M}$, the Solver produces an answer $\hat{y}$, an execution trace τ, and a resource record c:

$$(\hat{y}, \tau, c) = \mathrm{Run}(H, V, q; \mathcal{M}). \tag{1}$$

The trace records tool calls, observations, and decisions. The Editor can jointly revise the harness code for prompt construction, tool implementation, observation processing, memory, and execution control.

These components connect evidence acquisition to decision-making by determining what to inspect, how observations are represented and retained, and when to gather more evidence or answer. Their coordination matters because even a correct observation can lose its temporal or entity associations before reaching the answering step. Harness evolution can therefore change both the observations acquired and how they support reasoning.

We use a training set $\mathcal{D}_{\mathrm{tr}}$ for investigation and revision and a fixed, video-disjoint selection set $\mathcal{D}_{\mathrm{g}}$. Selection examples and per-item outcomes remain private to the evaluator.

3.2 FRAMEWORK OVERVIEW

Video-RSI uses the same frozen language model as Solver and Editor, accumulating improvements in executable code (Xu & Chen, 2026). At each attempt, the Solver executes H_t on training questions. The Editor examines its traces and revisits videos to test failure explanations before proposing a candidate $\widetilde{H}_t$. Independent evaluation on the private selection set determines whether the candidate replaces H_t for the next attempt. Investigation occurs offline. Additional observations guide reusable code changes, while the selected harness answers subsequent questions using their own inputs.

3.3 Active Video Investigation

An execution trace may not reveal whether an incorrect answer stems from missing an event, misperceiving it, or failing to use existing evidence. Active investigation lets the Editor revisit the original video to distinguish these explanations before revising the harness.

From failure explanations to diagnostic queries. The Editor examines training outcomes, traces, and code, prioritizing representative failures that additional evidence could clarify. Queries vary the temporal region, sampling density, modality, or perceptual question to test competing explanations. For example, denser inspection can reveal whether a short event fell between sampled frames. The interface supports video observations, speech transcripts, on-screen text, and reruns of the unmodified incumbent. Perception requests omit the gold answer.

Let $\mathcal{I}_0$ contain the current code, training traces, available labels, and prior feedback. The Editor selects queries u_k and accumulates returned evidence o_k:

$$\begin{aligned} u_k &= \mathrm{Query}_{\mathrm{Editor}}(\mathcal{I}_k), \\ o_k &= \mathrm{Investigate}(u_k; H_t, \mathcal{D}_{\mathrm{tr}}, \mathcal{M}), \\ \mathcal{I}_{k+1} &= \mathcal{I}_k \cup \{(u_k, o_k)\}. \end{aligned} \tag{2}$$

Each query specifies a training example and the observation needed to test an explanation. The Editor interprets the accumulated evidence to choose the next query, while H_t remains unchanged.

From new evidence to a revision hypothesis. New observations can support, contradict, or leave an explanation unresolved, redirecting subsequent queries. Finding a missed event supports revising temporal coverage. If it was already observed, the diagnosis instead examines how its evidence was represented and used. Investigation ends with a supported revision hypothesis or an unresolved case when no further informative query is available.

The Editor compares relevant cases to identify a reusable change, its applicable conditions, and successful behaviors to preserve. Training annotations may locate informative intervals during diagnosis, but the revision must use cues available to the Solver on new questions. The diagnosis therefore connects observed failures to a procedure for finding and using evidence without knowing the answer or event location in advance.

3.4 Cost-Aware Harness Evolution

Diagnosis-guided revision. The Editor maps the diagnosis to code changes, specifying when they apply and which successful behaviors to preserve. Revisions can span the harness components in Section 3.1, such as changing inspection cues or preserving temporal relationships in observations and memory. A reusable change may require coordinated edits to a tool's output and the code that consumes it. The Editor can debug candidates through training-side trials, distinct from the pre-edit investigation, before submitting a candidate for selection.

Cost-aware selection. After execution checks, the evaluator compares the candidate and incumbent on the same private selection examples under matched conditions. It measures answer accuracy and visual cost, defined as mean frames processed by successful visual-model calls over complete question-answering executions. This end-to-end assessment matters because a more elaborate tool can reduce later inspection, while a locally useful repair can increase overall visual usage.

Let A_t, C_t denote the incumbent's accuracy and mean frame count on the selection set, and $\widetilde{A}_t, \widetilde{C}_t$ the corresponding candidate measurements. With $\Delta A_t = \widetilde{A}_t - A_t$, the gate admits two improvement directions:

$$\begin{aligned} g_t = \underbrace{[\Delta A_t > 0] \wedge [\widetilde{C}_t \le (1+\alpha) C_t]}_{\text{accuracy gain with bounded cost growth}} \\ \vee \underbrace{[-\epsilon \le \Delta A_t \le 0] \wedge [\widetilde{C}_t \le (1-\beta) C_t] \wedge [C_t > 0]}_{\text{cost reduction with bounded accuracy loss}}. \end{aligned} \tag{3}$$

Here, $\alpha \ge 0$ bounds relative cost growth, $0 < \beta < 1$ sets the required relative cost reduction, and $\epsilon \ge 0$ bounds accuracy loss. Their values are fixed during evolution and specified in the experimental

setup. If $g_t = 1$, the candidate becomes H_{t+1}. Otherwise, $H_{t+1} = H_t$. The Editor receives only aggregate accuracy and frame counts to guide subsequent revisions. Per-example gate traces remain private. The selected harness is finally evaluated on held-out data. Algorithm 1 summarizes the complete evolution process.

Algorithm 1 Recursive harness evolution in Video-RSI

Input: initial harness H_0, fixed services $\mathcal{M}$, training set $\mathcal{D}_{\mathrm{tr}}$, private gate $\mathcal{D}_{\mathrm{g}}$, attempt budget T.
Output: final accepted harness H_T.

for $t = 0, \ldots, T-1$ **do**
  Execute H_t on training examples and inspect its traces.
  Initialize investigation context $\mathcal{I}_0$ and set $k = 0$.
  while further investigation is warranted and resources permit **do**
    Select a query to distinguish unresolved failure explanations.
    Acquire evidence and construct $\mathcal{I}_{k+1}$ using Eq. (2).
    Revise the supported explanations and remaining questions.
    Set $k \leftarrow k + 1$.
  end while
  Synthesize a diagnosis and plan a reusable behavior change.
  Edit H_t into a candidate $\widetilde{H}_t$ and run training-side checks.
  Validate and compare the candidate with H_t on $\mathcal{D}_{\mathrm{g}}$.
  Accept or retain the incumbent according to Eq. (3).
end for
return H_T

4 Experiments

4.1 Experimental Setup

Benchmarks and data splits. We evaluate on the long-video splits of Video-MME (Fu et al., 2025) and LongVideoBench (Wu et al., 2024) with subtitles, the official public subset of EgoSchema (Mangalam et al., 2023), and MLVU test (Zhou et al., 2025). For evolution, we use 216 LVBench questions (Wang et al., 2025a) for investigation and revision, and another 72 questions as a fixed, video-disjoint development set for candidate selection. Development examples and per-question outcomes remain private. The Editor receives only aggregate feedback.

