VIDEOHARNESS-SRSI: STRUCTURED RECURSIVE SELF-IMPROVEMENT FOR VIDEO UNDERSTANDING HARNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM system capabilities depend not only on model parameters but also on the executable *harnesses* that organize information acquisition, tool use, reasoning, and memory. Recursive harness evolution can automate their design, but how it can reliably discover generalizable systems from limited, noisy, and repeatedly reused feedback remains unclear. We study this question in agentic video understanding, where rich multimodal evidence and coupled mechanisms make harness optimization challenging. We first introduce VIDEOHARNESS-BRSI, a recursive self-improvement framework that evolves video harnesses around a frozen vision–language model. Analyzing over 4,000 evolved candidates, we find it competitive with hand-engineered systems but identify three interconnected failure modes: *validation overfitting*, *search-budget dilution*, and the *early-experience trap*. We trace all three to a common root: the same imperfect validation evidence drives both harness revisions and the proposer's search strategy. We therefore propose VIDEOHARNESS-SRSI, an evidence-governed framework in which every search decision rests on evidence that is transferable, sufficient, and gathered before the search commits. It restricts the proposer to content-free feedback, informs exploration with instrumented warm-up experiments, and develops mechanisms in isolated probe–race branches before factorial composition and joint refinement. Across four video-understanding benchmarks, VIDEOHARNESS-SRSI improves the equally weighted mean held-out score by 6.2% relative to VIDEOHARNESS-BRSI and reduces the mean validation–test gap by 37.9%. Reliable harness evolution thus depends on how search is organized and informed, not only on how well it optimizes validation performance.

1 INTRODUCTION

An agentic system is shaped not only by its foundation model but by its *harness*: the executable code around the frozen model that decides what the model observes, which tools run, and how intermediate outputs become an answer (Ning et al., 2026). A well-designed harness can substantially improve performance (Lee et al., 2026), but designing one is hard, since each mechanism admits many implementations whose effectiveness hinges on other components. Automated agent design and recursive harness evolution replace manual engineering with an LLM proposer that iteratively generates, evaluates, and refines harnesses from accumulated feedback (Hu et al., 2025; Shang et al., 2025; Lee et al., 2026). Yet, as harnesses become more complex, a fundamental question arises: *How can recursive evolution reliably discover generalizable designs when experimental evidence is limited, noisy, and repeatedly reused, and the search space contains many interacting mechanisms?*

We study this question through *agentic video understanding*, a particularly demanding setting for harness optimization where a video harness must organize extensive, heterogeneous visual evidence via frame selection, retrieval, perception tools, multi-turn reasoning, reflection, memory, and multi-agent collaboration. These mechanisms are interdependent—retrieving more frames may recover missing evidence but also add distractors—so no mechanism can be judged in isolation. Moreover, evolution must infer useful changes from repeated evaluations on finite video–question pairs, whose rich content invites the proposer to specialize to seen examples. Video thus couples two challenges:

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.

learning reliable lessons from complex feedback, and efficiently exploring a combinatorial space of interacting mechanisms (Wang et al., 2025; Long et al., 2025a; Liang et al., 2025).

To study these challenges, we first build VIDEOHARNESS-BRSI (Baseline Recursive Self-Improvement), in which a multimodal LLM coding proposer iteratively revises the executable harness around a frozen vision–language model, guided by accumulated multimodal evidence and execution feedback. Evolution discovers competitive harnesses: for example, in a Video-MME (Fu et al., 2025)run evolving only retrieval, the evolved harness matches the held-out accuracy (68.4%) of a manually designed Adaptive Keyframe Sampling (AKS) variant (Tang et al., 2025) with 38.7% fewer frames. Yet across 4,000+ evolved candidates, validation success does not reliably translate into generalization. We trace this gap to a single source: the loop learns only from *validation verdicts*—score changes on a finite, repeatedly reused validation set—which mix transferable improvements with instance-specific fit and noise, yet decide both which harness is kept and where the proposer searches next. This evidence misleads evolution in three ways. (1) *Validation Overfitting* concerns what a verdict measures: instance-level multimodal feedback lets the proposer fit observed examples, so verdicts reward non-transferable gains. (2) *Search-Budget Dilution* concerns how much evidence a verdict rests on: joint evolution spreads a limited budget over many mechanism combinations, and coupled edits change many answers at once, so each configuration is evaluated too rarely and noisily to attribute its gains. (3) *Early-Experience Trap* concerns how verdicts are acted on: a single noisy early outcome persists as a judgment in the proposer's memory, so the proposer abandons useful mechanisms prematurely and over-explores early successes.

Motivated by these findings, we further propose VIDEOHARNESS-SRSI (Structured Recursive Self-Improvement), an evidence-governed framework built on one principle: every search decision should rest on evidence that reflects transferable quality, suffices for the decision, and is gathered before search commits. The proposer sees only aggregate scores and *content-free execution statistics*, never the validation videos, questions, or answers, letting it diagnose harness behavior without tailoring revisions to seen data. Evolution then proceeds in stages: a *warm-up* complements free proposals with *controlled experiments* that ablate a mechanism or vary its intensity, measuring its contribution while keeping under-tested mechanisms open. This evidence seeds *mechanism-specific branches*, each evolved in isolation from a shared starting harness under a *probe–race* budget that tries every mechanism before focusing on the most promising. Finally, *factorial composition* tests whether branch improvements transfer and how they interact, before *joint refinement* co-optimizes them. Each stage thus supplies the evidence the next acts on.

We evaluate on four video question answering and temporal grounding benchmarks: Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), OVO-Bench (Niu et al., 2025), and MomentSeeker (Yuan et al., 2026). VIDEOHARNESS-SRSI yields stronger, better-generalizing harnesses than VIDEOHARNESS-BRSI, raising the equally weighted mean held-out score by 6.2% and cutting the mean validation–test gap by 37.9%; ablations validate its design. These findings suggest that the organization and evidence quality of evolution matter beyond validation optimization.

Our main contributions are:

- **Diagnosis.** An empirical investigation of recursive harness evolution in agentic video understanding shows that it discovers competitive harnesses and characterizes three failure modes rooted in shared imperfect validation evidence.
- **Framework.** We propose VIDEOHARNESS-SRSI, an evidence-governed framework combining content-free proposer feedback, instrumented warm-up experiments, and isolated mechanism development before joint refinement.
- **Evaluation.** On four benchmarks, VIDEOHARNESS-SRSI improves mean held-out score by 6.2% and cuts the mean validation–test gap by 37.9% over VIDEOHARNESS-BRSI.

2 INVESTIGATION OF VIDEO UNDERSTANDING HARNESS EVOLUTION

Building on prior work in recursive harness self-improvement and hand-designed video understanding harnesses, we first introduce VIDEOHARNESS-BRSI, a baseline recursive self-improvement framework for agentic video understanding, to understand how harness evolution behaves in this more complex setting. As shown in Figure 1(a), an agentic coding proposer P evolves executable harnesses around a frozen VLM M and directly accesses the multimodal evidence be-

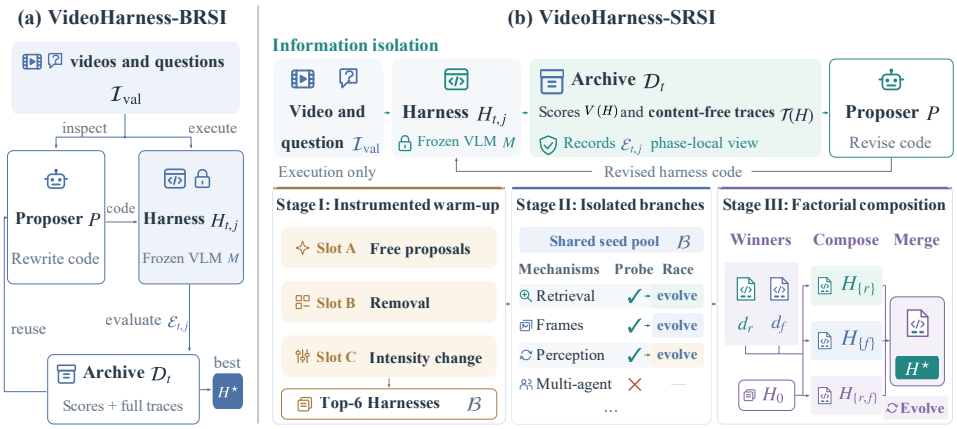


Figure 1: The illustration of VIDEOHARNESS-BRSI and VIDEOHARNESS-SRSI.

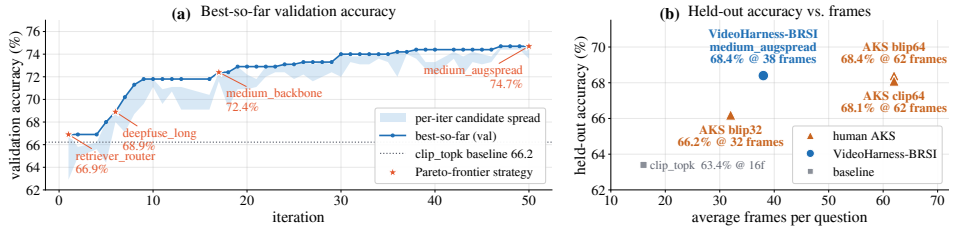


Figure 2: Performance of VIDEOHARNESS-BRSI on Video-MME.

hind each outcome: the validation videos and questions $\mathcal{I}_{\text{val}}$, prior implementations, and execution traces. This full access deliberately extends the design of Meta-Harness (Lee et al., 2026) to video understanding. Starting from an archive $\mathcal{D}_0$ of seed harnesses, iteration t proposes candidates $H_{t,j} \sim P(\cdot \mid \mathcal{D}_{t-1}, \mathcal{I}_{\text{val}})$, evaluates each on the validation set $\mathcal{S}_{\text{val}}$ to obtain a record $\mathcal{E}_{t,j}$ of scores, frame counts, and traces, and adds $(H_{t,j}, \mathcal{E}_{t,j})$ to the archive $\mathcal{D}_t$. Unless otherwise specified, M is Qwen3-VL-8B-Instruct (Bai et al., 2025) and P is Claude Opus 5 (Anthropic, 2026). Related work is reviewed in Appendix A, and VIDEOHARNESS-BRSI is specified in Appendix C.

2.1 CAN VIDEOHARNESS-BRSI EVOLVE EFFECTIVE VIDEO HARNESSES?

We first examine whether recursive evolution can discover harness designs competitive with expert-engineered alternatives. A fair test needs a strong hand-designed reference for the same mechanism, so we evolve only the *retrieval strategy* and compare with Adaptive Keyframe Sampling (AKS) (Tang et al., 2025), a state-of-the-art retrieval method; evolved and manual harnesses then differ only in how that mechanism is designed. We run on Video-MME (Fu et al., 2025), a widely used benchmark on which AKS was evaluated. We use 450 questions for validation and the remaining 2,250 for held-out evaluation.

Finding 1: Recursive evolution discovers harnesses competitive with manual designs. As shown in Figure 2, starting from the `clip_topk` seed, VIDEOHARNESS-BRSI evolves the `medium_augspread` harness, reaching 68.4% held-out accuracy with 38 frames on average, matching the strongest AKS variant based on 62 frames. Additional experiments demonstrate that recursive evolution also improves other mechanism families and generalizes across different benchmark datasets, establishing that automated harness evolution can discover effective designs without relying exclusively on expert-crafted strategies. The setting, the seed harnesses (Table 8), and these additional experiments are detailed in Appendix D.

2.2 DIAGNOSING VIDEOHARNESS-BRSI IN VIDEO UNDERSTANDING

Despite these gains, VIDEOHARNESS-BRSI shows three systematic failure modes, as shown in Figure 3. Since our baseline shares the core logic of prior harness evolution—propose new candidates from past ones and their scores, then keep those that win on validation (Hu et al., 2025; Zhang et al., 2025b; Lee et al., 2026)—these failures expose limitations of this general strategy rather than

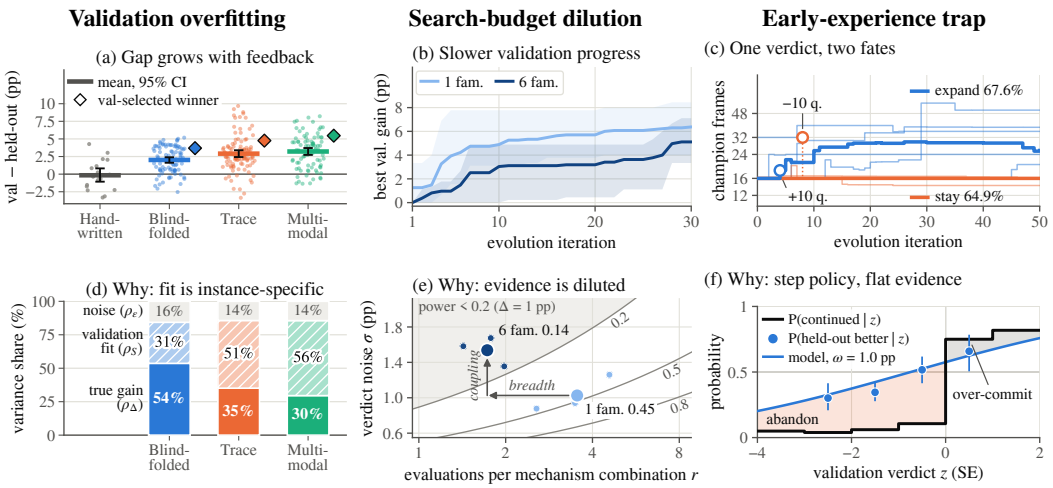


Figure 3: **Three failure modes of VIDEOHARNESS-BRSI: what happens (top) and why (bottom).** Dots in (a) are evolved harnesses; bands in (b) span runs. (c) Thick lines: the two runs in the text, with their frame-raising probes circled; percentages: mean held-out accuracy. (e) Small/large dots: runs/space means; contours: power to detect a 1-point gain.

of our baseline. We further theoretically explain them through a common signal the evolution loop learns from, the *validation verdict* $\hat{\delta} = V(H) - V(H^{\text{inc}})$: a candidate's validation accuracy minus that of the current champion H^{inc}, which decides both what is kept and where the proposer searches next. Since the goal is to improve the accuracy on unseen questions, evolution works only as far as $\hat{\delta}$ tracks the true gain. We therefore split each validation score as $V(H) = \mu(H) + S(H) + e(H)$: the population accuracy $\mu(H)$ on new questions, a fit $S(H)$ specific to the validation questions (zero for harnesses written without seeing them), and zero-mean sampling noise $e(H)$. Subtracting the champion's split, and noting that the champion's noise is biased upward by $\omega > 0$ because it is a never re-evaluated maximum of noisy scores (Smith & Winkler, 2006), gives

$$\hat{\delta} = \underbrace{\mu(H) - \mu(H^{\text{inc}})}_{\Delta\text{: true gain}} + \underbrace{S(H) - S(H^{\text{inc}})}_{S\text{: validation fit}} - \underbrace{\omega}_{\text{champion inflation}} + \underbrace{\varepsilon}_{\text{noise}}. \tag{1}$$

Here only Δ transfers to held-out data; ε collects the remaining zero-mean noise, whose standard deviation σ grows with the number of answers the candidate changes. Each failure mode lets a different term overwhelm Δ: validation overfitting inflates S (Section 2.2.1), search-budget dilution inflates ε (Section 2.2.2), and the early-experience trap lets ε and ω decide irreversible choices (Section 2.2.3). Appendix (E derives this model.

2.2.1 VALIDATION OVERFITTING

We start with the validation fit S in Eq. (1). A harness can fit the validation questions only through what the proposer learns about them, and VIDEOHARNESS-BRSI, following Meta-Harness, allows the proposer to directly inspect validation videos, questions, and execution traces. If this access drives the fit, the gap between validation and held-out accuracy should grow with how much the proposer observes. We therefore keep the evolution loop fixed and peel back this access in three nested arms: *Blindfolded* sees outcome-level information only, i.e., each question's correctness and behavioral counters such as frames and tool calls; *Trace* adds question text, options, and interaction traces; *Multimodal* adds video frames, recovering VIDEOHARNESS-BRSI's full access. We use LongVideoBench (Wu et al., 2024), whose questions refer to specific moments in long videos, so content feedback reveals much about each instance. Each arm runs three seeds at each of three validation fractions (33%, 50%, 75%), so the effect does not hinge on one split size.

Finding 2: Richer feedback widens the validation–test gap by letting harnesses fit the validation questions. We call validation minus held-out accuracy a harness's *gap*. As shown in Figure 3(a), hand-written baselines show no gap because they never see validation data, whereas evolved harnesses do, and richer feedback widens it: from 2.0 (Blindfolded) to 3.2 points (Multimodal) on average. To tell how much of the validation progress comes from the true gain Δ rather than the validation instance-specific fit S, we analyze all evolved harnesses of a run together. These harnesses

reach different validation scores, and by Eq. (1) the differences among them have three sources: Δ, S, and ε (ω is common to all of them, since they face the same champion). On held-out data, whose questions are new, only the differences due to Δ remain. We measure what fraction of the spread in validation scores each source accounts for, giving the true-gain share ρ_Δ, the validation-fit share ρ_S, and the noise share ρ_ε, all estimated from observed validation and held-out scores. As shown in Figure 3(d), ρ_Δ falls from 54% (Blindfolded) to 35% (Trace) and 30% (Multimodal), while ρ_ε stays at 14–16%, so the loss goes entirely to ρ_S, which rises from 31% to 56%: richer feedback does not make validation noisier, but makes it reward fitting the questions rather than real improvement. This fit traces to what the proposer learns about the questions: proposers exposed to question text copy validation questions into their code more often than chance, whereas Blindfolded proposers do not. *Takeaway: the proposer should observe how a harness behaves, not the instances on which it is evaluated.* The protocol, the estimators of the three shares, the supporting theory, and robustness checks are in Appendix F.

2.2.2 Search-Budget Dilution

We next turn to the noise ε. Video harnesses combine mechanisms with distinct roles: frame selection decides which moments the model sees (Tang et al., 2025), retrieval gathers question-relevant evidence (Luo et al., 2025), tools extend perception (Wang et al., 2024; Zhang et al., 2026b), and memory carries information across long videos (Yeo et al., 2026; Long et al., 2025b). Searching over more of these mechanisms enlarges the search space, yielding potentially higher accuracy. But a larger space also spreads a fixed evaluation budget over more candidates, whereas a candidate's true gain can be told from ε only after enough verdicts accumulate to average the noise out. Whether the added expressiveness or the thinner evidence dominates is thus an empirical question. We keep everything else fixed and vary only the search space, comparing on Video-MME at a matched budget of 87 candidates, with three seeds each, the smallest space, retrieval only (one mechanism family, as in Section 2.1), with the full harness (six families).

Finding 3: Larger search spaces leave too little evidence to tell real gains from noise, slowing evolution without improving held-out accuracy. As shown in Figure 3(b), retrieval alone gains 6.4 validation points after 87 candidates, whereas the full harness gains only 5.1 and lags for most of the run. Its deployed champions are also no better on held-out data, so the added expressiveness goes unused. To see why, note that a single verdict is noisy, so the loop can trust that a mechanism combination truly helps only after several candidates using it have been evaluated on the validation set. We therefore measure the *power* of each evolution run, the chance of detecting a real gain of 1 point, which is larger than most improvements the loop accepts. Power depends on two quantities: the number of validation evaluations r per mechanism combination, which averages the noise out, and the noise σ of each verdict, which hides the gain. A larger space can lower power through two channels: *breadth*, spreading the same budget over more mechanism combinations, which lowers r; and *coupling*, letting mechanisms feed into one another, so that even a local edit changes more answers and, since a verdict is decided only by the answers that change, raises σ. As shown in Figure 3(e), where each evolution run is placed by its r, and σ and the curves mark equal power, both channels are at work: from one family to six, r falls from 3.5 to 1.7, moving the runs left, and σ rises from 1.0 to 1.5 points, moving them up. Together they push the full harness into the shaded region of power below 0.2, which means the full-harness search misses most real 1-point improvements. Coupling is also visible directly: edits in the full harness change more answers than comparable edits in retrieval-only search, and larger edits are more often harmful under the same evaluation budgets. *Takeaway: mechanisms should be developed where evaluations concentrate and edits stay local, before their interactions are optimized.* Intermediate spaces with two and three families, the power analysis, and edit footprints are in Appendix G.

2.2.3 The Early-Experience Trap

The first two failure modes concern whether a verdict is right; this one concerns what the evolution loop does with it. In VideoHarness-BRSI, a verdict decides not only which candidate is kept but also which *direction*, a kind of change such as raising the frame budget or adding re-ranking, the proposer pursues next, so early verdicts shape the rest of a run. For each verdict, we first record whether later proposals build on the candidate (*continuation*), which shows how the loop reacts. We then compare the candidate with the champion on held-out data never shown to the proposer, which shows whether the verdict was right. We analyze this on 14 Video-MME runs, 12 of which evolve retrieval and form the frame-budget contrast below.

Finding 4: A single noisy early verdict can fix a run's course, as weak verdicts drive irreversible decisions. Figure 3(c) traces each run's champion frame count: the same kind of probe can meet opposite verdicts. In two runs of the same setting (thick lines), a probe raising the frame budget (circled) wins by ten validation questions in one run and loses by ten in the other. The first run then expands; the second records frame count as "MINED OUT" in its memory, and its champion never exceeds 16 frames. This single verdict matters: across 12 runs, champions that expand reach 67.6% held-out accuracy, versus 64.9% for those staying near 16 frames. Eq. (1) explains why such a verdict misleads: a direction's first verdict is $\Delta - \omega + \varepsilon$, and both error terms are large, since most verdict variation is noise and the inflated champion puts every new direction about $\omega \approx 1$ point behind. Weighing such weak evidence properly, a loop would continue a direction with probability rising smoothly with its verdict. Figure 3(f) plots, against the verdict z (in standard errors), how often later proposals build on a candidate (black) and how often it truly beats the champion on held-out data (blue). The loop instead acts as a step function, continuing about 77% of winning candidates but only 8% of the rest, whereas the held-out win rate rises gradually, already reaching one half for slightly negative verdicts. The shaded gap marks likely-better candidates that were abandoned. Because a rejected direction is rarely retried, an unlucky first verdict is never corrected. Our model of the resulting chance of losing a beneficial direction matches, for the frame budget, the half of runs in Figure 3(c) that never expand. *Takeaway: early evidence should come from controlled comparisons collected regardless of earlier verdicts, and untested directions should remain open.* The verdict model, the persistence analysis, a replication on LongVideoBench, and case studies are in App. H.

Implications. The three failure modes share one root, imperfect evidence that steers both the harness and the search, and reinforce one another: a misleading early win narrows the search, and a broad space leaves little budget to revisit it. Standard evolutionary levers, such as clade-UCB parent selection and initialization from AKS, change where the next verdict is taken but not what it measures, and do not improve held-out accuracy. (App. I) The three takeaways instead act on the evidence itself and form the design principles of VIDEOHARNESS-SRSI (Section 3)..

3 VIDEOHARNESS-SRSI

Section 2.2 traced all three failure modes to the validation verdict that VIDEOHARNESS-BRSI learns from: rich feedback lets it reward fitting the validation questions (S), a large search space leaves it too noisy to separate real gains (ε), and noisy early verdicts close directions for good (ε and ω). VIDEOHARNESS-SRSI therefore keeps the propose–evaluate–accumulate loop of VIDEOHARNESS-BRSI and changes only the evidence behind its decisions (Figure 1(b)). Each component implements one takeaway: *information isolation* limits what the proposer can fit (Section 3.1), and *three-stage evolution* concentrates evidence on one mechanism at a time (Section 3.2), beginning with an *instrumented warm-up* that keeps early directions open (Section 3.2.1). The full algorithm (Alg. 2) and more details are in App. J.

3.1 INFORMATION ISOLATION

Finding 2 demonstrates that the validation fit grows with what the proposer observes about the validation instances, and that the Blindfolded arm kept the largest transferable share of its progress (54%) while deploying harnesses at least as accurate on held-out data as the richer arms. Since a harness can fit the validation questions only through what the proposer learns about them, we close this channel at its source instead of penalizing edits afterwards, when the information has already been used. In VIDEOHARNESS-SRSI, the proposer never accesses the validation videos, questions, options, or answers; only the harness it writes does, at execution time. The proposer receives each candidate's validation accuracy and a *content-free trace* $\mathcal{T}(H)$: per question, its correctness under an opaque identifier plus behavioral counters such as model and tool calls, frames, retrieval, and memory operations. This is exactly the Blindfolded feedback, so isolation lowers the fit without starving the search. A deny list on all proposer tools enforces it; details are in Appendix J.1.

3.2 THREE-STAGE EVOLUTION

Finding 3 shows that searching the full harness jointly loses power in two ways: *breadth* spreads the budget over many mechanism combinations, and *coupling* lets each edit change many answers.

Restricting the search to a single mechanism would avoid both, but forfeits the expressiveness of the full space. We instead *order* the search: first collect controlled evidence about every mechanism (Stage I), then improve each mechanism separately, where evaluations concentrate and edits stay local (Stage II), and only then optimize their interactions among a few mechanisms whose effects are already measured (Stage III). The search thus keeps the full space while each decision gets the evidence of a small one.

3.2.1 STAGE I: INSTRUMENTED WARM-UP

Finding 4 reveals that VIDEOHARNESS-BRSI gathers evidence only where earlier verdicts were favorable, so one unlucky verdict can close a beneficial direction for the rest of a run. The warm-up runs the ordinary loop but reserves part of each round for *controlled interventions* that are collected regardless of earlier verdicts. Besides three free proposals, each round includes one *removal*, which deletes one of the harness's seven mechanism families, and one *intensity* change, which substantially changes how much a mechanism is used, such as the frame budget of Finding 4. For each warm-up candidate H with parent $p(H)$, we record its verdict and its change in mechanism usage,

$$\mathbf{w}(H) = \big(\hat{\delta}(H, p(H)),\ \mathbf{u}(H) - \mathbf{u}(p(H))\big), \tag{2}$$

where $\mathbf{u}(H)$ summarizes runtime mechanism usage from $\mathcal{T}(H)$; slot requests are audited from this usage change rather than trusted. Evidence on each mechanism thus accumulates over many comparisons instead of resting on its first verdict, and because all removals are judged the same way, the champion's inflation shifts them alike and does not distort how mechanisms compare. The warm-up summarizes this evidence in a per-mechanism *warm-up report*; mechanisms without enough evidence are flagged as under-explored rather than discarded. When accuracy plateaus, the six best warm-up harnesses form a seed pool $\mathcal{B}$, and the best of them becomes the *backbone* H_0 shared by all later stages, so later stages draw on several strong lineages rather than a single, possibly lucky, champion (ablation in Section 4.2). The mechanism families, slot audit, stopping rule, and warm-up report are detailed in Appendix J.2.

3.2.2 STAGE II: ISOLATED BRANCH EVOLUTION

To remove coupling, Stage II splits the harness into six mechanism axes—*retrieval*, *frame budget*, *read schedule*, *multi-role*, *pixels and detectors*, and *derived state*—plus a free-form axis that guards against a misspecified split, giving the axis set $\mathcal{A} = \{\alpha_1, \ldots, \alpha_6, \alpha_{\text{free}}\}$. Each axis $\alpha \in \mathcal{A}$ evolves from H_0 in its own isolated branch, and a fixed template states which parts of the harness the branch may edit and a test that every edit must pass (e.g., only frame-budget edits may change the number of frames). Edits therefore stay local, and a branch's gains are attributable to its mechanism. Templates are fixed in advance, so the choice of axes cannot itself fit the validation set; axes defined after the warm-up generalize worse (Section 4.2). From the warm-up report, the proposer writes each axis a *charter* c_α: the mechanism to try, a first edit, and a criterion that would falsify it.

To counter breadth, a *probe–race* schedule concentrates evaluations without leaving any axis untried: every axis is probed with three candidates, the top half by best probe verdict against H_0 race on to fifteen, and at most three raced axes, those with a clearly positive mean verdict first, proceed to composition. Since all axes share the backbone H_0, the champion's inflation shifts them alike and leaves their ranking intact. Mopre details are detailed in Appendix J.3.

3.2.3 STAGE III: FACTORIAL COMPOSITION AND JOINT REFINEMENT

Gains found in separate branches need not add up. Greedy merging cannot tell whether they do. The proposer therefore distills each selected axis $\alpha \in \mathcal{A}_{\text{comp}}$ into a fixed, portable modification d_α, the smallest edit of H_0 carrying the mechanism. We evaluate every combination as a factorial design:

$$H_{\mathcal{U}} = \text{Apply}\big(H_0, \{d_\alpha : \alpha \in \mathcal{U}\}\big), \quad \mathcal{U} \subseteq \mathcal{A}_{\text{comp}}, \quad H_\emptyset = H_0. \tag{3}$$

With three axes this costs seven evaluations, yet each modification's effect is averaged over four comparisons rather than resting on one, and the same comparisons measure interactions. Joint refinement then starts from all combinations and may modify the selected mechanisms together: coupling returns, but only among at most three mechanisms whose effects are already measured. The final harness $H^\star$ is the validation-best one after refinement; this last selection on validation reintroduces some fit and champion inflation, a source of the residual gaps in Section 4.1. The composition lattice, interaction estimates, and refinement are detailed in Appendix J.4.

Table 1: **Main results.** Evolved harnesses report mean ± sample standard deviation over three seeds; fixed baselines are deterministic.

Method	Video-MME		LVB		OVO-Bench		MomentSeeker	
	Val	Test	Val	Test	Val	Test	Val	Test
Fixed Baselines								
uniform	60.44	61.64	58.97	59.15	52.76	52.20	11.94	11.43
clip_topk	66.21	63.40	62.33	62.07	51.94	53.32	16.58	17.98
tool_survey	65.78	62.27	59.42	60.49	50.51	51.28	16.72	17.66
reflect_loop	60.44	60.04	62.11	60.61	51.33	55.36	16.23	17.66
debate_pair	63.78	61.42	57.40	55.56	44.58	45.86	16.42	17.92
memo_multiturn	63.11	58.98	58.52	59.93	55.42	54.65	15.91	16.00
Evolved Harnesses								
VIDEOHARNESS-BRSI	$72.2_{\pm 1.8}$	$66.5_{\pm 1.2}$	$70.2_{\pm 0.3}$	$66.6_{\pm 1.0}$	$72.9_{\pm 2.1}$	$67.2_{\pm 1.7}$	$23.8_{\pm 0.7}$	$19.0_{\pm 1.8}$
VIDEOHARNESS-SRSI	$75.0_{\pm 1.9}$	$70.1_{\pm 1.0}$	$71.1_{\pm 2.0}$	$67.3_{\pm 1.0}$	$73.8_{\pm 2.4}$	$72.5_{\pm 3.1}$	$25.3_{\pm 0.3}$	$23.0_{\pm 0.7}$

4 EXPERIMENTS

Experimental setting. We evaluate VIDEOHARNESS-SRSI on four benchmarks covering general, long-form, and streaming video question answering and temporal grounding: Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), and OVO-Bench (Niu et al., 2025), scored by accuracy, and MomentSeeker (Yuan et al., 2026), scored by mIoU. Each benchmark is split once, at random, into a validation set, the only data evolution sees, and a held-out test set. All experiments use Qwen3-VL-8B-Instruct (Bai et al., 2025) as the frozen executor and Claude Opus 5 (Anthropic, 2026) as the proposer, with three seeds per benchmark. Full details are in Appendix B.

4.1 MAIN RESULTS

As shown in Table 1, VIDEOHARNESS-SRSI yields stronger and better-generalizing harnesses than VIDEOHARNESS-BRSI. Both evolved frameworks beat every fixed baseline on held-out data on all four benchmarks, exceeding the strongest baseline by 5.12 (VIDEOHARNESS-BRSI) and 8.52 points (VIDEOHARNESS-SRSI) on average. VIDEOHARNESS-SRSI raises the mean test score from 54.83 to 58.23 and lowers the mean validation–test gap from 4.95 to 3.08, with clear held-out gains and smaller gaps on Video-MME, OVO-Bench, and MomentSeeker. On OVO-Bench, test accuracy rises from 67.2 to 72.5 while validation accuracy rises only from 72.9 to 73.8: the gain comes mostly from better transfer rather than a stronger validation fit. Information isolation and structured evolution mitigate but do not eliminate overfitting because repeated selection on the validation set can still favor harnesses specialized to it.

Table 2: Held-out score vs. frame budget.

	Frames/q	BRSI	SRSI	Δ
VM	≤55.1	66.50	**68.93**	+2.43
LVB	≤69.2	66.60	**67.19**	+0.59
OVO	≤27.3	67.20	**68.42**	+1.22
MS	≤60.3	19.00	**21.41**	+2.41

Is the gain bought with a larger inference budget? Evolution may raise accuracy simply by letting the executor look at more frames. The deployed VIDEOHARNESS-SRSI harnesses do spend more on three benchmarks, e.g., 106.1 versus 55.1 frames per question on Video-MME, while on LongVideoBench they use fewer (64.8 vs. 69.2). To separate a better harness from a larger budget, we cap each VIDEOHARNESS-SRSI run at the mean frames per question of the deployed VIDEOHARNESS-BRSI harnesses, select among the candidates it evaluated under this cap by validation accuracy alone, and report the held-out score. As shown in Table 2, VIDEOHARNESS-SRSI still beats VIDEOHARNESS-BRSI on all four benchmarks, by 1.66 points on average, with a comparable number of executor calls. About half of the gain in Table 1 thus survives at equal budget: VIDEOHARNESS-SRSI finds more accurate harnesses at the same cost, not only more expensive ones. Per-seed costs and the full cost–accuracy frontier are in Appendix N.

4.2 ABLATIONS

Ablation of the instrumented warm-up. On Video-MME with three seeds, we compare the full warm-up of VIDEOHARNESS-SRSI with a *free-only* variant that spends the same five-candidate budget per round on free proposals. As shown in Figure 4, instrumented warm-up interventions improve generalization without changing validation fit. Both reach nearly the same warm-up validation accuracy, yet the full warm-up transfers better to the held-out split (69.0 vs. 67.1), and the

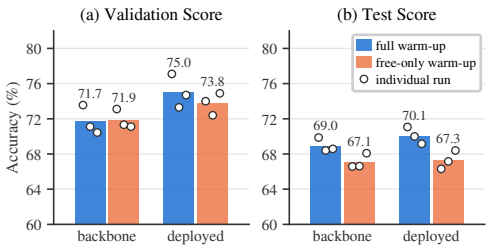


Figure 4: Ablation of the warm-up stage.