Models and evolution protocol. The Solver and Editor use DeepSeek-V4-Pro, with Qwen3.6-Plus for visual observations. All model weights are frozen. The initial harness H_{S0} supports global and local video inspection, transcripts, OCR, and observation memory. We run 20 revision attempts using Eq. (3) with $\alpha = 0.1$, $\beta = 0.2$, and $\epsilon = 1/|\mathcal{D}_{\mathrm{g}}|$, allowing at most one fewer correct answer in the cost-reduction branch. The final accepted harness is frozen for the main evaluation, while intermediate accepted versions are evaluated to analyze evolution dynamics. Active-investigation and trajectory-only variants share training examples, annotation access, initialization, models, and per-attempt resource limits, differing only in pre-edit access to new video observations.

Baselines. We compare with MLLMs, video agentic models, and harness self-improvement methods. We reproduce VideoSeek (Lin et al., 2026) and VideoHarness-RSI (Xu & Chen, 2026) using the same model (DeepSeek-V4-Pro) as Video-RSI. Results for the remaining methods are taken from their original publications. The initial harness H_{S0} serves as the reference for evolution gains. Trajectory-only revision and an accuracy-only gate assess the contributions of investigation and candidate selection.

Evaluation metrics. We report multiple-choice accuracy and mean visual frames per question. For our evaluations, frame usage sums all frames processed by successful visual-model calls during each complete execution, then averages across questions. Offline Editor observations are excluded.

Table 1: Video understanding results. Acc. denotes accuracy (%) and Frames denotes visual frame usage. Bold indicates column-best values. † denotes official M-Avg.

Method	MLVU Test		LongVideoBench Long val		Video-MME Long		EgoSchema	
	Acc.↑	Frames↓	Acc.↑	Frames↓	Acc.↑	Frames↓	Acc.↑	Frames↓
MLLMs								
GPT-4o	54.9†	0.5 fps	60.9	256	65.3	384	70.4	–
Gemini 1.5 Pro	–	–	58.6	256	67.4	1233	–	–
Gemini 2.0 Flash	–	–	45.7	256	63.0	1233	71.2	–
Qwen2.5-VL-72B	–	–	–	–	53.2	256	–	–
GPT-5	–	–	64.5	384	67.9	384	–	–
Video Agentic Models								
VideoAgent	–	–	–	–	46.4	24.6	60.2	8.4
VideoTree	–	–	–	–	53.1	98.0	66.2	62.4
DrVideo	–	–	–	–	51.7	493.2	66.4	–
VCA	–	–	–	–	54.2	18.1	73.6	**7.2**
MR.Video	–	–	61.6	–	63.4	–	73.8	–
DVD	–	–	68.6	2816	67.3	4932	76.6	–
FrameThinker	–	–	–	–	47.6	24.1	–	–
LVAgent	–	–	–	–	74.3	–	–	–
EVA-GRPO	–	–	–	–	48.4	22.8	–	–
VideoSeek	68.1	83.9	70.9	71.9	78.7	**17.8**	73.2	72.3
Harness Self-Improvement								
VideoHarness-RSI	64.1	83.2	68.4	92.1	69.7	92.8	78.0	66.8
Video-RSI (Ours)	**72.9**	**61.1**	**71.1**	**41.1**	**80.0**	21.6	**78.2**	42.4

4.2 Main Results

Table 1 places Video-RSI alongside MLLMs, video agentic models, and harness self-improvement methods. Our analysis focuses on the reproduced VideoSeek and VideoHarness-RSI implementations. Published results provide broader context. With its harness evolved on LVBench and frozen for evaluation, Video-RSI achieves higher accuracy than both reproduced baselines across all four benchmarks, with lower frame usage in all but the VideoSeek comparison on Video-MME.

Relative to VideoSeek, the accuracy–cost balance varies across benchmarks. On MLVU, Video-RSI gains 4.8 percentage points while using 27.2% fewer frames, combining better answers with reduced visual usage. On LongVideoBench, the main benefit is substantially fewer frames at similar accuracy. On EgoSchema, both accuracy and frame usage improve. On Video-MME, higher accuracy accompanies a moderate increase in frame usage. Thus, the advantage is not a uniform reduction in observation, but a favorable balance between answer quality and visual usage across different evaluation settings.

Compared with VideoHarness-RSI, Video-RSI achieves higher accuracy with fewer frames on every benchmark. The accuracy gain is largest on Video-MME, whereas EgoSchema shows closely comparable accuracy with substantially lower frame usage. The evolved harness thus offers two practical benefits by improving answer quality and reducing the observations needed for comparable performance. The controlled comparisons in Section 4.3 further examine how active investigation and cost-aware selection contribute to the resulting harness.

4.3 Ablation Study

Table 2 examines how harness evolution depends on the evidence available before revision and the criterion used to retain candidates. The trajectory-only variant restricts the Editor to existing execution traces, while active investigation allows additional video observations before editing. Both

Table 2: Ablations of active investigation and candidate selection. Acc. is accuracy (%). Frames is mean visual frames per question.

Variant	MLVU Test		LongVideoBench Long val		Video-MME Long		EgoSchema	
	Acc.↑	Frames↓	Acc.↑	Frames↓	Acc.↑	Frames↓	Acc.↑	Frames↓
Initial harness	62.4	68.5	67.7	51.6	73.4	38.6	71.4	66.0
Trajectories only	64.7	**60.3**	67.2	43.1	73.7	28.5	76.6	51.8
Accuracy-only gate	67.1	81.9	69.5	66.4	74.9	43.1	77.4	60.5
Video-RSI	**72.9**	61.1	**71.1**	**41.1**	**80.0**	**21.6**	**78.2**	**42.4**

use 20 revision attempts under the shared conditions in Section 4.1. The accuracy-only gate variant examines selection without the visual-cost criterion. We evaluate the resulting harnesses on all four benchmarks, using the initial harness as a reference.

Active investigation improves the resulting harness. Active investigation yields higher final accuracy than trajectory-only revision on every benchmark. On MLVU, the gain reaches 8.2 percentage points with similar inference-time frame usage. On the other three benchmarks, higher accuracy accompanies fewer frames. This pattern links additional evidence during offline revision to better deployed behavior without a consistent increase in inference-time observation. The investigation-enabled run also retains five of twenty revisions, compared with three for trajectory-only revision. Alongside the endpoint results, these observations support using evidence beyond the existing trace to guide reusable harness changes.