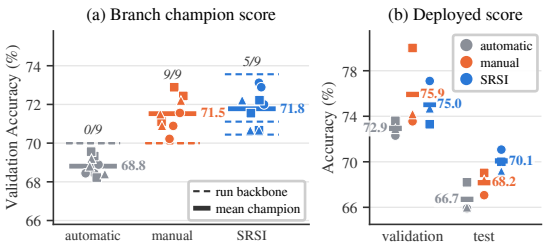


Figure 5: Ablation of axis selection.

advantage persists to the final system (70.1 vs. 67.3). The removal and intensity slots thus supply more informative evidence for later search rather than a stronger warm-up champion.

Ablation: who should choose the axes? On Video-MME with three seeds, we compare three ways to define three branch axes: *automatic*, where one proposer call defines axes and charters from the warm-up report; *manual*, where a researcher defines them from the same report; and the templates of VIDEOHARNESS-SRSI with probe–race allocation (Section 3.2.2). All variants share warm-up outputs, backbones, and downstream stages. As shown in Figure 5, the fixed templates of VIDEOHARNESS-SRSI generalize best. None of the nine automatic branches improves on its starting backbone, whereas all nine manual ones do; VIDEOHARNESS-SRSI reaches slightly higher mean branch accuracy, with five of nine branches improving on stronger backbones. The manual variant scores highest on validation (75.9 vs. 75.0) but lower on held-out data (68.2 vs. 70.1). Automatic decomposition misses productive directions and researcher-written charters overfit to validation, whereas VIDEOHARNESS-SRSI achieves the strongest held-out performance.

Ablation of the proposer model. We compare different proposer models, keeping the executor and every other setting fixed. As shown in Table 3, VIDEOHARNESS-SRSI remains effective with weaker proposers: all three exceed every fixed baseline on held-out data (best: 63.4), and with Claude Sonnet 5 it still surpasses VIDEOHARNESS-BRSI driven by Claude Opus 5 (67.7 vs. 66.5). Held-out accuracy nonetheless tracks proposer capability, suggesting that VIDEOHARNESS-SRSI will benefit from future proposers.

Table 3: VIDEOHARNESS-SRSI with different proposers on Video-MME.

Proposer	Val	Test
Opus 5	$75.0_{\pm 1.9}$	$70.1_{\pm 1.0}$
Sonnet 5	$73.3_{\pm 0.6}$	$67.7_{\pm 0.4}$
Opus 4.8	$71.4_{\pm 0.8}$	$65.8_{\pm 0.1}$

Does VIDEOHARNESS-SRSI mitigate the failure modes? We re-measure the three diagnostics of Section 2.2 on the runs of Table 1. *Validation overfitting* is already reflected in Table 1, where VIDEOHARNESS-SRSI narrows the validation–test gap on three of the four benchmarks; on LongVideoBench, information isolation alone raises the held-out accuracy of VIDEOHARNESS-BRSI from 65.4 to 66.9% at equal validation accuracy, close to VIDEOHARNESS-SRSI's 67.3%, and this gain holds on held-out questions whose videos never appear in validation. For the *early-experience trap*, the warm-up tests the frame budget in every run, so no VIDEOHARNESS-SRSI run stays at the seeds' budget, whereas four of the twelve main-result VIDEOHARNESS-BRSI runs do. For *search-budget dilution*, the probe–race concentrates the branch budget on a few axes, so each axis decision on Video-MME rests on 9.80 evaluations instead of 1.55, raising the power to detect a 1-point gain from 0.13 to 0.50 (Table 4). Protocols and per-run results of these ablations and diagnostics are in Appendices K–M and O.

Table 4: Evidence per decision on Video-MME: evaluations per configuration r, verdict noise σ, and power to detect a 1-point gain.

	Stage	$r\uparrow$	$\sigma\downarrow$	Power$\uparrow$
BRSI	Full harness	1.55	1.55	0.13
SRSI	I: warm-up	1.59	1.62	0.12
	II: branch	**9.80**	1.61	**0.50**
	III: merge	1.68	**1.12**	0.22
	Whole run	2.31	1.48	0.18

5 CONCLUSION

We introduced VIDEOHARNESS-BRSI and analyzed over 4,000 evolved harnesses. Evolution finds competitive harnesses, but its validation gains do not reliably generalize, because the same imperfect validation evidence adapts both the harness and the proposer's search strategy. VIDEOHARNESS-SRSI governs this evidence, improving held-out performance and narrowing the validation–test gap on four benchmarks.

Reproducibility Statement

We describe every component needed to reproduce our experiments. *Models:* the executor is the open-weight Qwen3-VL-8B-Instruct with greedy decoding; retrieval uses CLIP ViT-B/32 and ViT-L/14-336 and detection uses Grounding DINO (Appendix B.1). *Data:* all four benchmarks are public; the seeded validation/held-out partitions, their sizes, and the preprocessing of each benchmark are given in Appendix B.2 and Table 6. *Algorithms and settings:* VideoHarness-BRSI and VideoHarness-SRSI are specified in Algorithms 1 and 2, with the harness interface, leak guard, and evaluation protocol in Appendix C.1, per-question inference limits in Table 7, the seeded harnesses in Table 8, all stage settings and the per-seed candidate budget in Appendix J.5 and Table 17, and the information-isolation rules in Appendix J.1. *Prompts:* the main proposer prompts are reproduced in Appendix P. *Experiments and statistics:* Table 5 lists every study with its runs, budgets, and candidate counts; each appendix states its protocol, estimators, and tests, and all evolved results report the mean and standard deviation over three seeds. *Sources of variation:* the proposer is a proprietary model accessed through an API, so the exact candidates a run proposes cannot be reproduced token for token, and model versions may change; we therefore report results over independent seeds and analyze run-to-run variation explicitly (Section 2.2). Executor and tool outputs are cached by content, so re-evaluating a fixed harness reproduces its scores up to batching effects of at most ± 0.5 points.

AI Use Statement

AI as the object of study. Large language models are part of the systems we study: Claude Opus 5 (and, in the proposer ablation, Claude Sonnet 5 and Claude Opus 4.8) acts as the proposer that writes harness code inside the evolution loops, and Qwen3-VL-8B-Instruct is the frozen executor that answers benchmark questions. Their roles, prompts, and access restrictions are specified in Sections 2 and 3 and Appendices J and P. The harnesses they produce are experimental data and are evaluated as such; they are distinct from the research assistance described below.

AI assistance to the authors. Separately, the authors used generative AI assistants (LLM-based chat and coding assistants) in the following ways. *Research ideation and experimental design:* brainstorming research questions, hypotheses, and candidate experiments and ablations; the authors selected, specified, and approved all hypotheses, method designs, and experimental protocols. *Theoretical modeling:* drafting and checking the verdict model, propositions, and proofs in the appendix; the authors verified every statement and proof. *Implementation:* writing and debugging parts of the experiment orchestration, data-processing, analysis, and plotting code; the authors reviewed this code, tested it on controlled cases (e.g., a stubbed executor), and cross-checked reported statistics against raw run logs. *Result interpretation:* feedback on experimental results and alternative explanations; all interpretations and claims were made and checked by the authors against the data. *Writing:* grammar correction, rephrasing, restructuring, and consistency checks of the manuscript. No AI system is an author, and no reported number was produced by an AI assistant rather than by the logged experiments. The authors take full responsibility for all content of this paper, including text, claims, proofs, code, and results produced with AI assistance.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

Anthropic. Introducing claude opus 5. `https://www.anthropic.com/news/claude-opus-5`, July 2026. Accessed: 2026-09-12.

W. Brian Arthur. Competing technologies, increasing returns, and lock-in by historical events. *The Economic Journal*, 99(394):116–131, 1989.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pp. 41–53, 2010.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Monica Brezzi and Tze Leung Lai. Incomplete learning from endogenous data in dynamic allocation. *Econometrica*, 68(6):1511–1516, 2000.

Jerker Denrell and James G. March. Adaptation as information restriction: The hot stove effect. *Organization Science*, 12(5):523–538, 2001.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.

Ronald A. Fisher. *The Genetical Theory of Natural Selection*. Clarendon Press, Oxford, 1930.

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, 2025. URL https://arxiv.org/abs/2405.21075.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026.

Zhengyang Liang, Daoan Zhang, Huichi Zhou, Rui Huang, Bobo Li, Yuechen Zhang, Shengqiong Wu, Xiaohan Wang, Jiebo Luo, Lizi Liao, and Hao Fei. Univa: Universal video agent towards open-source next-generation video generalist, 2025. URL https://arxiv.org/abs/2511.08521.

Minhua Lin, Juncheng Wu, Zijun Wang, Zhan Shi, Yisi Sang, Bing He, Zewen Liu, Tianxin Wei, Zongyu Wu, Zhiwei Zhang, et al. Harness updating is not harness benefit: Disentangling evolution capabilities in self-evolving llm agents. *arXiv preprint arXiv:2605.30621*, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024.

Zewen Liu, Zhan Shi, Yisi Sang, Bing He, Minhua Lin, Tianxin Wei, Dakuo Wang, Benoit Dumoulin, Wei Jin, and Hanqing Lu. Adaptive auto-harness: Sustained self-improvement for agentic system deployment on open-ended task streams. *arXiv preprint arXiv:2606.01770*, 2026.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö. Arık. Vista: A test-time self-improving video generation agent, 2025a. URL https://arxiv.org/abs/2510.15831.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory, 2025b. URL https://arxiv.org/abs/2508.09736.

Frederic M. Lord and Melvin R. Novick. *Statistical Theories of Mental Test Scores*. Addison-Wesley, Reading, MA, 1968.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-rag: Visually-aligned retrieval-augmented long video comprehension, 2025. URL https://arxiv.org/abs/2411.13093.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Xuying Ning, Katherine Tieu, Dongqi Fu, Tianxin Wei, Zihao Li, Yuanchen Bei, Jiaru Zou, Mengting Ai, Zhining Liu, Ting-Wei Li, Lingjie Chen, Yanjun Zhao, Ke Yang, Bingxuan Li, Cheng Qian, Gaotang Li, Xiao Lin, Zhichen Zeng, Ruizhong Qiu, Sirui Chen, Yifan Sun, Xiyuan Yang, Ruida Wang, Rui Pan, Chenyuan Yang, Dylan Zhang, Liri Fang, Zikun Cui, Yang Cao, Pan Chen, Dorothy Sun, Ren Chen, Mahesh Srinivasan, Nipun Mathur, Yinglong Xia, Hong Li, Hong Yan, Pan Lu, Lingming Zhang, Tong Zhang, Hanghang Tong, and Jingrui He. Code as agent harness, 2026. URL https://arxiv.org/abs/2605.18747.

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

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, 2024.

H. Allen Orr. Adaptation and the cost of complexity. *Evolution*, 54(1):13–20, 2000.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7957–7968, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Michael Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9 (2):185–202, 1974.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1232–1240, 2016.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic llm agent search in modular design space. In *International Conference on Learning Representations*, volume 2025, pp. 3841–3865, 2025.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

James E. Smith and Robert L. Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pp. 58–76. Springer, 2024.

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, 2025. URL https://arxiv.org/abs/2405.19209.

Tianxin Wei, Zhan Shi, Minhua Lin, Bing He, Zewen Liu, Yisi Sang, Yuanchen Bei, Xuying Ning, Jiaru Zou, Ting-Wei Li, et al. Evo-harness: Context-to-harness skill compilation for self-evolving agents. *arXiv preprint arXiv:2608.15071*, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

Siwei Wu, Jincheng Ren, Yizhi Li, Haau-Sing Li, Chengran Yang, Yuxuan Zhang, Weicheng Gu, Jian Yang, Riza Batista-Navarro, Chuanyi Zhang, Xianglong Liu, Ming Zhou, Bryan Dai, and Chenghua Lin. Modularrsi: Modular and generalizable recursive harness self-improvement, 2026. URL https://arxiv.org/abs/2609.14857.

Peng Xia, Rujun Han, Zifeng Wang, Yanfei Chen, Yufan Zhang, Yoonho Lee, Chengsong Huang, Han Yu, Zhongying CuiZhu, Yifei Ming, Huaxiu Yao, Burak Gokturk, Tomas Pfister, and Chen-Yu Lee. Rrsi: Regularized recursive self-improvement of agent harnesses, 2026. URL https://arxiv.org/abs/2609.24972.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Woongyeong Yeo, Kangsan Kim, Jaehong Yoon, and Sung Ju Hwang. Worldmm: Dynamic multimodal memory agent for long video reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25599–25609, 2026.

Huaying Yuan, Jian Ni, Zheng Liu, Yueze Wang, Junjie Zhou, Zhengyang Liang, Bo Zhao, Zhao Cao, Ji-Rong Wen, and Zhicheng Dou. Momentseeker: A task-oriented benchmark for long-video moment retrieval. *Advances in Neural Information Processing Systems*, 38, 2026.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Guibin Zhang, Kaijie Chen, Guancheng Wan, Heng Chang, Hong Cheng, Kun Wang, Shuyue Hu, and Lei Bai. Evoflow: Evolving diverse agentic workflows on the fly. *arXiv preprint arXiv:2502.07373*, 2025a.

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, 2026a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025b.

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, 2026b.

OVERVIEW OF THE APPENDIX

The appendix follows the order of the paper, and Table 5 indexes every experiment with its scale and the appendix that documents it.

- Appendix A reviews related work on recursive self-improvement and video understanding harnesses.
- Appendix B describes the models, benchmarks, splits, inference budgets, seeded harnesses, and metrics.
- Appendix C specifies VIDEOHARNESS-BRSI: its evolution loop and study configurations (Section 2).
- Appendix E formalizes the verdict model of Eq. (1), and Appendices F–H analyze the three failure modes of Section 2.2, each with its protocol, full results, propositions and proofs, and robustness checks.
- Appendix I reports the parent-selection and initialization studies behind the simple remedies discussed in Section 2.2.
- Appendix J gives the implementation, statistical procedures, and configuration of VIDEOHARNESS-SRSI (Section 3).
- Appendix K gives the protocol and additional analyses of the warm-up ablation (Section 4.2).
- Appendix L ablates information isolation on LongVideoBench, and Appendix M re-scores held-out results on videos absent from validation.
- Appendix N reports the inference cost of the evolved harnesses and the budget-matched comparison (Section 4.1).
- Appendix O reports failure-mode diagnostics of VIDEOHARNESS-SRSI, and Appendix P reproduces its main proposer prompts.

A RELATED WORK

Recursive Self-Improvement for Harnesses. Automated optimization around frozen language models has progressed from prompts (Fernando et al., 2023; Yang et al., 2024; Pryzant et al., 2023; Yuksekgonul et al., 2024) to multi-stage programs (Khattab et al., 2023; Opsahl-Ong et al., 2024), reflective prompt evolution (Agrawal et al., 2026), and executable workflows and agent architectures (Hu et al., 2025; Zhang et al., 2025b;a; Novikov et al., 2025; Zhang et al., 2026a). Recent work evolves complete harnesses from past code, scores, and traces (Lee et al., 2026), searches modular design spaces (Shang et al., 2025; Wu et al., 2026), regularizes edits against overfitting (Xia et al., 2026), and studies harness updating (Lin et al., 2026), continual adaptation (Liu et al., 2026), and skill compilation (Wei et al., 2026). These methods refine how harnesses are proposed but take the validation signal that selects them as given. We instead make this signal the object of study: we show that it misleads evolution in three systematic ways and derive each component of VIDEOHARNESS-SRSI from one of them. Rather than regularizing edits, we control what information reaches the proposer; rather than modularizing search for efficiency, we factorize it so that each mechanism accumulates sufficient evidence and factorial composition tests whether gains transfer; and we counter the path dependence of noisy early verdicts, which prior methods leave unaddressed, with instrumented warm-up experiments.

Video Understanding Harnesses. Hand-designed video harnesses make long videos tractable through compression (Shen et al., 2024), keyframe selection (Tang et al., 2025), hierarchical representation (Wang et al., 2025), and evidence retrieval (Luo et al., 2025), while agentic systems coordinate retrieval, tools, and memory (Wang et al., 2024; Zhang et al., 2026b; Yeo et al., 2026; Long et al., 2025b). Each commits to one design; we treat these mechanisms as a search space in

Table 5: **Index of experiments.** All experiments use the frozen Qwen3-VL-8B-Instruct executor and Claude Opus 5 as the proposer. *Budget*: iterations × candidates per iteration of one run. *Candidates*: scheduled total over the study's runs; for the early-experience trap, the candidates actually proposed (the "over 4,000" of the abstract). Studies share runs, so the column does not sum. BRSI: VIDEOHARNESS-BRSI; SRSI: VIDEOHARNESS-SRSI. VM: Video-MME; LVB: LongVideoBench; OVO: OVO-Bench; MS: MomentSeeker.