Cost-aware selection improves the accuracy–cost balance. The accuracy-only gate improves accuracy over the initial harness on all four benchmarks, but increases frame usage on three. Video-RSI achieves higher accuracy than this variant while using fewer frames on every benchmark. On Video-MME, it uses roughly half as many frames. Thus, the cost-aware run obtains visual savings without sacrificing endpoint accuracy in these evaluations. The comparison shows why accuracy gains alone do not fully characterize a useful harness revision. Selecting for both objectives can retain more efficient behavior while preserving improvements in answer quality.

4.4 Evolution Dynamics

Figure 3 tracks the retained harness over 20 revision attempts on the development set and MLVU. The MLVU evaluations were performed retrospectively after evolution and did not inform candidate selection. On the development set, the first accepted revision produces most of the net reduction in frame usage, while subsequent revisions steadily improve accuracy with smaller, nonmonotonic changes in visual cost. The run thus combines an early efficiency gain with continued refinement of answer quality.

The MLVU trajectory reveals that improvements on the selection set need not transfer at every step. The first accepted revision reduces frame usage but lowers MLVU accuracy, despite improving both development metrics. Accuracy recovers with the next revision, and the final two accepted revisions account for most of the net MLVU accuracy gain. The final harness therefore reflects cumulative refinement beyond the initial efficiency improvement. An early development gain alone would give an incomplete picture of its evaluation behavior.

Accuracy and visual cost also follow different paths during later refinement. From H_{S2} onward, MLVU accuracy rises together with frame usage, yet the final harness remains more accurate and uses fewer frames than H_{S0}. This pattern is consistent with the selection rule, which permits bounded cost increases for accuracy gains, rather than requiring frame usage to decrease at every update. Later revisions can use more frames while retaining a net efficiency advantage over the initial harness, improving the final accuracy–cost balance.

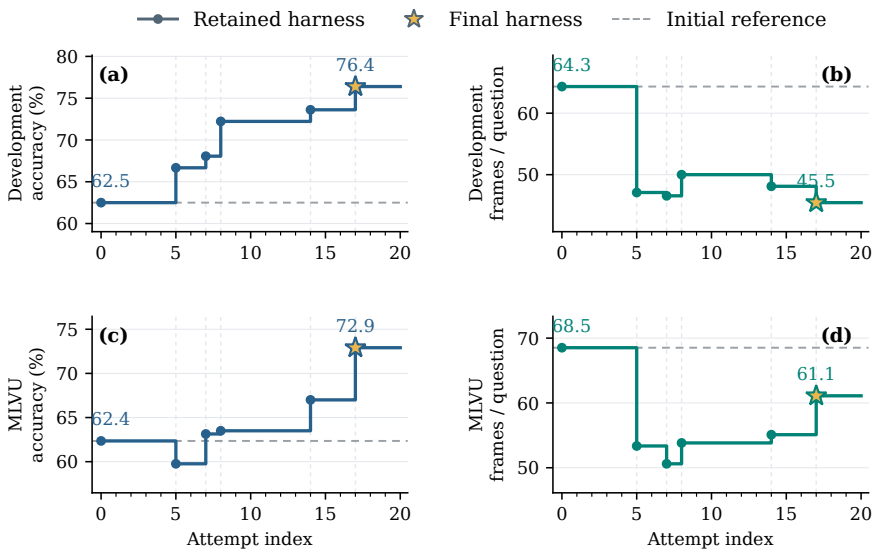


Figure 3: Evolution over 20 revision attempts. Accuracy and mean frames per question on (a,b) the development set and (c,d) MLVU. Curves track the retained harness. Vertical guides mark accepted revisions, and flat intervals indicate an unchanged incumbent.

4.5 Analysis of Evolved Harnesses

Comparing the initial harness with the retained revisions reveals three changes in how the Solver acquires and uses evidence:

Structured perception. In H_{S2}, free-form visual responses give way to goal-directed observations with structured findings, image references, and explicit uncertainty. The harness maps image references to frame timestamps, preserving the link between a finding and its temporal evidence. This interface lets follow-up reasoning target unresolved details and their supporting frames without changing model weights.

Evidence reuse through relation graphs. Introduced in H_{S4}, the graph mechanism extracts entities, events, and relations from accumulated visual, transcript, and OCR observations. For eligible relational or temporal questions, it returns evidence gaps that can guide further inspection. Graph queries themselves process no new frames, making evidence organization an intermediate step between existing observations and additional acquisition.

Selective observation routines. The Global-64 route in H_{S3} expands the initial 32-frame overview for selected whole-video questions. The occurrence tool in H_{S5} defines counting units, verifies candidate events, and consolidates overlapping evidence to avoid duplicate counts. These routines coexist with transcript-based localization and observation reuse. The final harness improves accuracy while using 10.8–44.1% fewer frames than H_{S0} across the four benchmarks, showing that richer observation tools can coexist with lower aggregate visual usage.

These retained changes illustrate how evolution modifies both evidence acquisition and its use in reasoning. The reported savings measure visual frames rather than total model computation.

5 Conclusion

We presented Video-RSI, a framework for improving video understanding agents through harness evolution with fixed model weights. The same language model serves as Solver and Editor, revisiting training videos to test failure explanations and translating the resulting diagnoses into reusable code changes. Cost-aware selection retains revisions according to both answer accuracy and visual usage. Across four benchmarks, the evolved harness improves accuracy while using fewer frames than its initial version, with ablations supporting the contributions of active investigation and cost-aware selection. Analysis of the retained code reveals changes in structured perception, evidence reuse, and selective observation. Together, these findings support harness evolution as a practical route to improving how frozen-model video agents acquire and use evidence.

AI USE STATEMENT

We used generative AI tools to assist with code editing, figure and table preparation, and reviewing the manuscript for clarity of expression. The authors reviewed the AI-assisted content and take responsibility for the final manuscript, code, and scientific claims.

REFERENCES

Boyu Chen, Zhengrong Yue, Siran Chen, Zikang Wang, Yang Liu, Peng Li, and Yali Wang. LVAgent: Long video understanding by multi-round dynamical collaboration of MLLM agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20237–20246, October 2025. doi: 10.1109/ICCV51701.2025.01882. URL https://doi.org/10.1109/ICCV51701.2025.01882.

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

Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. Framethinker: Learning to think with long videos via multi-turn frame spotlighting. In *International Conference on Learning Representations*, volume 2026, pp. 74904–74933, 2026.

Shengran Hu, Cong Lu, and Jeff Clune. Automated Design of Agentic Systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/36b7acf6f6010652b3f2a433774a66fe-Abstract-Conference.html.

Shiqi Huang, Ziyue Wang, Zhongrong Zuo, Han Qiu, Qi She, and Bihan Wen. EvoVid: Temporal-Centric Self-Evolution for Video Large Language Models, 2026. URL https://arxiv.org/abs/2605.21931.

Minjoon Jung, Byoung-Tak Zhang, and Lorenzo Torresani. EvoGround: Self-Evolving Video Agents for Video Temporal Grounding, 2026. URL https://arxiv.org/abs/2605.13803.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-End Optimization of Model Harnesses, 2026. URL https://arxiv.org/abs/2603.28052.

Jingyang Lin, Jialian Wu, Jiang Liu, Ximeng Sun, Ze Wang, Xiaodong Yu, Jiebo Luo, Zicheng Liu, and Emad Barsoum. VideoSeek: Long-horizon video agent with tool-guided seeking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5465–5475, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_VideoSeek_Long-Horizon_Video_Agent_with_Tool-Guided_Seeking_CVPR_2026_paper.html.

Ziyu Ma, Chenhui Gou, Hengcan Shi, Bin Sun, Shutao Li, Hamid Rezatofighi, and Jianfei Cai. DrVideo: Document retrieval based long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18936–18946, June 2025. doi: 10.1109/CVPR52734.2025.01764. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ma_DrVideo_Document_Retrieval_Based_Long_Video_Understanding_CVPR_2025_paper.html.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46212–46244, 2023.

Ziqi Pang and Yu-Xiong Wang. MR. Video: MapReduce as an effective principle for long video understanding. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 6640–6672. Curran Associates, Inc., 2025. doi: 10.52202/085713-0231. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0a02c2bc2e2148b803c4ade1d71e1d25-Paper-Conference.pdf.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. LVBench: An Extreme Long Video Understanding Benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, October 2025a. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_LVBench_An_Extreme_Long_Video_Understanding_Benchmark_ICCV_2025_paper.html.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *Computer Vision – ECCV 2024*, volume 15138 of *Lecture Notes in Computer Science*, pp. 58–76. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72989-8_4. URL https://doi.org/10.1007/978-3-031-72989-8_4.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3283. IEEE, 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857, 2024.

Guoyang Xu and Hao Chen. VideoHarness-RSI: Recursive harness self-improvement for long-video understanding with frozen vision-language models, 2026. URL https://arxiv.org/abs/2608.24302.

Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. VCA: Video curious agent for long video understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20168–20179, 2025.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. Self-taught optimizer (STOP): Recursively self-improving code generation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=46Zgqo4QIU.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-Harness: Harnesses That Improve Themselves, 2026a. URL https://arxiv.org/abs/2606.09498.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026b.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, XiongHui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025a.

Ruixu Zhang, Deyi Ji, Lanyun Zhu, Xuanyi Liu, Yuxin Meng, Ruihang Chu, and Yujiu Yang. VideoZero: Self-Evolution Video Understanding, 2026c. URL https://arxiv.org/abs/2605.14733.

Xiaoyi Zhang, Zhaoyang Jia, Zongyu Guo, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Deep video discovery: Agentic search with tool use for long-form video understanding. *Advances in Neural Information Processing Systems*, 38:89863–89895, 2025b.

Yaolun Zhang, Ruohui Wang, Jiahao Wang, Yepeng Tang, Xuanyu Zheng, Haonan Duan, Hao Lu, Hanming Deng, and Lewei Lu. EVA: Efficient reinforcement learning for end-to-end video agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12289–12299, 2026d.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, 2025.

A From Active Investigation to Harness Revision and Transfer

The ablation in Section 4.3 assesses the benefit of active investigation. Relative to trajectory-only revision, the final investigation-enabled harness is more accurate on all four evaluation benchmarks, by 8.2, 3.9, 6.3, and 1.6 percentage points on MLVU, LongVideoBench, Video-MME, and EgoSchema, respectively. It also retains five rather than three revisions over 20 attempts. The two cases below address how this benefit can arise. In both cases, the Editor investigates evidence absent from the original Solver trace, identifies a concrete observation failure, and specifies the evidence-gathering behavior a reusable revision should support.

These are training-side investigations. The Editor could use training answers and annotated intervals.

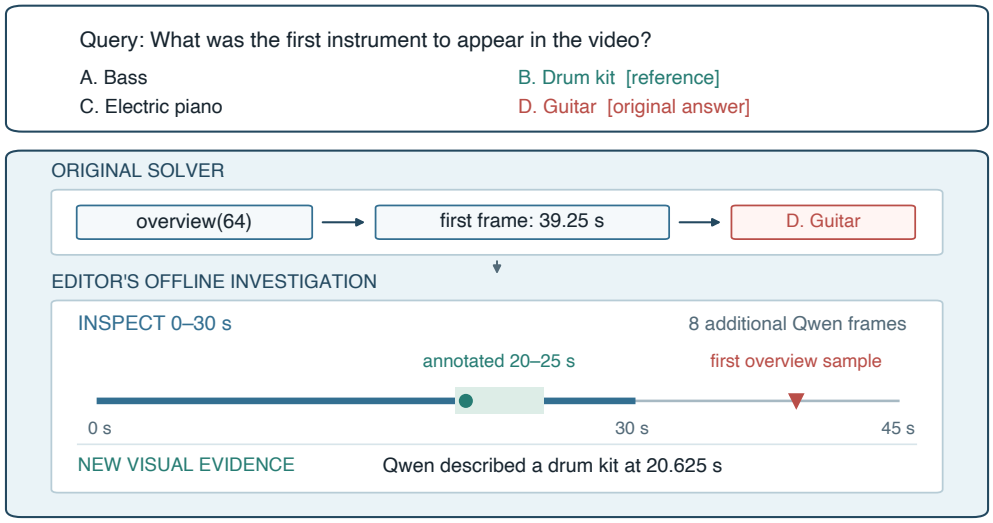


Figure 4: Opening-instrument investigation. The original global overview first sampled the video at approximately 39.25 s, after the annotated 20–25 s target event. The Editor's additional, annotation-guided 0–30 s inspection returned a Qwen description of a drum kit near 20.625 s. The timeline depicts recorded timestamps, not video-frame images.

The original Solver treated its earliest sampled frame as evidence about the video's beginning and selected guitar. Yet that frame was at 39.25 s, after the annotated 20–25 s target event. By inspecting 0–30 s, the Editor obtained a Qwen description of a drum kit at 20.625 s. The investigation thus exposed a coverage failure rather than an instrument-recognition error on observed frames. The round-eight edit plan proposed `start_of_video_localization`, retained in the fixed H_{S5}, to trigger short-window inspection for opening-related questions.