Study	Benchmarks	Factor varied	Runs	Budget	Candidates	Where
Diagnosis of VIDEOHARNESS-BRSI *(Section 2)*						
Effectiveness	VM	– (retrieval only)	1	50×3	150	Fig. 2
Validation overfitting	LVB	proposer feedback; validation fraction	27	30×3	2,430	App. F
Search-budget dilution	VM	number of mechanism families (1, 2, 3, 6)	11	varies[a]	1,820	App. G
Early-experience trap	VM, LVB	– (re-analysis of existing runs)	44	varies[b]	4,806	App. H
Simple remedies (Section 2.2)						
Parent selection	VM, LVB	parent-suggestion rule	18	30×3	1,620	App. I.1
Initialization	VM	starting harness (simple seeds vs. AKS)	7	50×3	1,050	App. I.2
Main results and ablations (Section 4)						
Main results: BRSI	all four	–	12	50×3	1,800	Tab. 1
Main results: SRSI	all four	–	12	staged[c]	$\leq$1,872	Tab. 1
Warm-up ablation	VM	warm-up slot allocation	6	staged[c]	$\leq$936	App. K
Axis selection	VM	who defines the branch axes	9	staged[c]	$\leq$1,404	Fig. 5

[a] Per space in Table 13. [b] Per setting in Table 15. [c] At most 156 candidates per run (Eq. (19)); totals are upper bounds.

which multimodal feedback invites specialization to validation instances and a mechanism's value hinges on its context.

B EXPERIMENTAL SETUP

This section describes the models, benchmarks, inference budgets, seeded harnesses, and metrics shared by all experiments; Table 5 lists the scale of each experiment.

B.1 MODELS

The **executor** is a frozen Qwen3-VL-8B-Instruct (Bai et al., 2025) with greedy decoding. It is the only model that sees video or answers questions; "roles" and "agents" inside a harness are prompts to the same model. Executor and tool outputs are cached by content, so re-evaluating an unchanged harness reproduces its outputs (batching varies seeded-harness scores by at most ± 0.5 points). The **proposer** is Claude Opus 5 (Anthropic, 2026), run as a Claude Code agent at maximum reasoning effort in a new session per iteration; it writes harness code but never answers questions or runs the benchmark. Query-conditioned retrieval ranks a per-video pool of frames (4 fps, at most 4,096) with CLIP ViT-B/32 and ViT-L/14-336 (Radford et al., 2021); open-vocabulary detection uses Grounding DINO (Liu et al., 2024).

B.2 BENCHMARKS AND DATA SPLITS

Each benchmark has one fixed validation/held-out partition (Table 6), drawn once and reused by every method, seed, and study except the validation ladder of Appendix F. Every partition in this paper, including that ladder, is a seeded random split of questions, except on MomentSeeker, where whole source videos are assigned to one side; on the other benchmarks, a held-out question may come from a video that also has validation questions (Appendix M). Evolution and selection use only the validation split, and held-out results never reach the proposer. All benchmarks are used

Table 6: **Benchmark splits.** Seeded random splits; the split unit is the question, or the source video for MomentSeeker.

Benchmark	Pool	Validation	Held-out	Split unit	Metric
Video-MME	2,700	450	2,250	question	accuracy
LongVideoBench	1,337	446	891	question	accuracy
OVO-Bench	1,468	489	979	question	accuracy
MomentSeeker	1,000	261	739	video	mIoU

Table 7: **Default inference settings and per-question limits.** The frame and pixel values are defaults that a harness may override per call; the limits are enforced by the interface.

	Multiple choice	Temporal grounding
Default frames per pass	16	32
Default pixels per frame	230,400	151,200
Default output tokens	16	64
Maximum frames per pass	768	256
Executor calls per question	12	
Tool calls per question	60	
Pixel re-rendering calls per question	12	
Working memory (characters)	900	

video-only: subtitles and metadata such as duration categories or task labels are withheld from harness code.

- **Video-MME**: all 2,700 questions on 900 videos.
- **LongVideoBench**: the released validation set (the test set has no public answers), after removing items with inconsistent answer indices.
- **OVO-Bench**: the backward-tracing and real-time-perception multiple-choice modes, each evaluated on the clip truncated at query time so that no future frames are visible.
- **MomentSeeker**: text-query moment retrieval, where each source video lies entirely on one side of the split.

B.3 Inference Budgets

Table 7 lists the default settings and per-question limits. A harness may override the frame count, pixel budget, and output length; MomentSeeker defaults to more, lower-resolution frames, since temporal grounding depends mainly on coverage. A primitive that fails or exhausts its budget returns an empty result, so every harness must return a valid answer without any tool. Search spaces without perception tools, including the LongVideoBench runs of Appendix F, remove them from the interface.

B.4 Seeded Harnesses

Six deliberately simple, hand-written harnesses (Table 8) seed every full-harness run and serve as fixed baselines. Self-reflection and Role Debate read the frames of Retrieval, and Tool-gated Selection reads as many, so their differences from Retrieval reflect the added mechanism rather than the frame budget. On MomentSeeker they predict a time span instead of an option letter. Where perception tools are disabled, Tool-gated Selection is replaced by Dense Uniform, one call on 64 uniform frames. Spaces with one or two mechanism families start instead from five simple seeds: uniform and block-center sampling, CLIP top-k retrieval, and two space-specific retrieval or selection seeds.

B.5 Evaluation Metrics

Multiple-choice benchmarks report accuracy, parsing the first valid option letter from the harness output (none scores as incorrect). MomentSeeker reports mIoU, the mean temporal IoU of the pre-

Table 8: **Seeded harnesses.** Frames: mean frames shown to the executor per Video-MME question, summed over calls.

Harness (code name)	Procedure	Frames
UNIFORM (`uniform`)	Samples frames uniformly over the video and answers in one grounded call.	16
RETRIEVAL (`clip_topk`)	Ranks pool frames by CLIP similarity to the question and options, backfills the selection with uniform frames to the budget, and answers in one call.	16
SELF-REFLECTION (`reflect_loop`)	Answers on the retrieved frames, confronts the executor with its own answer on the same frames, and takes the revision (at most two rounds).	32
TOOL-GATED SELECTION (`tool_survey`)	Asks the executor which entities would resolve the question, detects them in a retrieved candidate pool, and answers on the frames that contain them.	16
ROLE DEBATE (`debate_pair`)	A proponent argues for the leading option, a skeptic for the strongest rival, and an arbiter decides after looking at the frames again with both arguments.	80
MEMORY SURVEY (`memo_multiturn`)	Splits the video into shots, writes one observation per shot keyframe into working memory, answers with the memory digest, and re-retrieves once on a cue named by the executor if the answer remains uncertain.	56

dicted span, clamped to the video and maximized over annotated intervals (unparsable predictions score zero); it is also the evolution objective. A harness's cost is its mean number of frames shown to the executor per question, summed over calls. Evolved results average three seeds and report the standard deviation across seeds. Unless noted otherwise, paired comparisons of two harnesses use McNemar's test (McNemar, 1947), contrasts between groups of runs use Welch's t-test or exact permutation tests, and bootstrap intervals resample runs rather than questions, since candidates of one run are dependent.

C VIDEOHARNESS-BRSI DETAILS

This section specifies VIDEOHARNESS-BRSI as used in Section 2: the evolution loop (Appendix C.1) and the configurations of the individual studies (Appendix C.2).

C.1 EVOLUTION LOOP

A harness is a Python class whose `answer` method receives a context object for one question—the video, the question, and its options—and returns an answer. Every access to frames, the executor, tools, or memory goes through the context, which enforces the per-question limits of Table 7. Algorithm 1 summarizes the loop.

Algorithm 1 VIDEOHARNESS-BRSI: the propose–evaluate–accumulate loop.

Require: frozen VLM M; proposer P; seed harnesses $\mathcal{H}_0$; splits $\mathcal{S}_{\text{val}}$, $\mathcal{S}_{\text{test}}$; iterations T; candidates per iteration k

$\mathcal{D}_0 \leftarrow \{(H, \textsc{Eval}(H, \mathcal{S}_{\text{val}})) : H \in \mathcal{H}_0\}$ ▷ seed archive
for $t = 1, \ldots, T$ **do**
  start a new session of P in a workspace exposing $\mathcal{D}_{t-1}$, the traces, and $\mathcal{I}_{\text{val}}$
  P writes k harness files and a manifest: name, declared parent, hypothesis, mechanism tags
  for each candidate H in the manifest **do**
    if H fails to import or violates a leak rule **then** record the failure and skip H
    $\mathcal{E} \leftarrow \textsc{Eval}(H, \mathcal{S}_{\text{val}})$ ▷ per-question answers, correctness, cost, traces
    $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(H, \mathcal{E})\}$; update the Pareto frontier over (accuracy, mean frames)
evaluate the frontier, the validation-best harness, and $\mathcal{H}_0$ on $\mathcal{S}_{\text{test}}$ ▷ never shown to P
return the validation-best harness ▷ deployed

Proposer sessions. Each iteration starts a new session of the proposer (Appendix B.1) in the run's workspace. A task description specifies the objective, the harness interface, the mechanism families that the search space charters, and the candidate rules: every iteration must add k new candidates,

each changing a mechanism rather than only constants. It prescribes a workflow: analyze the archive and the execution traces, prototype on a stubbed or real executor, implement k new harnesses, check that each imports and passes the leak guard, and register them in a manifest. The manifest declares, for each candidate, the parent it modifies, a one-sentence hypothesis, and the mechanisms it uses. Unless stated otherwise, $k = 3$.

Persistent memory. Besides the archive, the only state that survives an iteration is a persistent memory directory of the proposer's agent environment, which later sessions load automatically (up to 160 KB per run in our logs). The proposer uses it to record its interpretation of verdicts, including lists of directions it considers exhausted; Appendix H.6 shows how such entries propagate early verdicts.

Evaluation. Each candidate is evaluated once on the full validation split, and its score is never recomputed. Executor and tool outputs are cached by content, so calls that repeat an earlier input, within or across candidates, return the cached output. Before evaluation, a static leak guard rejects any candidate that accesses the file system, the network, or evaluation logs; references dataset-identity or metadata handles such as the video path, question and video IDs, or the task type; hashes text; or embeds lookup tables (collection literals with more than 12 strings or strings longer than 4,000 characters). Harness code sees only the question, the options, and the video's frame count, frame rate, and duration. A candidate that fails to import, violates a leak rule, or crashes during evaluation is recorded as failed and cannot become the champion.

Frontier and deployment. The loop maintains the Pareto frontier of validation performance against mean frames per question. After the last iteration, the frontier members, the validation-best harness, and the seeded harnesses are evaluated once on the held-out split. The deployed harness is the validation-best harness. Held-out results are stored outside the proposer's workspace.

C.2 Study Configurations

Budgets. Unless stated otherwise, retrieval-family runs evolve for 50 iterations and full-harness runs for 30 (50 for VideoHarness-BRSI in the main results of Table 1, which use Multimodal feedback and no parent suggestions on every benchmark), with three candidates per iteration; one perception-tools run uses 100 iterations of five candidates (Table 13).

Parent suggestions. The default loop leaves the choice of parent entirely to the proposer. The parent-selection study of Appendix I.1 adds a list of suggested parents to the task description and varies only the rule that produces it. All other runs, including the full-harness runs of the search-space ladder and the main results, receive no parent suggestions; deployment is the validation-best harness in all cases.

D Evolving Capability of VideoHarness-BRSI

This section gives the experimental setting behind Section 2.1 and explains its three design choices: the evolved mechanism family, the validation/held-out split, and the iteration budget.

D.1 Experimental Setting

Evolving retrieval for a fair comparison. The question of Section 2.1 is whether recursive evolution can match expert manual design, so the comparison needs a strong hand-designed reference. We use Adaptive Keyframe Sampling (AKS) (Tang et al., 2025), which is a retrieval method: it scores frames by their relevance to the question and allocates the frame budget accordingly. We therefore let VideoHarness-BRSI evolve only the retrieval strategy in the main text. Evolved and hand-designed harnesses then differ in the same mechanism and run on the same frozen executor, so a gap between them reflects how the mechanism was designed rather than which mechanism was used. Evolving other or more mechanism families would give VideoHarness-BRSI components that AKS lacks, and a win could no longer be credited to evolution alone. Section D.2 reports evolution in other mechanism families and on other datasets.

A 1:5 validation/held-out split. We split Video-MME at random into 450 validation and 2,250 held-out questions, a ratio of 1:5 (Table 6). This split keeps most of the benchmark away from evolution, for two reasons.

- **Evaluation cost.** Every candidate is run in full on the validation split, which takes a frozen-VLM call on the sampled frames of every question. Evaluation is therefore the slow step of each iteration, and a small validation split shortens every one of the 150 evaluations in a run.
- **A stricter test of generalization.** A small validation split gives the proposer less of the benchmark to fit, while the large held-out split measures the deployed harness precisely: with 2,250 questions, the standard error of a held-out accuracy near 68% is about one point, against about two points on the 450 validation questions. A larger validation share would let evolution fit a larger part of the benchmark itself and leave a smaller, noisier held-out split to detect it. Under the 1:5 split, a harness evolved on one sixth of the data must match the hand-designed methods on the remaining five sixths, which it never sees. That the evolved harness still matches the strongest of them on this split (Figure 2(b)) therefore strengthens, rather than weakens, the conclusion that VIDEOHARNESS-BRSI is competitive with human design.

Iteration budget. Each run evolves for 50 iterations of three candidates, 150 candidate evaluations in total. This budget was chosen from experience with preliminary runs.

D.2 EVOLUTION ACROSS MECHANISM FAMILIES AND DATASETS

The retrieval-only comparison isolates one mechanism for a fair test against AKS, but the evolving capability of VIDEOHARNESS-BRSI is not specific to retrieval or to Video-MME.

Other mechanism families. The search-space ladder of Appendix G evolves, on the same Video-MME split, retrieval together with frame selection, with perception tools, and the full harness of six families, which adds reflection, role-based collaboration, and working memory. In every space, evolution raises validation accuracy over the best seeded harness by 5.1–7.6 points within 87 candidates, and the deployed champions reach 65.7–66.8% held-out accuracy (Table 14), above the best fixed baseline on Video-MME (`clip_topk`, 63.4%; Table 1). Evolution thus improves harnesses whichever mechanism families it may edit; what the families change is how efficiently it does so (Section 2.2.2).

Other datasets. The main results evolve the full harness on all four benchmarks (Table 1). On held-out data, VIDEOHARNESS-BRSI exceeds the strongest fixed baseline on every benchmark: 66.5 versus 63.4% on Video-MME, 66.6 versus 62.1% on LongVideoBench, 67.2 versus 55.4% on OVO-Bench, and 19.0 versus 18.0 mIoU on MomentSeeker, 5.12 points on average. Evolution therefore transfers across general, long-form, and streaming question answering and temporal grounding, although, as Section 2.2 shows, its validation progress overstates these held-out gains.

E A MODEL OF VALIDATION VERDICTS

This section derives the verdict model of Eq. (1); Appendices F–H instantiate it for the three failure modes.

Scores and instance-specific fit. Let $y_H(q) \in \{0, 1\}$ indicate that harness H answers question q correctly, with questions drawn from a population $\mathcal{Q}$. On a validation split $\mathcal{S}_{\text{val}}$ of n questions,

$$V(H) = \frac{100}{n} \sum_{q \in \mathcal{S}_{\text{val}}} y_H(q), \qquad \mu(H) = 100\, \mathbb{E}_{q \sim \mathcal{Q}}\big[y_H(q)\big], \tag{4}$$

and $T(H)$ is held-out accuracy, an unbiased estimate of $\mu(H)$ because the held-out split never influences search. $V(H)$ is not, since H is built from validation feedback. We decompose $V(H) = \mu(H) + S(H) + e(H)$ with

$$S(H) = \mathbb{E}\big[V(H) \mid H\big] - \mu(H), \qquad \mathbb{E}\big[e(H) \mid H\big] = 0. \tag{5}$$

The *instance-specific fit* $S(H)$ is the systematic, validation-specific part of the score; $S \equiv 0$ for any harness chosen without access to $\mathcal{S}_{\text{val}}$, such as a hand-written baseline. In psychometric terms (Lord & Novick, 1968), validation measures $\mu + S$ reliably but only μ validly, and the two separate exactly when S varies across candidates.

Paired verdicts and the behavioral footprint. A candidate is compared with the incumbent H^{inc} on the same questions. With $D_q = y_H(q) - y_{H^{\text{inc}}}(q)$ and n_{10}, n_{01} the numbers of questions with $D_q = 1$ and $D_q = -1$,

$$\hat{\delta} = \frac{100\,(n_{10} - n_{01})}{n}, \qquad d = n_{10} + n_{01}, \tag{6}$$

and, treating questions as exchangeable given the pair,

$$\operatorname{Var}(\hat{\delta}) = \frac{10^4}{n}\Big(\frac{\mathbb{E}\,d}{n} - \Big(\frac{\mathbb{E}\,\hat{\delta}}{100}\Big)^2\Big) \approx \frac{10^4\,d}{n^2}, \tag{7}$$

the scale of McNemar's statistic (McNemar, 1947). We call d the *behavioral footprint* and use the paired standard error $\sigma_H = 100\sqrt{d_H}/n$ and $z = \hat{\delta}/\sigma_H$. The key consequence is that an edit changing more answers yields a noisier verdict, whether or not the changes help. Exact clones ($d = 0$) get $z = 0$ and are excluded from averages of σ. Because the calculation ignores adaptive reuse of the validation set and within-video dependence, $|z| < 1.96$ is only a nominal marker of an unresolved comparison.

Incumbent optimism. The incumbent is the validation maximum so far, so the noise in its cached score is biased upward, $\omega = \mathbb{E}\big[e(H^{\text{inc}})\big] > 0$, even though $\mathbb{E}[e(H) \mid H] = 0$ for any fixed H (the optimizer's curse (Smith & Winkler, 2006)); for m equally accurate harnesses with independent Gaussian noise of standard deviation s, $\omega = s\,c_m$ with $c_m = \mathbb{E}\max_{i \le m} Z_i \approx \sqrt{2\ln m}$. Since the incumbent is never re-evaluated, every later verdict inherits $-\omega$. Subtracting the two decompositions,

$$\hat{\delta} = V(H) - V(H^{\text{inc}}) = \underbrace{\mu(H) - \mu(H^{\text{inc}})}_{\Delta} + \underbrace{S(H) - S(H^{\text{inc}})}_{S} - \omega + \underbrace{e(H) - e(H^{\text{inc}}) + \omega}_{\varepsilon},$$

which is Eq. (1) with S the candidate's fit relative to the incumbent's and $\mathbb{E}\,\varepsilon = 0$. Each validation-specific part of the incumbent's score thus enters once: its systematic fit $S(H^{\text{inc}})$ through S, and the selection bias of its noise through ω.

Three ways a verdict fails. Evolution improves μ only insofar as $\hat{\delta}$ tracks Δ. **Validation overfitting** (App. F): content-revealing feedback enlarges S, which selection rewards. **Search-budget dilution** (App. G): larger spaces lower the evidence per hypothesis through fewer evaluations per mechanism combination and larger footprints d. **Early-experience trap** (App. H): single low-SNR verdicts, offset by $-\omega$, are rarely revisited. The model is a lens, not a claim of exact Gaussianity; each appendix states its assumptions.

F Validation Overfitting

This section supports Section 2.2.1. Its conclusions are:

- **Richer feedback, larger gaps.** The mean validation–held-out gap of evolved harnesses grows from 2.0 (Blindfolded) to 2.9 (Trace) and 3.2 points (Multimodal), while hand-written baselines show none (Appendix F.2).
- **Theory.** Selecting on validation accuracy transfers only a fraction β of the validation advantage, and the non-transferable fit a harness can acquire is bounded by the information its feedback carries about the validation sample, which grows along nested channels (Appendix F.3).
- **Measurement.** Validation scores are equally reliable in every arm, yet β falls from 0.54 to 0.30, and the lost share reappears as instance-specific fit; the model predicts the winners' held-out accuracy to within 0.34 points (Appendix F.4).
- **Mechanism.** Proposers that see question text quote validation questions in their code $1.55\times$ (Trace) and $1.27\times$ (Multimodal) as often as chance, whereas Blindfolded proposers do not (Appendix F.5).

F.1 PROTOCOL

All 27 runs use LongVideoBench (video only), the frozen Qwen3-VL-8B-Instruct executor, and the full-harness space with perception tools disabled, evolving for 30 iterations of three candidates. The 1,337 scorable questions are split at random, as in Appendix B.2, into nested validation sets of 33% (446 / 891 validation / held-out questions), 50% (668 / 669), and 75% (1,003 / 334). Each arm runs three seeds per ladder. Only validation-frontier candidates are evaluated on the held-out split, which the proposer never sees. The three arms differ only in what the proposer observes about the validation set (Table 9); each arm's feedback is a deterministic function of the next one's. After dropping 13 incomplete validation/held-out pairs, 281 proposed candidates remain (89 Blindfolded, 103 Trace, 89 Multimodal), against $6 \times 3 = 18$ distinct hand-written baseline values.

Table 9: **Information arms** (three seeds $\times$ three ladders per arm).

Arm	What the proposer observes about the validation set
Blindfolded	Per-question predicted letter and correctness, and about ten behavioral counters (model and tool calls, frames, rendering, memory). No question text, options, or frames.
Trace	Blindfolded feedback plus each question, its options, the full interaction trace, and the first raw answer.
Multimodal	Trace feedback plus on-demand rendered validation frames, and a written anti-overfitting protocol.

F.2 RESULTS

Table 10 reports the gaps behind Figure 3(a). The gap grows monotonically with feedback when ladders are pooled; per-ladder orderings are noisy with three runs per cell. Winners' validation accuracy rises with feedback ($70.2 \rightarrow 71.1\%$) while their held-out accuracy declines ($66.5 \rightarrow 65.7\%$), a trend within the confidence intervals.

Table 10: **Validation–held-out gaps (pp).** Candidate CIs are 95% bootstrap intervals over candidates, with run-clustered intervals in brackets. "Winner" is each run's validation argmax, averaged over the nine runs of an arm.

	n	Mean gap, candidates	Val > held-out	Winner val / held-out (%)	Winner gap
Hand-written	18	-0.16 $[-1.07, 0.82]$	39%	–	–
Blindfolded	89	2.02 $[1.62, 2.39]$ $[1.11, 2.95]$	84%	70.2 / 66.5 $[65.6, 67.4]$	3.69
Trace	103	2.91 $[2.43, 3.39]$ $[2.13, 4.00]$	89%	71.0 / 66.3 $[64.4, 67.8]$	4.76
Multimodal	89	3.21 $[2.73, 3.71]$ $[2.16, 4.22]$	90%	71.1 / 65.7 $[64.8, 66.5]$	5.46

F.3 THEORY: SELECTION ON A PARTIALLY TRANSFERABLE SIGNAL

Model. For the m candidates $h_1, \ldots, h_m$ of one run, Eq. (1) written per candidate gives

$$V_i = \mu_i + S_i + e_i, \qquad T_i = \mu_i + e'_i, \tag{8}$$

with population accuracy μ_i, instance-specific fit S_i, and split noise e_i, e'_i. With $\sigma_V^2 = \sigma_\mu^2 + \sigma_S^2 + \sigma_e^2$, the *transfer coefficient* is

$$\beta = \frac{\mathrm{Cov}(V, T)}{\mathrm{Var}\, V} = \frac{\sigma_\mu^2}{\sigma_\mu^2 + \sigma_S^2 + \sigma_e^2}. \tag{9}$$

Relation to Eq. (1). A candidate's verdict is $\hat{\delta}_i = V_i - V(H^{\mathrm{inc}})$. For candidates compared against the same incumbent, the incumbent's score, including its optimism ω, is a common offset, so $\mathrm{Var}_i\, \hat{\delta}_i = \mathrm{Var}\, V$ and the three terms of Eq. (1) that vary across candidates are $\Delta_i = \mu_i - \mu(H^{\mathrm{inc}})$, S_i, and $\varepsilon_i = e_i$ up to constants. Hence

$$\beta = \frac{\mathrm{Var}\, \Delta}{\mathrm{Var}\, \Delta + \mathrm{Var}\, S + \mathrm{Var}\, \varepsilon},$$

the share of verdict variance carried by the only transferable term; $1-\beta$ splits into an instance-specific share (S) and a noise share (ε), which Appendix F.4 measures separately. ω shifts every verdict equally and therefore affects which verdicts clear zero (Appendix H.2), not β. The reliability κ of Proposition H.1 is the same ratio estimated on paired verdicts; here we write it on scores because held-out accuracy T_i is observed per candidate.

Proposition F.1 (Selection on a partially transferable signal) *Let (μ_i, S_i, e_i, e_i') be independent across i, with independent Gaussian components of means $(\bar{\mu}, \bar{S}, 0, 0)$ and variances $(\sigma_\mu^2, \sigma_S^2, \sigma_e^2, \sigma_{e'}^2)$. Let $i^\star = \arg\max_i V_i$, $H^\star = h_{i^\star}$, $\bar{V} = \bar{\mu} + \bar{S}$, $\bar{T} = \bar{\mu}$, and $c_m = \mathbb{E}[\max_{i \le m} Z_i]$ for i.i.d. standard normal Z_i. Then*

(i) $\mathbb{E}[T_i \mid V_1, \dots, V_m] = \bar{T} + \beta\,(V_i - \bar{V})$ *for every i;*

(ii) $\mathbb{E}[T(H^\star)] - \bar{T} = \beta\,\sigma_V c_m$, *whereas* $\mathbb{E}[V(H^\star)] - \bar{V} = \sigma_V c_m$;

(iii) $\mathbb{E}[V(H^\star) - T(H^\star)] = \bar{S} + (1-\beta)\,\sigma_V c_m$;

(iv) with σ_μ^2, σ_e^2, and m fixed, a larger σ_S raises $\mathbb{E}[V(H^\star)]$ and the gap but lowers $\mathbb{E}[T(H^\star)]$.

Proof. (i) By independence across candidates, $\mathbb{E}[T_i \mid V_1, \dots, V_m] = \mathbb{E}[T_i \mid V_i]$. Since (T_i, V_i) is bivariate Gaussian with $\mathrm{Cov}(T_i, V_i) = \sigma_\mu^2$, the conditional mean is linear with slope $\sigma_\mu^2/\sigma_V^2 = \beta$.

(ii) The index $i^\star$ is a function of $(V_1, \dots, V_m)$, so by the tower property and (i),

$$\mathbb{E}[T_{i^\star}] = \mathbb{E}\Big[\textstyle\sum_i \mathbf{1}\{i^\star = i\}\,\mathbb{E}[T_i \mid V_1, \dots, V_m]\Big] = \bar{T} + \beta\,\big(\mathbb{E}[V_{i^\star}] - \bar{V}\big),$$

$$\mathbb{E}[V_{i^\star}] - \bar{V} = \sigma_V c_m \qquad \text{since } V_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\bar{V}, \sigma_V^2).$$

(iii) Subtracting (ii) from $\mathbb{E}[V_{i^\star}]$ and using $\bar{V} - \bar{T} = \bar{S}$ gives the gap.

(iv) Writing each term as a function of σ_V, which increases with σ_S,

$$\mathbb{E}[V_{i^\star}] = \bar{V} + \sigma_V c_m \qquad \text{(increasing in } \sigma_V),$$

$$\mathbb{E}[T_{i^\star}] = \bar{T} + \frac{\sigma_\mu^2}{\sigma_V}\, c_m \qquad \text{(decreasing in } \sigma_V),$$

$$\frac{\partial}{\partial \sigma_S}\, \frac{c_m(\sigma_S^2 + \sigma_e^2)}{\sigma_V} = \frac{c_m\,\sigma_S(\sigma_V^2 + \sigma_\mu^2)}{\sigma_V^3} > 0 \qquad \text{(gap increasing in } \sigma_S). \qquad \square$$

Remarks. With $\sigma_S = 0$ the proposition reduces to the optimizer's curse (Smith & Winkler, 2006): selection alone overfits by $(\sigma_e^2/\sigma_V)\,c_m$, and adaptive fit adds a second, feedback-dependent term. Statement (i) conditions on V, so β can be estimated from the frontier candidates that receive held-out evaluations; it also survives correlated lineages as long as each candidate's quality is mean-independent of the others' scores given its own.

Proposition F.2 (Information bound and nested feedback channels) *Let $\mathcal{S}_{\mathrm{val}} = (Z_1, \dots, Z_n) \overset{\text{i.i.d.}}{\sim} \mathcal{Q}$, let $\ell(h, Z) \in \{0, 1\}$ indicate a correct answer, and let $V(h) = \frac{100}{n}\sum_j \ell(h, Z_j)$ and $\mu(h) = 100\,\mathbb{E}_{Z\sim\mathcal{Q}}\ell(h, Z)$. Let the proposer output H from its feedback and randomness independent of $\mathcal{S}_{\mathrm{val}}$. Then*

(i) $\big|\mathbb{E}[V(H) - \mu(H)]\big| \le 100\sqrt{I(\mathcal{S}_{\mathrm{val}}; H)/(2n)}$;

(ii) a harness chosen without access to $\mathcal{S}_{\mathrm{val}}$ has $I(\mathcal{S}_{\mathrm{val}}; H) = 0$ and no expected gap;

(iii) if the observation maps satisfy $\psi_{\mathrm{blind}} = g \circ \psi_{\mathrm{trace}}$ and $\psi_{\mathrm{trace}} = g' \circ \psi_{\mathrm{mm}}$ for deterministic g, g', then

$$I(\mathcal{S}_{\mathrm{val}}; \psi_{\mathrm{blind}}) \le I(\mathcal{S}_{\mathrm{val}}; \psi_{\mathrm{trace}}) \le I(\mathcal{S}_{\mathrm{val}}; \psi_{\mathrm{mm}}),$$

and any leakage $I(\mathcal{S}_{\mathrm{val}}; H)$ achievable with Blindfolded feedback is achievable with Trace feedback, and likewise from Trace to Multimodal.

Proof. (i) Since $\ell \in [0, 1]$, the empirical mean over n i.i.d. questions is $\frac{1}{2\sqrt{n}}$-sub-Gaussian for every fixed h. The mutual-information generalization bound (Russo & Zou, 2016; Xu & Raginsky, 2017)

then gives

$$\left|\mathbb{E}\left[\tfrac{V(H)-\mu(H)}{100}\right]\right| \le \sqrt{2 \cdot \tfrac{1}{4n} I(\mathcal{S}_{\mathrm{val}}; H)}.$$

(ii) With $I = 0$ the bound vanishes.

(iii) The chain follows from the data-processing inequality along $\mathcal{S}_{\mathrm{val}} \to \psi_{\mathrm{mm}} \to \psi_{\mathrm{trace}} \to \psi_{\mathrm{blind}}$. A Trace proposer can apply g to each observation and run any Blindfolded strategy exactly, so every joint law of $(\mathcal{S}_{\mathrm{val}}, H)$ achievable under Blindfolded feedback is achievable under Trace feedback, and likewise from Trace to Multimodal. □

What the bound implies. The bound limits how much adaptive fit a channel permits, not how a proposer uses it. Inverting (i), $I(\mathcal{S}_{\mathrm{val}}; H) \ge 2n\,(\mathrm{gap}/100)^2$ nats: plugging in the winners' mean excess gaps, they must carry at least 1.9 (Blindfolded), 3.3 (Trace), and 4.1 nats (Multimodal) about their validation sample, an illustrative lower bound that grows with feedback.

F.4 Measuring Transfer and Instance-Specific Fit

The three shares. For the evolved harnesses H_i, $i = 1, \ldots, N$, of a run, let $V(H_i)$ and $T(H_i)$ be the validation and held-out accuracy, $V^A(H_i)$ and $V^B(H_i)$ the accuracy on two random halves A, B of the validation questions, and Var_i, Cov_i the variance and covariance over the N harnesses. The shares of Figure 3(d) are

$$\rho_\Delta = \frac{\mathrm{Cov}_i\big(V(H_i), T(H_i)\big)}{\mathrm{Var}_i\, V(H_i)}, \qquad \rho_\varepsilon = \frac{\mathrm{Var}_i\big(V^A(H_i) - V^B(H_i)\big)}{4\,\mathrm{Var}_i\, V(H_i)}, \qquad \rho_S = 1 - \rho_\Delta - \rho_\varepsilon. \tag{10}$$

Across the harnesses of a run, validation scores differ by $\Delta + S + \varepsilon$ and held-out scores by $\Delta + \varepsilon'$ with independent noise ε'. Since Δ is the only term the two share, $\rho_\Delta = \mathrm{Var}\,\Delta / \mathrm{Var}\, V$, which is also the slope of held-out on validation accuracy: a harness one point higher on validation is on average ρ_Δ points higher on held-out data. A harness's two half-scores share its Δ and S and differ only by noise; each half carries twice the noise variance of the full set, so $\mathrm{Var}(V^A - V^B)$ is four times the noise variance of a full validation score, and ρ_ε needs no held-out data. The validation-fit share ρ_S is the remainder. The paragraphs below give the estimators pooled over runs; ρ_Δ is the transfer coefficient β.

Transfer coefficient. With run means $\bar{V}_r$, $\bar{T}_r$ over tested candidates, we estimate β within runs, which removes run-level offsets from the seed, starting harness, and split:

$$\hat{\beta} = \frac{\sum_r \sum_{h \in r} (V_h - \bar{V}_r)(T_h - \bar{T}_r)}{\sum_r \sum_{h \in r} (V_h - \bar{V}_r)^2}.$$

Resampling runs (10,000 replicates) gives

$$\hat{\beta}_{\mathrm{blind}} = 0.54\ [0.33, 0.74], \qquad \hat{\beta}_{\mathrm{trace}} = 0.35\ [0.24, 0.44], \qquad \hat{\beta}_{\mathrm{mm}} = 0.30\ [0.15, 0.41],$$

with $\Pr(\beta_{\mathrm{blind}} > \beta_{\mathrm{mm}}) = 0.98$ and a one-sided permutation $p = 0.010$ for $\beta_{\mathrm{blind}} - \beta_{\mathrm{mm}}$. **Half or less of a validation difference survives on held-out data, and less so with richer feedback.**

Variance decomposition. Split-half resampling isolates noise without held-out data: for 200 random halves A, B of the validation questions, $\frac{1}{4}\mathrm{Var}(V^A - V^B)$ estimates σ_e^2, unaffected by selection on V. With $\mathrm{Var}\, V$ computed within runs over all complete proposals (excluding, of 2,319, the 29 catastrophic ones more than 10 pp below the best baseline and one with a partial evaluation), the shares of Figure 3(d) are

$$\hat{\rho}_\Delta = \hat{\beta}, \qquad \hat{\rho}_S = 1 - \hat{\beta} - \hat{\sigma}_e^2 / \widehat{\mathrm{Var}} V, \qquad \hat{\rho}_\varepsilon = \hat{\sigma}_e^2 / \widehat{\mathrm{Var}} V.$$

Table 11 shows that the noise share is the same in every arm ($\hat{\sigma}_e^2 \approx 1.0\,\mathrm{pp}^2$), so **the validation score is equally reliable, but its instance-specific share grows from** 0.31 **to** 0.56; it exceeds Blindfolded's in 98% (Multimodal) and 96% (Trace) of bootstrap replicates.