The second case exposes a missing evidence modality. The original Solver used `read_transcript` without visual observation and selected *Turning Tables*. The Editor instead asked whether the song title appeared on screen. Its 225–235 s inspection returned Qwen-reported text containing *Creep* at 228.125 and 229.375 s, a clue unavailable in the transcript-only trace. This identifies a precise observation goal of reading the title, retaining its frame location, and checking performer identity.

Together, the cases show how active investigation can make harness revision more targeted. It exposes missed temporal coverage and missing visual evidence, each suggesting a concrete observation behavior for the Solver. The main-text ablation shows that the investigation-enabled evolution run yields a stronger final harness on all four benchmarks.

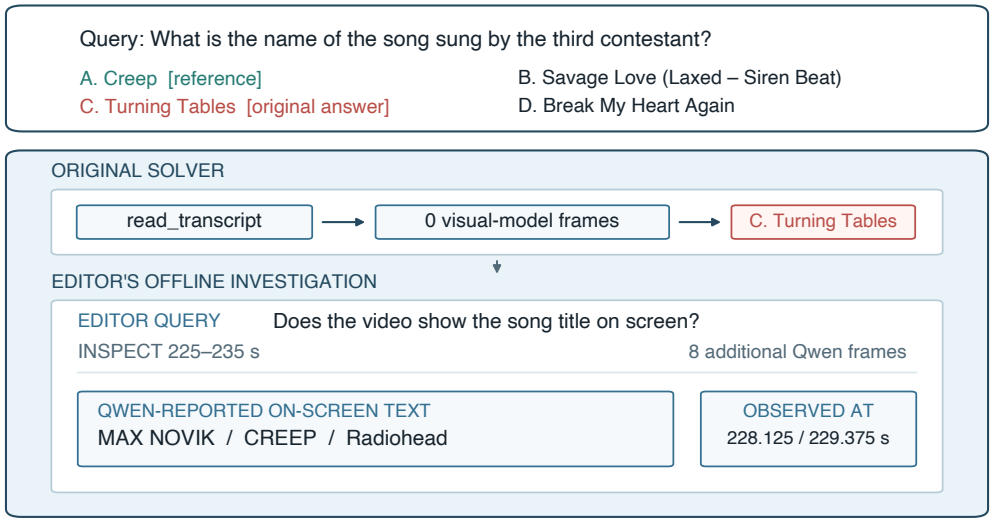


Figure 5: Song-title investigation. The original Solver answered from the transcript without a visual-model call. During an annotation-guided 225–235 s inspection, Qwen reported on-screen text containing *Creep* at 228.125 and 229.375 s. The text panel reproduces the reported observation and is not a video-frame image.

B Performance by Task Type and Video Duration

To identify where harness evolution helps, we compare the initial harness (H_{S0}) with Video-RSI (H_{S5}) by the official task types of MLVU, LongVideoBench Long, and Video-MME Long, then examine MLVU by video duration. In the tables, Δ denotes Video-RSI minus the initial harness. Accuracy differences are in percentage points, and frame differences are in mean visual frames per question. Fix denotes a wrong-to-right transition and Harm a right-to-wrong transition on the same question.

B.1 Breakdown by Task Type

Table 3: Task-type results on MLVU Test. Initial harness and Video-RSI denote H_{S0} and H_{S5}. Δ is their accuracy difference in percentage points. Fix/Harm count paired answer changes.

Task type	N	Accuracy (%)			Answer changes	
		Initial harness	Video-RSI	Δ	Fix	Harm
Anomaly Recognition	39	48.72	64.10	+15.38	6	0
Action Counting	60	25.00	73.33	+48.33	29	0
Egocentric Reasoning	53	60.38	66.04	+5.66	3	0
Needle QA	60	58.33	65.00	+6.67	4	0
Action Order	70	50.00	74.29	+24.29	17	0
Plot QA	50	60.00	68.00	+8.00	4	0
Sports QA	36	66.67	58.33	−8.34	0	3
Topic Reasoning	91	93.41	91.21	−2.20	0	2
Tutorial QA	43	88.37	76.74	−11.63	0	5
Overall	502	62.35	72.91	+10.56	63	10

MLVU task types. On MLVU, Video-RSI improves overall accuracy by 10.56 percentage points, with 63 Fix and 10 Harm (Table 3). The largest gains occur in Action Counting (+48.33 points, $N = 60$) and Action Order (+24.29 points, $N = 70$). Anomaly Recognition also gains 15.38 points. The improvement is especially pronounced on questions that require counting and ordering events.

Table 4: Task-type results on LongVideoBench Long val. Initial harness and Video-RSI denote H_{S0} and H_{S5}. Δ is their difference in accuracy percentage points or frames/question. Fix/Harm count paired answer changes. † marks $N < 30$.

Task type	N	Initial harness		Video-RSI		Δ		Answer changes	
		Acc.	Frames	Acc.	Frames	Acc.	Frames	Fix	Harm
E2O†	12	83.33	53.08	83.33	52.00	0.00	−1.08	0	0
E3E	51	66.67	61.49	68.63	42.82	+1.96	−18.67	9	8
O2E†	24	54.17	70.17	62.50	48.12	+8.33	−22.05	2	0
O3O	30	53.33	68.40	83.33	38.67	+30.00	−29.73	10	1
S2A	30	76.67	49.73	86.67	65.33	+10.00	+15.60	4	1
S2E	42	59.52	64.05	69.05	62.76	+9.53	−1.29	8	4
S2O	33	72.73	67.42	72.73	63.52	0.00	−3.90	4	4
SAA	36	66.67	69.11	58.33	48.83	−8.33	−20.28	1	4
SOS	42	83.33	72.50	83.33	63.10	0.00	−9.40	5	5
SSS	42	35.71	93.43	47.62	53.48	+11.91	−39.95	12	7
T2A	42	92.86	20.81	88.10	13.02	−4.76	−7.79	3	5
T2E	33	75.76	11.09	84.85	8.24	+9.09	−2.85	4	1
T2O†	18	83.33	19.06	83.33	17.89	0.00	−1.17	1	1
T3E	33	75.76	29.12	60.61	19.30	−15.15	−9.82	0	5
T3O	39	66.67	24.26	66.67	19.77	0.00	−4.49	1	1
TAA	36	63.89	30.67	66.67	19.47	+2.78	−11.20	5	4
TOS†	21	47.62	53.86	52.38	68.81	+4.76	+14.95	4	3
Overall	564	67.73	51.59	71.10	41.07	+3.37	−10.52	73	54

LongVideoBench Long. LongVideoBench Long combines a 3.37-point accuracy gain with 10.52 fewer frames per question on average (Table 4). O3O gains 30.00 points while saving 29.73 frames per question, and SSS gains 11.91 points while saving 39.95 frames. Across the 564 questions, 73 Fix and 54 Harm yield a net accuracy gain.