Predicting the winners. Proposition F.1(ii) predicts each winner as $\hat{T}(H^\star) = \bar{T}_r + \hat{\beta}_{-r}\,(V(H^\star) - \bar{V}_r)$, with $\hat{\beta}_{-r}$ fit on the arm's other runs. The predictions (67.0, 66.0, 66.0%) match the observed 66.5, 66.3, and 65.7% with mean absolute error 0.34 pp, whereas full transfer ($\beta = 1$) overestimates by 1.5–2.2 pp (Figure 6a).

Table 11: **Reliability versus validity of the validation score.** Shares of across-candidate validation variance, with 95% run-cluster bootstrap CIs; reliability is one minus the noise share.

Arm	Candidates	Transfers (β)	Instance-specific	Noise	Reliability
Blindfolded	752	0.54 [0.33, 0.74]	0.31 [0.13, 0.50]	0.16 [0.14, 0.17]	0.84
Trace	774	0.35 [0.24, 0.44]	0.51 [0.42, 0.62]	0.14 [0.11, 0.18]	0.86
Multimodal	763	0.30 [0.15, 0.41]	0.56 [0.44, 0.70]	0.14 [0.11, 0.19]	0.86

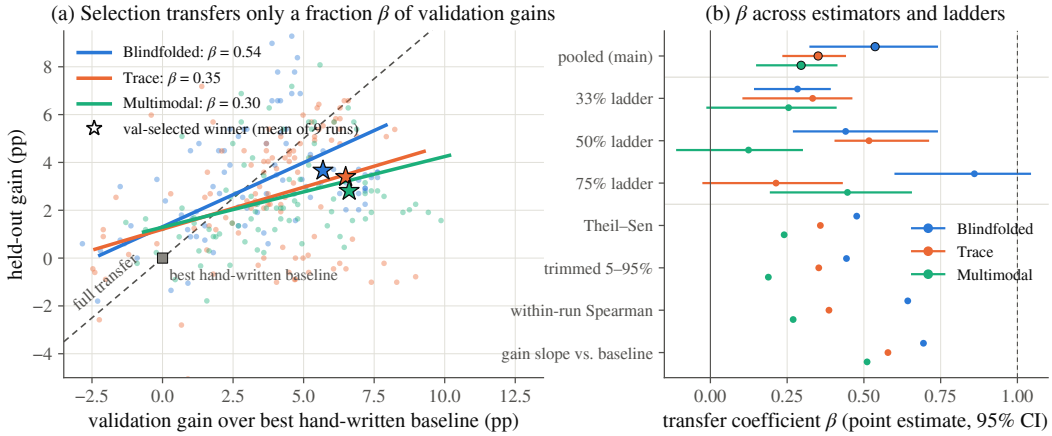


Figure 6: **Transfer of validation gains.** (a) Held-out versus validation gain over the run's best hand-written baseline for every tested candidate. Lines have slope $\hat{\beta}$ through the arm's mean; stars mark the mean winners. Selection moves right along a line, so a flatter line converts the same validation advantage into less held-out gain. (b) β under alternative estimators and per ladder, with 95% CIs.

F.5 Direct Evidence of the Content Channel

Validation and held-out questions are a random split of one pool, so text chosen without regard to the split matches validation questions at the base rate $\pi = |\text{validation}|/|\text{pool}|$. Using the 18 runs of the 33% and 50% ladders, for which question text of both splits is available within the 75% validation set ($\pi = 0.445$ and 0.666), we collect every word 5-gram in evolved code that occurs in at most two question stems and test its matches against π by re-drawing the split 20,000 times. The leak guard (Appendix C.1) rules out lookup tables by construction.

Table 12 shows that **proposers that see question text quote validation questions beyond chance, and Blindfolded proposers do not.** The quotations sit almost entirely in docstrings and comments that record the validation cases motivating a design (e.g., "a white bowl containing potatoes"): of the 168 matched Trace questions, 144 are quoted in docstrings, 25 in comments, and only 4 in functional literals (a question can be quoted in several places), where vocabulary is at most weakly enriched (ratios 0.94–1.08). Content thus enters through design decisions rather than copied keywords, producing the behavioral fit measured above.

Table 12: **Verbatim validation-question 5-grams in evolved code** (33% and 50% ladders, six runs per arm). p: one-sided permutation test over the split.

Arm	Matched questions	Validation	Expected	Ratio	p
Blindfolded	50	26	27.1	0.96	0.68
Trace	233	168	108.3	1.55	$< 10^{-4}$
Multimodal	69	49	38.7	1.27	0.007

F.6 Robustness

Robustness. The pooled ordering Blindfolded > Trace > Multimodal of β holds under Theil–Sen slopes, trimming, median per-run slopes, within-run Spearman correlations, and gain slopes relative

Table 13: **The search-space ladder on Video-MME.** K: chartered mechanism families; P: callable primitives. The deployed champion is the validation-selected harness at the end of each run.

Space	K	P	Runs	Iterations × candidates	Champion found at iteration
Retrieval only	1	8	3	50×3	21, 46, 50
+ frame selection	2	8	3	50×3	28, 45, 39
+ perception tools	3	12	2	50×3; 100×5	49, 99
Full harness	6	33	3	30×3	28, 29, 21

to the best baseline (Figure 6b). Blindfolded exceeds Multimodal in every leave-one-ladder-out fit, whereas the adjacent orderings can reverse, so the contrast between the two ends of the information range is the robust one. Moving the catastrophic-proposal threshold to 5 or 15 pp changes the variance shares by at most 0.03, and even retaining all catastrophic proposals preserves the ordering of instance-specific shares (0.42, 0.57, 0.65).

G Search-Budget Dilution

This section supports Section 2.2.2. Its conclusions are:

- **Larger spaces do not help.** Deployed champions reach 66.8, 65.8, 66.0, and 65.7% held-out accuracy with one, two, three, and six mechanism families, and 94–100% of promotions in every space are statistically unresolved (Appendix G.2).
- **Theory.** The power to detect an advantage Δ is bounded by the evidence $\lambda = r\Delta^2/\bar{\sigma}^2$, which a larger space erodes through breadth (fewer evaluations r per mechanism combination) and coupling (larger footprints, hence larger σ^2) (Appendix G.3).
- **Measurement.** From one to six families, r falls $2.05\times$ and the verdict variance rises $2.24\times$, so the power for a 1-point gain drops from 0.45 to 0.14. Full-harness edits flip a median of 50 answers against 10, and since only 37% of flips are fixes, larger edits are also more harmful (Appendix G.4).

G.1 Protocol

All runs use Video-MME (450 validation / 2,250 held-out questions) with the same executor, proposer, loop, and seeds; only the search space and the iteration budget differ (Table 13), and all comparisons are made at a matched budget of 87 candidates. The nested spaces charter retrieval only (one family, 8 primitives); retrieval with frame selection (two families, 8 primitives), where a candidate may change either or both; perception tools (three families, 12 primitives); and the full harness (six families, 33 primitives), which adds reflection, role-based collaboration, and working memory. The main text contrasts the one- and six-family spaces; the two- and three-family spaces are reported only here. Two seeds were excluded before analysis because of a data error.

A candidate's *mechanism combination* is the set of mechanism tags its implementation exercises (retrieval, re-ranking, fusion, query rewriting, frame selection, reflection, prompting, pixel manipulation, perception tools, memory, routing, and frame budget; these are finer than the families of the space), assigned by fixed, outcome-blind keyword rules shared with Appendix H. Each candidate is compared with the champion in force when it was proposed, using $\hat{\delta}$, d, and $\sigma = 100\sqrt{d}/n$ from Eqs. (6)–(7). Evidence statistics use the first 87 candidates of every run, the largest budget all eleven runs reach, because evaluations per mechanism combination are comparable only at a common budget.

G.2 Results

No space improves held-out accuracy. Table 14 and Figure 7(a,b) extend the main-text comparison with a three-family space. That space reaches the largest validation gain (7.6 points after 87 candidates, 13.9 at the end), yet its champions beat the seed harness by only 3.5 points on held-out data. **Low evidence can thus appear as validation inflation rather than slow progress**: a third of its early proposals are near-clones differing from the champion on at most 1% of questions, and the

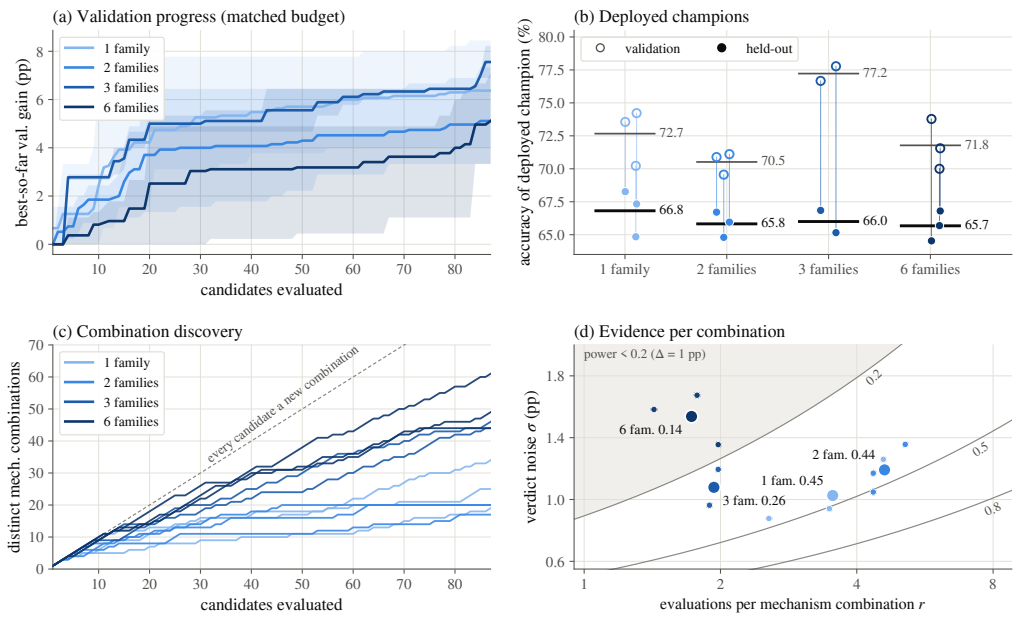


Figure 7: **All four spaces.** (a) Best-so-far validation gain against candidates evaluated (mean and min–max over runs), on a budget-matched axis. (b) Validation (hollow) and held-out (filled) accuracy of each run's deployed champion; bars are means. (c) Distinct mechanism combinations visited against candidates evaluated; the dashed line is the limit in which every candidate is a new mechanism combination. (d) Evidence landscape of Figure 3(e) with the three-family space, which has the breadth of the full harness but the footprint of retrieval-only evolution.

Table 14: **Outcome and evidence by space.** Validation gain is the best-so-far gain over the run's best seed harness after 87 candidates; accuracies are those of the deployed champions. Evidence columns use the first 87 candidates of each run: distinct mechanism combinations, evaluations per mechanism combination r, mean footprint d, mean verdict noise σ (pp), mean verdict $\hat{\delta}$ (pp), and the noncentrality λ and power of Proposition G.1 for an advantage Δ. All entries are means over runs.

	Outcome (%)			Search behavior					Power for Δ (pp)		
Space	Val. gain	Val.	Held-out	Combos	r	d	σ	$\hat{\delta}$	0.5	1	2
Retrieval only (1)	6.4	72.7	66.8	26.0	3.54	25.3	1.03	−1.10	0.15	0.45	0.95
+ frame selection (2)	5.1	70.5	65.8	19.0	4.61	31.8	1.19	−1.42	0.15	0.44	0.95
+ perception tools (3)	7.6	77.2	66.0	45.0	1.93	23.6	1.08	−0.76	0.10	0.26	0.73
Full harness (6)	5.1	71.8	65.7	51.3	1.73	51.3	1.54	−3.14	0.07	0.14	0.41

largest of many small, noisy verdicts is promoted. Larger spaces are also costlier: held-out gain per evaluation hour falls from 0.79 (one family) to 0.17 points (six), so counting candidates rather than time favors them.

Promotions are unresolved everywhere. Within the first 87 candidates, 94–100% of promotions lie within ± 1.96 paired standard errors, with mean size 0.5–0.9 points. What differs across spaces is not whether promotions are resolved, but how much evidence the search could have gathered per mechanism combination. Figure 7(c) shows the breadth channel directly: over candidates 44–87, one- and two-family runs discover 0.26 and 0.09 new mechanism combinations per candidate, whereas three- and six-family runs keep discovering 0.44 and 0.46, so r stays below two.

G.3 Theory: Evidence per Mechanism Combination

Proposition G.1 (Evidence per mechanism combination) *Let mechanism combination c be evaluated r_c times, with paired verdicts $\hat{\delta}_{c,1}, \ldots, \hat{\delta}_{c,r_c}$ against the incumbent. Assume that (i) all imple-*

mentations of c share an advantage Δ_c, and (ii) the verdicts are independent Gaussian with mean Δ_c and known variances $\sigma_{c,j}^2 = 10^4 d_{c,j}/n^2$. Define

$$\lambda_c = \Delta^2 \sum_{j=1}^{r_c} \sigma_{c,j}^{-2} = \frac{r_c \Delta^2}{\bar{\sigma}_c^2}, \qquad \bar{\sigma}_c^{-2} = \frac{1}{r_c} \sum_{j=1}^{r_c} \sigma_{c,j}^{-2}. \tag{11}$$

Then every level-η test of the null $\Delta_c = 0$ against the alternative $\Delta_c = \Delta > 0$ has power at most

$$\Phi\big(\sqrt{\lambda_c} - z_{1-\eta}\big),$$

which is $\Phi(\sqrt{\lambda_c} - 1.96)$ for $\eta = 0.025$. If a budget of B evaluations visits $M(B)$ distinct mechanism combinations, the mean evidence per mechanism combination is

$$\bar{\lambda} = \underbrace{\frac{B}{M(B)}}_{\textit{breadth}} \; \Delta^2 \; \underbrace{\overline{\sigma^{-2}}}_{\textit{coupling}}, \tag{12}$$

where $\overline{\sigma^{-2}}$ is the mean inverse verdict variance over the B evaluations.

Proof. Under (i)–(ii), the log-likelihood ratio of the alternative to the null is increasing in the precision-weighted statistic

$$Z = \frac{\sum_j \hat{\delta}_{c,j}/\sigma_{c,j}^2}{\big(\sum_j \sigma_{c,j}^{-2}\big)^{1/2}}, \qquad Z \sim \mathcal{N}(0,1) \text{ under the null}, \qquad Z \sim \mathcal{N}\big(\sqrt{\lambda_c}, 1\big) \text{ under the alternative.}$$

By the Neyman–Pearson lemma, the most powerful level-η test rejects when $Z > z_{1-\eta}$, with power $\Pr(\mathcal{N}(\sqrt{\lambda_c}, 1) > z_{1-\eta}) = \Phi(\sqrt{\lambda_c} - z_{1-\eta})$.

For Eq. (12), sum over mechanism combinations and divide by $M(B)$:

$$\sum_c \lambda_c = \Delta^2 \sum_{j=1}^{B} \sigma_j^{-2} \quad \Longrightarrow \quad \bar{\lambda} = \frac{1}{M(B)} \sum_c \lambda_c = \frac{B}{M(B)} \, \Delta^2 \, \overline{\sigma^{-2}}. \qquad \square$$

Remarks. The bound is optimistic, so the reported powers are upper bounds: heterogeneous implementations add variance τ^2 to each verdict, verdicts sharing questions and incumbents are positively dependent, and the loop acts on single comparisons ($r_c = 1$) rather than pooling. The proposition is the per-combination analogue of fixed-budget best-arm identification, where error grows with the number of arms sharing a budget (Audibert et al., 2010). Breadth also compounds: if proposals draw mechanism combinations from p over $\mathcal{C}$, the expected number of distinct mechanism combinations after N draws is $\sum_c [1 - (1 - p_c)^N]$, which approaches N (so $r \to 1$) when the proposer spreads over many more mechanism combinations than the budget can revisit, as happens once mechanism combinations span many families.

Estimates. For each run we set $r = 87/M(87)$ and take $\bar{\sigma}$ as the mean paired standard error over its first 87 candidates, with $\Delta = 1$ point (above the mean promotion size and the 0.65-point held-out value of doubling the frame budget; Appendix H). From one to six families,

$$r: \; 3.54 \to 1.73 \; (2.05\times \text{ fewer}), \qquad \bar{\sigma}^2: \; 2.24\times \text{ larger}, \qquad \lambda: \; 3.38 \to 0.76, \qquad \text{power}: \; 0.45 \to 0.14,$$

with 0.44 and 0.26 for two and three families. **Power decreases with the number of families** (Spearman $\rho = -0.74$, $p = 0.02$ over the nine nested runs; $\rho = -0.86$, $p < 0.001$ over all eleven), and the three full-harness runs have the three lowest values (exact $p = 1/84$). The ordering is unchanged when σ is summarized by its median, by the noise at the mean footprint, or by precision-weighting only verdicts with $d \geq 4.5$. The argument does not explain the slower validation progress of the two-family space, which is within seed variation.

G.4 Behavioral Footprint of Edits

Proposition G.2 (Footprint law) *Suppose an edit flips d validation answers, each independently a fix with probability q and a break otherwise. With $F \sim \text{Binomial}(d, q)$ fixes, the verdict $\hat{\delta} = 100\,(2F - d)/n$ satisfies*

$$\mathbb{E}[\hat{\delta}] = \frac{100\,d\,(2q-1)}{n}, \qquad \text{Var}[\hat{\delta}] = \frac{4 \cdot 10^4\, d\, q(1-q)}{n^2} \leq \frac{10^4\, d}{n^2}. \tag{13}$$

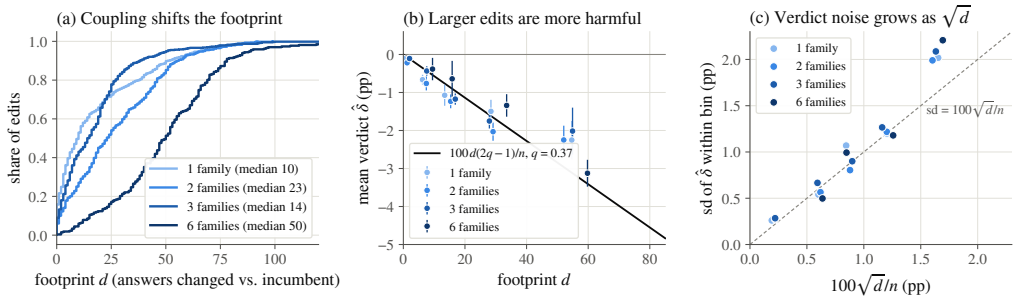


Figure 8: **The behavioral footprint of edits.** (a) Distribution of the footprint d, the number of validation answers whose correctness an edit flips relative to its incumbent. (b) Mean verdict by footprint bin (95% intervals); the line is Proposition G.2 with the pooled fix probability $q = 0.37$. Bins above 80 are dominated by crashes and omitted. (c) Standard deviation of verdicts within each bin against the paired noise scale $100\sqrt{d}/n$.

Hence, when $q < 1/2$, the expected effect decreases linearly in d while the noise grows as $\sqrt{d}$.

Proof. Substitute $\mathbb{E}[F] = dq$ and $\mathrm{Var}[F] = dq(1-q)$ into $\hat{\delta} = 100(2F - d)/n$; the inequality uses $q(1-q) \le 1/4$. □

Coupling enlarges the footprint. The median footprint is 50 answers in the full harness against 10, 23, and 14 for one, two, and three families (Figure 8a). This is not because full-harness proposals declare more changes: edits touching at most one family still flip 47.6 answers on average, against 19.8 with retrieval alone. **In a coupled harness a local change alters what every downstream stage receives**—retrieved frames, tool outputs, reflection prompts, and memory—so its effect is not local.

Larger footprints are more harmful and noisier. The fix probability is nearly constant across spaces ($q = 0.36$–0.39; pooled 0.37 $[0.36, 0.39]$). Because $q < 1/2$ near a strong incumbent, the mean verdict falls roughly linearly in d, as Proposition G.2 predicts (Figure 8b; $R^2 = 0.39$), and full-harness edits are the most harmful (-3.13 points, against -0.90, -1.53, and -1.02). Within-bin dispersion follows $100\sqrt{d}/n$ up to $d = 40$ and exceeds it beyond (panel c). This mirrors pleiotropy in Fisher's geometric model, where mutations affecting more traits are less likely to help (Fisher, 1930; Orr, 2000); here, coupling sets the pleiotropic reach.

H The Early-Experience Trap

This section supports Section 2.2.3. Its conclusions are:

- **Verdicts are weak and biased.** On Video-MME only 18% of verdict variance reflects true differences (8% in iterations 1–10), and the champion's cached score is inflated by $\widehat{\omega} \approx 1.0$ point (Appendix H.2).
- **The policy ignores this.** The held-out win rate changes gradually with the verdict, yet the loop continues winners with probability 0.77 and non-winners with 0.076, and treats false losers like true losers (Appendix H.3).
- **Theory.** Because rejected directions are rarely resampled, a weak beneficial direction is more likely lost than kept whenever $\omega > 0$; for the frame budget the loss probability is 0.54, matching the 6 of 12 runs that never expand it (Appendix H.4).
- **Consequences.** Seeds of one setting diverge $2.8\times$ more than independent sampling predicts, and runs that stay at 16 frames lose 2.7 held-out points (Appendices H.5 and H.6).

Table 15: **The 44 VIDEOHARNESS-BRSI runs analyzed in Section 2.2.3.** Video-MME runs search one or a few mechanism families for 50 iterations of three candidates; the 12 retrieval-family runs (all but self-reflection) form the frame-budget contrast of Figure 3(c). LongVideoBench runs search the full harness for 30 iterations under the feedback arms of Section 2.2.1, three seeds at each of three validation fractions.

Benchmark	Setting	Runs
Video-MME	retrieval only (16-frame seeds; one from 32-frame AKS)	6
	retrieval + frame selection	4
	retrieval + perception tools (one with 100×5)	2
	self-reflection	2
LongVideoBench	Blindfolded / Trace / Multimodal feedback	$9+9+9$
	clade-UCB parent suggestions (33% ladder)	3

H.1 CORPUS AND DEFINITIONS

We analyze 44 VIDEOHARNESS-BRSI runs (Table 15): 14 on Video-MME with 2,229 candidates and 30 on LongVideoBench with 2,577. Held-out evaluations, never shown to the proposer, exist for every candidate of three Video-MME runs and for frontier or best candidates elsewhere.

Each candidate c is compared with the incumbent, the validation champion when c is proposed, by its verdict $\hat{\delta}_c$ and $z_c = \hat{\delta}_c / \widehat{\mathrm{SE}}_c$ with $\widehat{\mathrm{SE}}_c = 100\sqrt{d_c}/n$ (Appendix E); δ_c^{test} is the held-out analogue. A candidate is *continued* if a later candidate names it as parent, and is a *false loser* if $\hat{\delta}_c \le 0$ but $\delta_c^{\mathrm{test}} \ge 0$. A retrieval-family run *expands the frame budget* if its deployed champion uses more than $1.2\times$ the 16-frame baseline. A within-run regression of held-out accuracy on mechanism families, iteration, and $\log_2$ frames over 491 tested candidates values one doubling of the frame budget at $\widehat{\mu}_f = 0.65 \pm 0.10$ pp (observational).

H.2 A VERDICT MODEL WITH INCUMBENT OPTIMISM

For a candidate–incumbent pair evaluated on both splits, Eq. (1) specializes to

$$\hat{\delta} = \Delta - \omega + u + \varepsilon, \qquad \delta^{\mathrm{test}} = \Delta + \varepsilon', \qquad \Delta \sim \mathcal{N}(m, \tau^2), \tag{14}$$

with independent sampling errors $\varepsilon, \varepsilon'$ of variances σ^2, σ_t^2, non-transferable fit u (the term S of Eq. (1)) of variance s^2, and incumbent optimism $\omega \ge 0$ from the optimizer's curse (Smith & Winkler, 2006). We take u zero-mean, so $\widehat{\omega}$ below estimates the incumbent's net optimism $\omega - \mathbb{E}\,S$, the total amount by which a verdict understates Δ on average; the two coincide when candidate and incumbent carry equal fit on average. The method-of-moments estimators are

$$\begin{aligned} &\widehat{m} = \overline{\delta^{\mathrm{test}}}, \quad \widehat{\omega} = \overline{\delta^{\mathrm{test}}} - \overline{\hat{\delta}}, && \widehat{\tau}^2 = \widehat{\mathrm{Cov}}(\hat{\delta}, \delta^{\mathrm{test}}), \\ &\widehat{\sigma}^2 = \overline{\widehat{\mathrm{SE}}^2}, \quad \widehat{s}^2 = \widehat{\mathrm{Var}}(\hat{\delta}) - \widehat{\tau}^2 - \widehat{\sigma}^2. \end{aligned} \tag{15}$$

Proposition H.1 (Evidential value of a verdict) *Under Eq. (14), let $\kappa = \tau^2/(\tau^2 + \sigma^2 + s^2)$ and $\mu(\hat{\delta}) = m + \kappa\,(\hat{\delta} + \omega - m)$. Then*

$$\Pr(\Delta > 0 \mid \hat{\delta}) = \Phi\left(\frac{\mu(\hat{\delta})}{\tau\sqrt{1-\kappa}}\right), \qquad \Pr(\delta^{\mathrm{test}} > 0 \mid \hat{\delta}) = \Phi\left(\frac{\mu(\hat{\delta})}{\sqrt{\tau^2(1-\kappa) + \sigma_t^2}}\right).$$

Both equal $1/2$ at $\hat{\delta}^* = m - \omega - m/\kappa$ ($= -\omega$ when $m = 0$), and their slopes in $\hat{\delta}$ are proportional to the verdict's reliability κ.

Proof. $(\Delta, \hat{\delta}, \delta^{\mathrm{test}})$ is jointly Gaussian with

$$\mathrm{Cov}(\Delta, \hat{\delta}) = \mathrm{Cov}(\delta^{\mathrm{test}}, \hat{\delta}) = \tau^2, \qquad \mathrm{Var}(\hat{\delta}) = \tau^2 + \sigma^2 + s^2.$$

Gaussian conditioning gives

$$\mathbb{E}[\Delta \mid \hat{\delta}] = \mathbb{E}[\delta^{\mathrm{test}} \mid \hat{\delta}] = \mu(\hat{\delta}), \qquad \mathrm{Var}(\Delta \mid \hat{\delta}) = \tau^2(1-\kappa), \qquad \mathrm{Var}(\delta^{\mathrm{test}} \mid \hat{\delta}) = \tau^2(1-\kappa) + \sigma_t^2,$$

and setting $\mu(\hat{\delta}) = 0$ yields $\hat{\delta}^*$. □

Table 16: **Verdict-model estimates** (Eq. (15); pp). Reliability is $\tau^2/(\tau^2+\sigma^2)$, i.e., κ of Proposition H.1 with $s^2 = 0$. Brackets: 95% run-bootstrap intervals.

Subset	Pairs	$\widehat{\tau}$	$\widehat{\sigma}$	$\widehat{\omega}$	Reliability
Video-MME, all	387	0.49 [0.31, 0.58]	1.04 [0.90, 1.23]	1.02 [0.64, 1.39]	0.18 [0.09, 0.24]
iterations 1–10	86	0.45 [0.18, 0.66]	1.49 [1.40, 1.59]	1.01 [0.69, 1.32]	0.08
iterations >10	301	0.48	0.87	1.02	0.24
fully tested runs	296	0.46	1.00	1.11	0.18
LongVideoBench, all	158	1.01 [0.72, 1.22]	1.06 [0.94, 1.16]	0.44 [0.21, 0.70]	0.48 [0.32, 0.57]

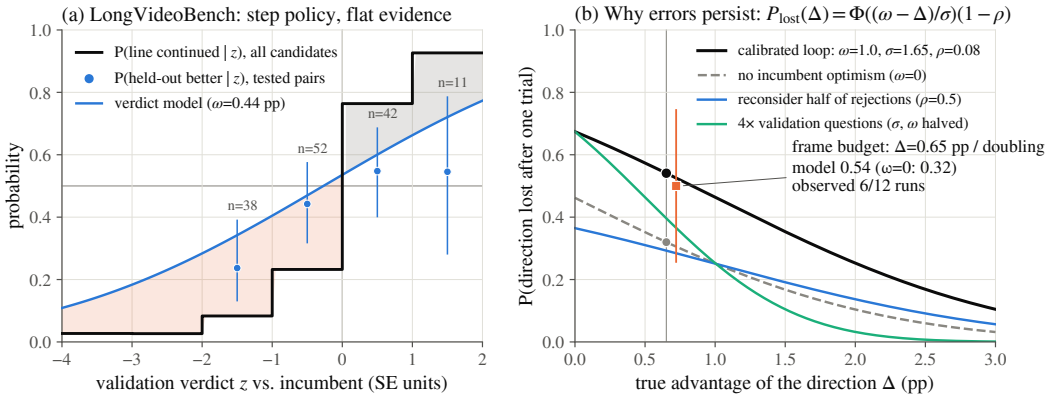


Figure 9: **Theory of the trap.** (a) LongVideoBench counterpart of Figure 3(f): the continuation policy is a step at $z = 0$, while the held-out win rate of the same verdicts changes gradually; the curve is Proposition H.1 with the fitted parameters. (b) Probability that a beneficial direction is lost after one trial (Proposition H.2) at the Video-MME calibration, without incumbent optimism, with a policy that reconsiders half of all rejections, and with four times as many validation questions (which halves σ and, being selection noise, ω). Points mark the frame-budget direction; the square is the observed fraction of retrieval-family runs that never expand the budget (Wilson 95% interval).

Implications. **A verdict of zero is not neutral**: with $\omega > 0$, a challenger that ties the incumbent is more likely than not to be better, and it must lose by about ω before it becomes a coin flip. When κ is small, the rational probability of continuing a line changes only gradually with $\hat{\delta}$.

Estimates. Table 16 reports run-bootstrap estimates after dropping clones and execution failures. On Video-MME the reliability is only 0.18, and lowest early (0.08 versus 0.24 after iteration 10) because early edits change more answers. The optimism $\widehat{\omega} \approx 1.0$ pp is stable over time and in the fully tested runs. LongVideoBench verdicts are more reliable (0.48) and less biased ($\widehat{\omega} = 0.44$), but the policy is the same.

H.3 Continuation Policy versus Evidential Value

Binning verdicts by z, Proposition H.1 tracks the held-out win rate (mean absolute deviation 0.035 on Video-MME, 0.085 on LongVideoBench), which crosses $1/2$ at $z \approx -0.75$ and -0.25 (Figure 9a). **The policy, by contrast, is nearly a step function at** $z = 0$: on Video-MME a one-question loss is continued with probability 0.070 and a one-question win with 0.685; winners overall with 0.77 and non-winners with $\rho = 0.076$ (odds ratio 41; LongVideoBench 34). False losers are abandoned as often as true losers (93% versus 96% on Video-MME), although 37% of tested non-winners beat their incumbent on held-out data (33% on LongVideoBench).

H.4 Why Early Errors Persist

Proposition H.2 (Persistence under endogenous sampling) *Consider a direction with true advantage Δ over the incumbent. Its first trial yields $\hat{\delta}_1 = \Delta - \omega + \varepsilon_1$ with $\varepsilon_1 \sim \mathcal{N}(0, \sigma^2)$. The*

loop continues the direction if $\hat{\delta}_1 > 0$; otherwise it reconsiders it with probability ρ, independently of Δ, and otherwise never evaluates it again. Then:

(a) the direction is lost with probability

$$P_{\text{lost}}(\Delta) = \Phi\left(\frac{\omega - \Delta}{\sigma}\right)(1-\rho);$$

(b) P_{lost} increases with ω, decreases with Δ and ρ, and increases with σ if and only if $\Delta > \omega$;

(c) as $\Delta/\sigma \to 0$ with ω/σ fixed, $P_{\text{lost}} \to \Phi(\omega/\sigma)(1-\rho) > (1-\rho)/2$ whenever $\omega > 0$, so a weak but beneficial direction is more likely lost than kept;

(d) after a rejection the loop's only estimate is $\hat{\delta}_1$, which is biased downward,

$$\mathbb{E}[\hat{\delta}_1 \mid \hat{\delta}_1 \le 0] = \Delta - \omega - \sigma\,\frac{\phi(a)}{\Phi(-a)} < \Delta, \qquad a = \frac{\Delta - \omega}{\sigma},$$

and is never corrected, whereas continued directions receive new, independent evidence.

Proof. (a) $\Pr(\hat{\delta}_1 \le 0) = \Phi((\omega - \Delta)/\sigma)$, and a rejected direction is lost unless independently reconsidered, with probability $1 - \rho$.

(b) Monotonicity in ω, Δ, and ρ follows from that of Φ; for σ,

$$\frac{\partial P_{\text{lost}}}{\partial \sigma} = -\,\phi\left(\frac{\omega - \Delta}{\sigma}\right)\frac{\omega - \Delta}{\sigma^2}\,(1-\rho),$$

which is positive if and only if $\Delta > \omega$.

(c) follows from (a), since $\Phi(x) > 1/2$ for $x > 0$. (d) is the mean of a Gaussian truncated above at zero. □

Interpretation. Part (d) is the hot-stove asymmetry of learning from self-selected experience (Denrell & March, 2001) and the incomplete learning of greedy allocation (Rothschild, 1974; Brezzi & Lai, 2000): negative errors persist because the alternative is never resampled. Persistent memory makes this worse, as labels such as "MINED OUT" set $\rho \approx 0$ for an entire family.

Calibration for the frame budget. With $\Delta = \widehat{\mu}_f = 0.65$ pp, $\sigma = 1.65$ pp (median early paired scale), $\omega = 1.01$ pp (Table 16), and $\rho = 0.076$,

$\Pr(\text{reject}) = 0.585$, $\quad \mathbf{P}_{\text{lost}} = \mathbf{0.54}$ $[0.39,\ 0.63]$ over plausible (ω, σ, ρ), $\quad P_{\text{lost}}\big|_{\omega=0} = 0.32$.

Six of the 12 retrieval-family runs never expand the budget (0.50; Wilson $[0.25, 0.75]$), consistent with the model, though too few to reject $\omega = 0$. Figure 9(b) compares remedies: reconsidering half of all rejections lowers P_{lost} to 0.29, quadrupling validation questions to 0.40, and judging a probe against its own parent removes ω directly.