Table 5: Task-type results on Video-MME Long. Initial harness and Video-RSI denote H_{S0} and H_{S5}. Δ is their difference in accuracy percentage points or frames/question. Fix/Harm count paired answer changes. † marks $N < 30$.

Task type	N	Initial harness		Video-RSI		Δ		Answer changes	
		Acc.	Frames	Acc.	Frames	Acc.	Frames	Fix	Harm
Action Reasoning	180	72.78	38.17	78.89	19.04	+6.11	−19.13	21	10
Action Recognition	63	68.25	53.62	69.84	23.71	+1.59	−29.91	8	7
Attribute Perception†	27	81.48	29.63	88.89	16.30	+7.41	−13.33	3	1
Counting Problem	48	54.17	68.85	62.50	62.79	+8.33	−6.06	8	4
Information Synopsis	163	85.28	27.93	88.34	13.60	+3.07	−14.33	14	9
OCR Problems†	14	78.57	37.14	71.43	12.00	−7.14	−25.14	1	2
Object Reasoning	240	75.83	33.80	82.50	17.15	+6.67	−16.65	30	14
Object Recognition	54	72.22	44.15	83.33	18.07	+11.11	−26.08	9	3
Spatial Perception†	3	33.33	42.67	33.33	26.67	0.00	−16.00	0	0
Spatial Reasoning†	11	90.91	36.00	81.82	22.55	−9.09	−13.45	0	1
Temporal Perception†	6	66.67	73.33	100.00	28.00	+33.33	−45.33	2	0
Temporal Reasoning	91	58.24	42.70	73.63	33.78	+15.38	−8.92	20	6
Overall	900	73.44	38.63	80.00	21.58	+6.56	−17.05	116	57

Video-MME Long. Video-MME Long yields a 6.56-point accuracy gain with 17.05 fewer frames per question (Table 5). Temporal Reasoning gains 15.38 points ($N = 91$), Object Recognition gains 11.11 points ($N = 54$), and Counting Problem gains 8.33 points ($N = 48$). Mean frame use falls in every listed task type. The 116 Fix and 57 Harm show that higher answer accuracy accompanies this broad reduction in visual usage.

Taken together, the task-type results show that Video-RSI is particularly effective on event counting and order, as well as temporal and object reasoning. Accuracy improves while mean frame use falls on all three benchmarks, reinforcing the accuracy–efficiency gain in the main results.

B.2 Breakdown by Video Duration

MLVU. Figures 6 and 7 report MLVU results for 502 questions in < 5, 5–15, and ≥ 15 minute groups. Accuracy rises by 10.0, 11.4, and 5.3 percentage points, respectively, while mean frame use falls by 3.2, 7.3, and 11.2 frames per question.

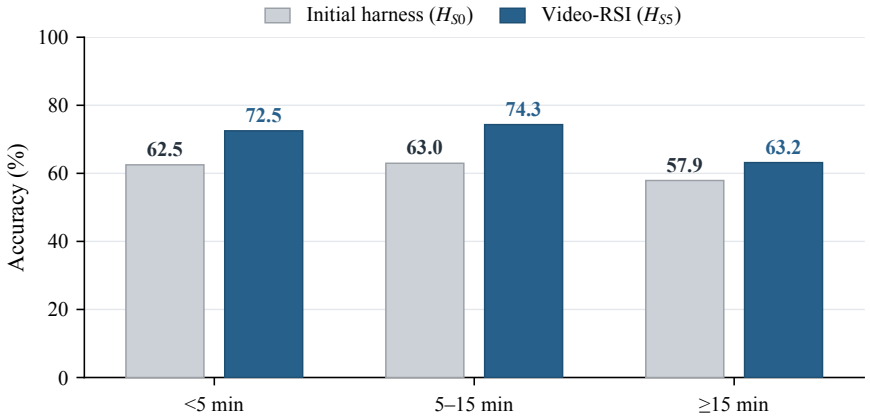


Figure 6: MLVU accuracy by video duration. The < 5, 5–15, and ≥ 15 minute groups contain 40, 405, and 57 questions, respectively. Bars compare the initial harness and Video-RSI within each group. Labels above the bars give percentages.

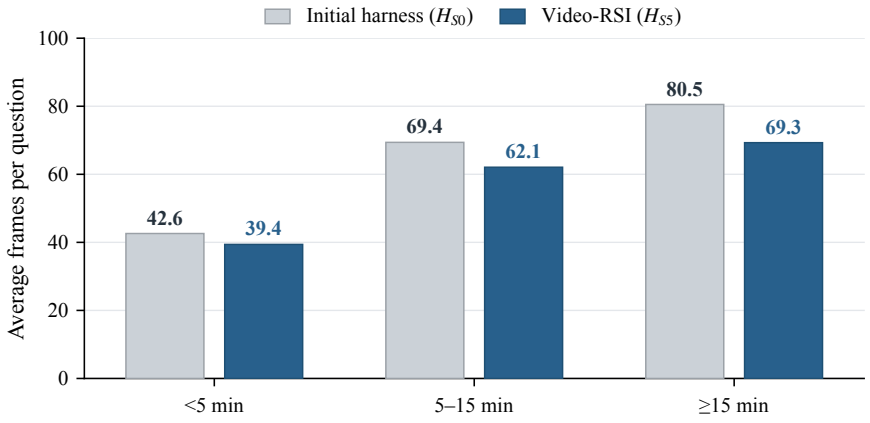


Figure 7: MLVU mean visual frames per question by video duration. The < 5, 5–15, and ≥ 15 minute groups contain 40, 405, and 57 questions, respectively. Bars compare the initial harness and Video-RSI within each group.

Across all three MLVU duration groups, Video-RSI improves accuracy while using fewer visual frames on average. The ≥ 15-minute group saves the most frames per question, making the efficiency gain most pronounced on the longest videos in this evaluation.

C Stability Across Complete Evolutions

To assess the stability of complete Video-RSI evolution, we perform two additional runs alongside the main run. One additional evolution accepted five harness revisions and the other accepted six. We evaluate the final Solver harness from each run. Figure 8 reports the accuracy of all three.

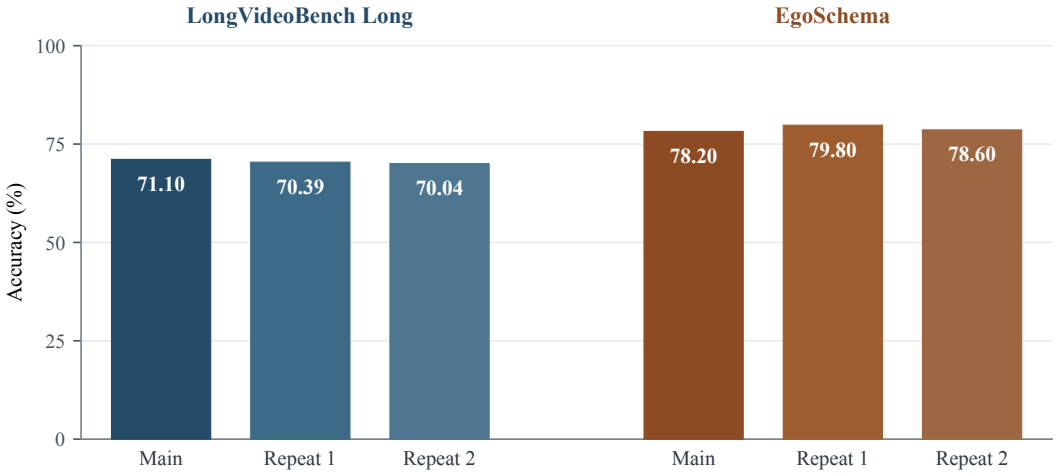


Figure 8: Run-to-run stability of final accuracy across the main Video-RSI evolution and two additional complete runs ($n = 3$). The two benchmark groups share a 0–100% accuracy axis. Bars show individual runs, with their accuracy values labeled inside.

Across the three complete evolutions, final accuracy ranges from 70.04% to 71.10% on LongVideoBench Long and from 78.20% to 79.80% on EgoSchema. The corresponding means are 70.51% and 78.87%, with sample standard deviations of 0.54 and 0.83 percentage points (sample variances of 0.292 and 0.693 pp^2). The close agreement among the three final harnesses shows that Video-RSI yields stable final performance across complete evolutions.

D Evolution Dynamics on Additional Benchmarks

We evaluated intermediate harnesses on the external benchmarks only after evolution had completed and the final harness was fixed. These retrospective evaluations did not inform candidate acceptance, stopping, or configuration selection. Figure 9 complements the MLVU trajectory in Figure 3 with the same retained stages on EgoSchema, Video-MME Long, and LongVideoBench Long val. The six markers correspond to H_{S0}–H_{S5} at attempt indices 0, 5, 7, 8, 14, and 17. Flat segments retain the incumbent harness between accepted revisions. Only the initial and final values are labeled, to one decimal place.

On all three benchmarks, the first accepted revision reduces mean visual-frame usage, and the final H_{S5} surpasses H_{S0} in accuracy while using fewer frames per question. The final values match the Video-RSI row of Table 1, providing the stage-wise counterpart to the main paper's cross-benchmark results.

E Sensitivity to Training-Set Size

The 216-question training set, with 36 questions from each of the six training-task categories, was fixed as the main configuration before external-benchmark evaluation. We subsequently added the 144- and 288-question runs to analyze sensitivity to training-set size. Their evaluation results were not used to choose the main configuration or the final main-run harness. Table 6 compares the final Solver harnesses on MLVU and EgoSchema, with the 216-question result matching the main run in Table 1.

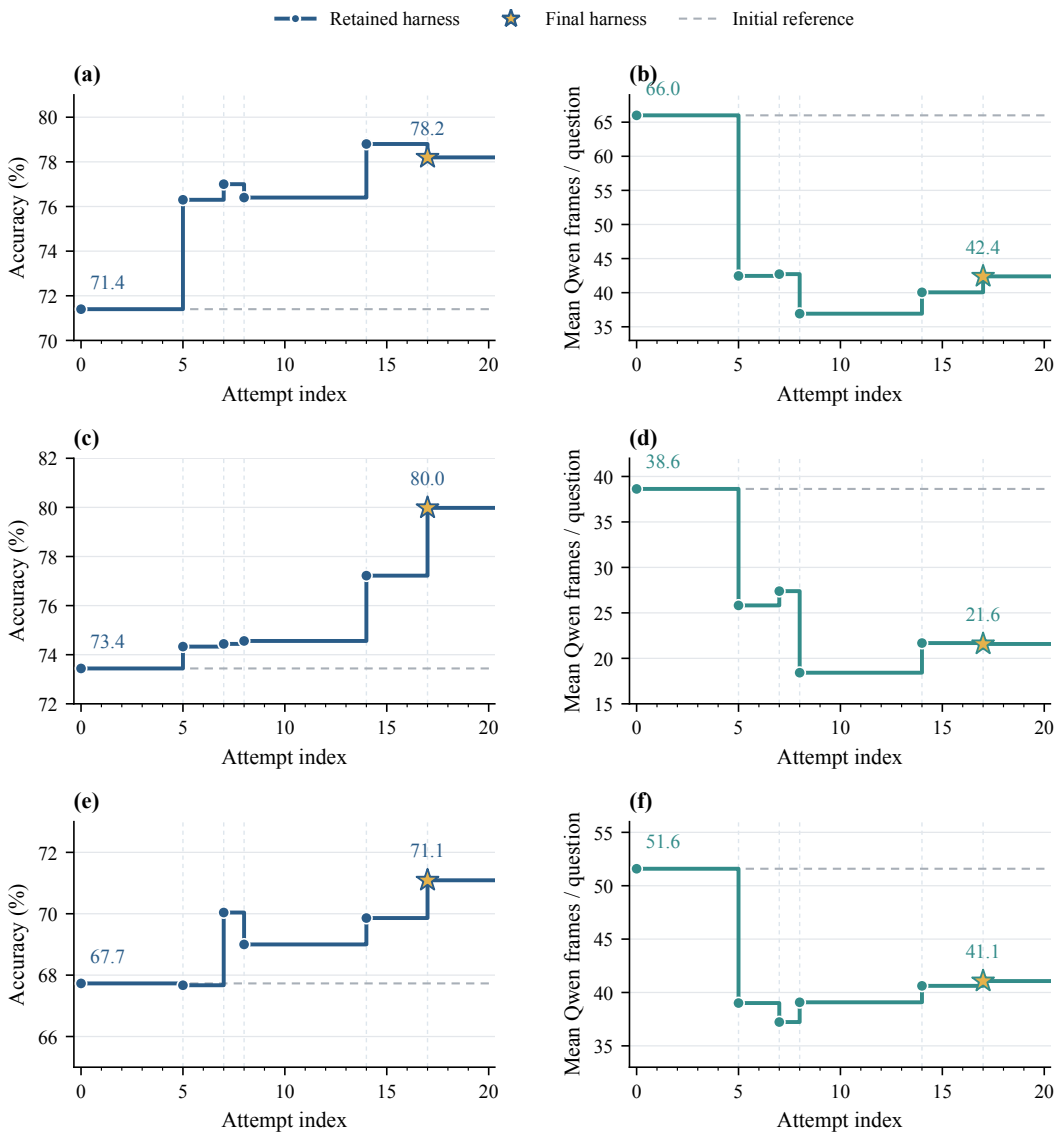


Figure 9: Stage-wise evaluation on the three non-MLVU benchmarks in Table 1, with EgoSchema in (a,b), Video-MME Long in (c,d), and LongVideoBench Long val in (e,f). Left panels show accuracy. Right panels show mean visual frames per question. Steps track the retained Solver harness over 20 revision attempts. Stars mark H_{S5}, and horizontal dashed lines mark H_{S0}.

Table 6: Training-set size and final evaluation on MLVU and EgoSchema. Acc. denotes accuracy (%), and Frames denotes mean visual frames per question.

Training questions	MLVU Test ($N = 502$)		EgoSchema	
	Acc. ↑	Frames ↓	Acc. ↑	Frames ↓
144	64.5	47.7	74.8	51.0
216	72.9	61.1	78.2	42.4
288	69.9	63.4	77.0	37.3

The 144-question run already transfers to both benchmarks, reaching 64.5% accuracy on MLVU and 74.8% on EgoSchema. The 288-question run reaches 69.9% and 77.0%, respectively. The 216-question main run attains the highest observed accuracy on both benchmarks, at 72.9% and 78.2%. The frame trade-off differs by benchmark. The 288-question run uses slightly more frames than the main run on MLVU (63.4 versus 61.1 per question), but fewer on EgoSchema (37.3 versus 42.4).

The 288-question training set includes 119 incorrect initial cases. These failures expand the pool available for active investigation while placing more demands on the fixed investigation budget because the Editor must select representative cases and consolidate their potentially different explanations into reusable harness revisions. In this retrospective comparison, the additional diagnostic breadth does not yield higher accuracy than the fixed 216-question main configuration.

F Baseline Reproduction and Implementation Details

We implement inference-time adaptations of VideoSeek and the released VideoHarness-RSI endpoint under the DeepSeek-V4-Pro reasoning model and Qwen3.6-Plus visual backend used in our evaluation. Both adaptations retain the baselines' video-navigation and evidence-organization policies while using the model stack of Video-RSI. The other baselines in Table 1 retain the results reported in their respective publications.

Table 7: Documented implementation settings of the re-evaluated baselines. Model IDs identify the requested API models. Visual budgets refer to individual operations, not a shared per-question frame cap.

Setting	VideoSeek	VideoHarness-RSI
Reasoning model ID	`deepseek-v4-pro`	`deepseek-v4-pro`
Visual model ID	`qwen3.6-plus`	`qwen3.6-plus`
Inference policy	Fixed released navigation policy	Released `CardinalityLedger` endpoint; no re-evolution
Operations	`overview`, `skim`, `focus`	CLIP retrieval, coarse-to-fine localization, burst inspection, counting ledger
Visual budget	$\alpha = 2$: 32 overview frames, 8 skim frames, up to 8 focus frames; 20 agent steps	Up to 320 CLIP candidate frames; up to 30 montages per localization call; 40-frame burst budget

VideoSeek. We start from the released VideoSeek implementation (Lin et al., 2026) and retain its `overview`, `skim`, and `focus` operations. DeepSeek-V4-Pro selects the next operation, the video interval to inspect, and the final answer. Qwen3.6-Plus supplies visual observations. Available timestamped subtitles enter through VideoSeek's subtitle interface. The policy is held fixed during benchmark inference.

VideoHarness-RSI. We use the released `CardinalityLedger` inference policy (Xu & Chen, 2026) as the VideoHarness-RSI endpoint, without rerunning its harness-search process. The policy builds a CLIP-indexed frame pool, localizes candidate intervals, and packs selected views into visual evidence. Qwen3.6-Plus processes this evidence, and DeepSeek-V4-Pro selects the final multiple-choice answer. Thus, the video-navigation policy is retained while the original single-VLM answering interface is adapted to our model stack.

Visual-use accounting. For our evaluations of Video-RSI, VideoSeek, and VideoHarness-RSI, we use a common visual-frame accounting rule. Frames denotes the total number of video-frame inputs processed across successful visual-model calls during a complete question-answering execution, averaged across questions. Repeated submission of the same frame counts on each successful call. Success denotes a returned provider response, not a correct answer. OCR inputs are recorded separately and excluded. Decoding, sampling, and other preprocessing do not add to this metric unless the resulting frames enter a successful visual call. Failed or pending requests are excluded. A successful retry contributes its input frames once. Reusing stored observations without a new call adds no frames, whereas resubmitting cached images counts normally. Reusing a saved prediction retains its original frame count. Offline Editor investigation is excluded.

MLVU score aggregation. Our MLVU evaluations report micro accuracy over the 502 multiple-choice test questions, giving equal weight to each question. Let n_t and a_t denote the number of questions and accuracy in task category t, respectively, with a_t expressed as a percentage. Micro accuracy and the official task-macro M-Avg use different aggregation rules:

$$A_{\text{micro}} = \frac{\sum_{t=1}^{9} n_t a_t}{\sum_{t=1}^{9} n_t}, \qquad A_{\text{M-Avg}} = \frac{1}{9} \sum_{t=1}^{9} a_t.$$

Video-RSI answers 366 of the 502 questions correctly, yielding 72.91% micro accuracy, displayed as 72.9% in Table 1.

Video-RSI implementation. The code of Video-RSI is provided in the Supplementary Material attached to this submission, which provides the harness implementations, Solver and Editor prompts, configuration templates, and execution entry points. The shared evolution framework coordinates training-side execution, active investigation, candidate revision, and development-set evaluation. The Editor records a modification plan before editing the candidate. Candidates undergo execution checks before paired evaluation against the current harness on the fixed development set. The framework then accepts the candidate or retains the current harness and returns aggregate evaluation feedback to the Editor.

Evolution configuration. Table 8 summarizes the default settings provided in the code package. The Editor and its investigation calls share a provider-call budget across analysis, planning, and editing within each revision attempt. Investigation frames are tracked separately under a cumulative frame limit. Execution failures preserve the current harness and the recorded attempt state.

Table 8: Default evolution and model-call settings provided in the accompanying Video-RSI code package.

Setting	Value
Shared Editor/investigation provider-call limit per attempt	160
Investigation frame limit per attempt	1,024
Editor time budget per attempt	3,600 s
Maximum output tokens per model call	8,192