H.5 Path Dependence across Seeds

If early noisy verdicts steer exploration, seeds of one setting should diverge more than independent samples would. Dividing the cross-seed standard deviation of each family's candidate share by its binomial value, **the median ratio is 2.8 on Video-MME and 1.7 on LongVideoBench**, the signature of reinforcement processes that amplify early outcomes (Arthur, 1989); re-ranking, for example, takes 93% of one seed's candidates but 2–11% in the other four. The held-out cost of staying at the baseline budget (64.9 versus 67.6%, $p = 5 \times 10^{-5}$) holds under alternative definitions of "stays" and without the AKS-seeded run ($p \le 0.003$).

H.6 Case Studies

A weak loss becomes a prohibition, or not. In retrieval-only seed 1, a 32-frame probe loses by 10 questions at iteration 8 ($z = -1.41$); memory then records frame count as "MINED OUT—do NOT re-propose", and no later candidate exceeds 16 frames. In seed 4, a 33.9-frame probe loses by 12 questions ($z = -1.52$), but a sibling at 18.3 frames wins; memory keeps the direction open, and the champion reaches 25–30 frames and 67.3% held-out, against 64.8% for seed 1. A Bayes-rational policy (Proposition H.1) would still give seed 1's rejected direction probability ≈ 0.29.

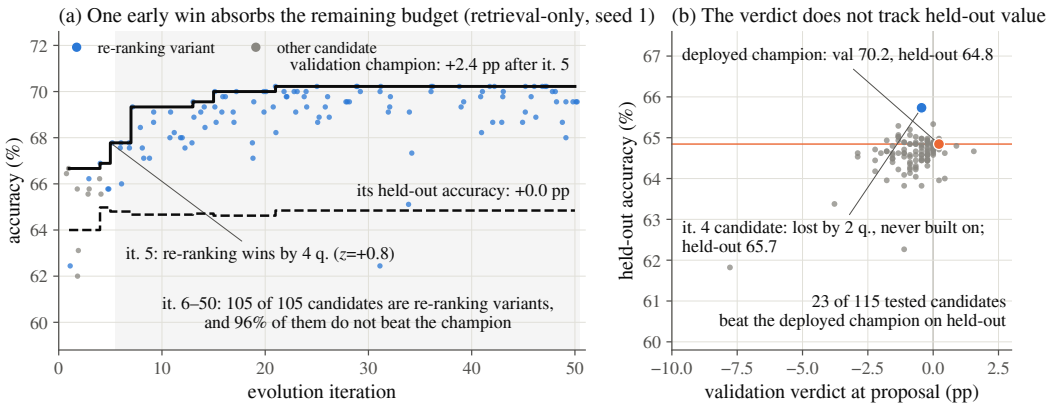


Figure 10: **Over-commitment to an early winner** (Video-MME, retrieval-only, seed 1; the run of Figure 3(c)). (a) Validation accuracy of every candidate, the validation champion (solid), and the champion's held-out accuracy (dashed); iterations 6–50 are shaded. (b) Held-out accuracy of the 115 tested candidates against their validation verdict at proposal. The horizontal line marks the deployed champion's held-out accuracy.

Over-commitment to an early winner. In the same seed 1, after a re-ranking candidate wins by four questions at iteration 5 ($z = +0.76$), all 105 later candidates are re-ranking variants and 96% fail to beat the champion (Figure 10a). Four promotions add 2.4 validation points but only 0.04 held-out points, and 23 of 115 tested candidates beat the deployed champion on held-out data (panel b).

I Why Simple Remedies Are Insufficient

This section supports the discussion of simple remedies in Section 2.2. Its conclusions are:

- **Parent selection.** Changing which harness the proposer builds on reshapes the search as intended, but does not improve held-out accuracy (Appendix I.1).
- **Stronger initialization.** Starting from AKS speeds up early validation progress, but the final harness is no better than those evolved from simple seeds (Appendix I.2).

Neither remedy changes the verdicts on which the loop acts, so neither removes the failure modes.

I.1 Parent Selection

Setting. We run 18 full-harness VideoHarness-BRSI runs (1,556 candidates, 30 iterations of three) on Video-MME and LongVideoBench. Within a benchmark, arms differ only in the list of suggested parents shown to the proposer each iteration: none (**plain loop**); the current champion (**greedy**); six names ranked by an upper confidence bound over the lineage tree, the best score in a subtree plus an exploration bonus $\gamma/\sqrt{\text{subtree size}}$ with $\gamma = 1.5$ (**clade-UCB**); or six names drawn uniformly from the archive (**random**). Video-MME runs all four arms and LongVideoBench compares the plain loop with clade-UCB, three seeds each. The Video-MME plain-loop runs are the full-harness runs of the search-space ladder.

Results. No parent rule yields a consistent held-out advantage (Figure 11a). On Video-MME the deployed harnesses reach 65.7, 66.9, 67.2, and 67.4% for the plain loop, greedy, clade-UCB, and random parents; among the three rules every pairwise difference is at most 0.5 points ($p \geq 0.74$, Welch), and all seed ranges overlap. On LongVideoBench, clade-UCB trails the plain loop by 1.01 ± 1.17 points over seed-paired per-question comparisons ($p = 0.48$). The rules do not separate on validation either (Figure 11b): clade-UCB trails the plain loop on LongVideoBench at every matched budget.

The manipulation worked. The proposer builds on a listed parent in 94–100% of clade-UCB, 74–86% of greedy, and 80–86% of random-parent proposals. On Video-MME the share of candidates

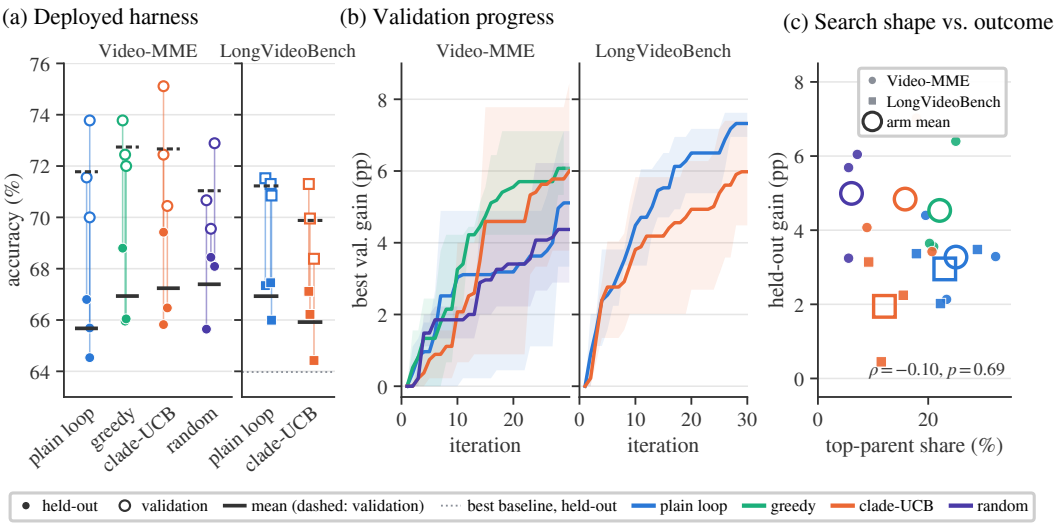


Figure 11: **Parent selection.** (a) Validation (hollow) and held-out (filled) accuracy of each run's deployed harness; bars mark arm means. (b) Best-so-far validation gain over the best seeded harness (mean and seed range). (c) Share of candidates bred from the most-used parent against held-out gain over the best baseline; large outlined markers are arm means.

bred from the champion falls from 47% (greedy) to 30% (clade-UCB) and 8% (random), while distinct parents rise from 19 to 32 and 47; on LongVideoBench clade-UCB halves the spread of validation scores (3.6 versus 6.6 points). Yet **no search-shape statistic predicts held-out gain** across the 18 runs (top-parent share $\rho = -0.10$, $p = 0.69$; distinct parents $\rho = -0.01$, $p = 0.98$; Figure 11c). What does is how quickly validation rose (candidates to 90% of the final gain, $\rho = -0.53$, $p = 0.025$): the quality of the harnesses the proposer happened to write, not the rule that chose their parents.

Interpretation. A parent rule decides which lineage the next verdict concerns, not how informative that verdict is. Every arm still compares single candidates against a selected champion on the same validation questions, so S, ω, and ε in Eq. (1) are unchanged and the evidence per mechanism combination cannot rise. This motivates organizing evolution around mechanisms, where evidence can be concentrated, rather than lineages. Since no rule differs significantly from the others in this matched comparison, the loops inside VIDEOHARNESS-SRSI, such as the warm-up, keep the default of leaving the parent choice to the proposer, and we do not repeat the parent-selection comparison for VIDEOHARNESS-SRSI.

I.2 STRONGER INITIALIZATION

Setting. We run seven Video-MME runs (924 candidates, 50 iterations of three). Six start from simple 16-frame seeds (three evolve retrieval alone, three retrieval with frame selection); the seventh evolves the retrieval-only space from Adaptive Keyframe Sampling (AKS) (Tang et al., 2025), which allocates 32 frames by CLIP relevance. AKS starts at 68.7% validation and 67.1% held-out accuracy, 3.4 and 4.0 points above the simple seeds.

Results. A stronger start accelerates early progress but not the endpoint (Figure 12). The AKS run reaches 70% validation accuracy at iteration 2 and 72% at iteration 7, against 4–15 and 21–45 iterations for retrieval-only runs from simple seeds, which pass the AKS starting level within 4–18 iterations. Its deployed harness (74.0% validation, 68.2% held-out) lies inside the range of the six other runs (69.6–74.2% and 64.8–68.3%) and is statistically tied with the three best (paired differences -0.1, $+0.8$, $+1.5$ points; $p \geq 0.06$). Evolution adds only 1.1 points to AKS ($p = 0.13$), against 1.4–5.8 points for simple seeds, four of six significantly, while using 36 frames against 14–25.

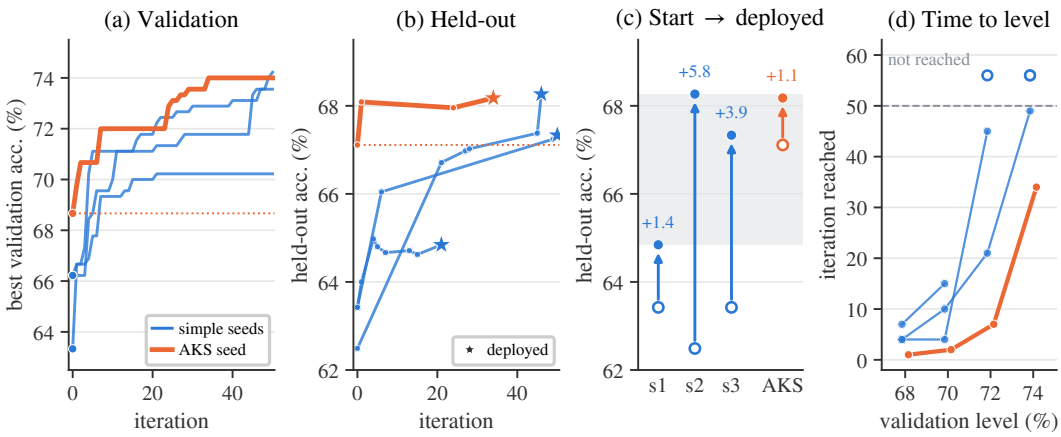


Figure 12: **Stronger initialization** (Video-MME, retrieval only; blue: three runs from simple seeds, orange: the AKS-seeded run). (a) Best-so-far validation accuracy. (b) Held-out accuracy of tested running champions; stars mark deployed harnesses. (c) Held-out accuracy of the starting point (hollow) and deployed harness (filled); shading spans the simple-seed endpoints. (d) Iteration at which each validation level is first reached; hollow markers are levels not reached in 50 iterations.

Interpretation. A stronger start mainly shortens the search to competitive performance; with one AKS run and a larger initial frame budget, it does not establish a reproducible final gain. It does suggest a way around the early-experience trap: supplying a useful mechanism before chance validation wins shape exploration, which motivates the instrumented warm-up of VIDEOHARNESS-SRSI.

J IMPLEMENTATION AND STATISTICAL DETAILS OF VIDEOHARNESS-SRSI

This section specifies what Section 3 leaves out. Each seed runs *WARM* (Stage I), *NOMINATE*, *PROBE*, and *RACE* (Stage II), and *COMPOSE* and *MERGE* (Stage III), after the six seeded harnesses are scored into a shared executor cache (Algorithm 2; settings in Table 17). All comparisons are paired verdicts $\hat{\delta}(H, H')$ over questions scored for both harnesses, with variance given by Eq. (7); on MomentSeeker they use the indicator $\mathrm{IoU} \geq 0.5$, while the loop and final selection use mIoU.

J.1 INFORMATION ISOLATION

WARM, PROBE, RACE, and MERGE run the loop of VIDEOHARNESS-BRSI (Appendix C.1) in separate workspaces; NOMINATE and COMPOSE are single proposer calls. Each per-question trace holds opaque identifiers for pairing, the prediction and correctness, and counters of executor and tool calls, frames, rendering, personas, memory operations, and exercised mechanism families; it never contains question or option text, frames, or gold answers. $\mathbf{u}(H)$ in Eq. (2) collects per-question means of these counters.

A deny list enforced on all file, search, and shell tools confines each proposer call to its own workspace and run, and keeps datasets, caches, held-out outputs, and evaluation code out of reach. Warm-up evidence reaches a branch only through its charter and the pool members copied into it; the merge workspace regains read access to earlier workspaces only after branch effects are measured, and no phase before MERGE evaluates the held-out split. (In the LongVideoBench and OVO-Bench runs, a monitor also evaluated validation-front members on the held-out split during search; these results were unreadable by the proposer and entered no decision.) Isolation withholds content, not outcomes: because records pair predictions with correctness, a proposer can reconstruct the answer key for questions some harness answers correctly (one Video-MME run did so for 402 of 450), but harness code cannot use it, since the leak guard bars access to question identity.

Algorithm 2 VIDEOHARNESS-SRSI as implemented

Require: Frozen VLM M, proposer P, validation set $\mathcal{S}_{\text{val}}$, seed archive $\mathcal{D}_0$, vocabulary $\mathcal{V}$, budgets $r_1 < r_2$

Stage I: Warm-up and seed selection (setup, prewarm, WARM)

Grow $\mathcal{D}^{\text{warm}} \supseteq \mathcal{D}_0$ by rounds of three free, one removal, and one intensity candidate until the plateau rule holds

$\mathcal{B} \leftarrow$ six best evolved harnesses in $\mathcal{D}^{\text{warm}}$; $H_0 \leftarrow \arg\max_{H \in \mathcal{B}} V(H)$; compute the warm-up report $\{\mathcal{R}_\alpha^{\text{warm}}\}$

Stage II: Isolated branch evolution (NOMINATE, PROBE, RACE)

$\mathcal{A} \leftarrow$ ladder-screened templates of $\mathcal{V}$ and a free-form axis; P writes charters $\{c_\alpha\}$

for all $\alpha \in \mathcal{A}$ **do**

  Evolve r_1 candidates in an isolated branch that holds $\mathcal{B}$, with parents H_0 or earlier branch candidates; $m_\alpha \leftarrow \max_{H \in \mathcal{D}_\alpha^{\text{probe}}} \hat{\delta}(H, H_0)$

$\mathcal{A}_{\text{race}} \leftarrow$ the k_{r} axes with the largest m_α, plus at most one tied axis

for all $\alpha \in \mathcal{A}_{\text{race}}$ **do**

  Continue the branch to r_2 candidates in total; estimate $(\bar{\delta}_\alpha, \hat{s}_\alpha)$ by Eq. (17)

$\mathcal{A}_{\text{adm}} \leftarrow$ axes with $\bar{\delta}_\alpha > 1.64\,\hat{s}_\alpha$ and $\bar{\delta}_\alpha > 0.5$, in decreasing order of $\bar{\delta}_\alpha$

$\mathcal{A}_{\text{comp}} \leftarrow$ at most three axes of $\mathcal{A}_{\text{race}}$, admitted axes first, then by $\bar{\delta}_\alpha$

Stage III: Factorial composition and joint refinement (COMPOSE, MERGE)

P fixes one modification d_α per $\alpha \in \mathcal{A}_{\text{comp}}$ and builds every cell $H_{\mathcal{U}}$ (Eq. (3))

Score the cells; record effects, interactions (Eq. (18)), and the gate-eligible outcome

Refine jointly for 10 rounds of three candidates, starting from a ledger that holds every $H_{\mathcal{U}}$

return $H^\star \leftarrow$ validation arg max of the refinement ledger; only now evaluate on held-out data

J.2 STAGE I: INSTRUMENTED WARM-UP

Mechanism families. VIDEOHARNESS-SRSI groups the harness primitives into seven families, the units to which removals, intensity changes, traces, and axis templates refer (Prompts 1–4): `retrieve` (frame selection and query-conditioned retrieval), `multiturn` (multi-pass reading and self-reflection), `roles` (multi-role collaboration), `perceive` (detection and localization tools), `render` (pixel re-presentation such as zoom, crop, and montage), `analyze` (shot, motion, and similarity signals), and `memory` (working memory). The search-space ladder of Appendix G instead counts the families chartered in the task description of VIDEOHARNESS-BRSI, which uses a different granularity (e.g., it separates retrieval from frame selection); its family counts are not in these units.

Slots and audit. Each round evaluates five candidates in three slots: Slot A makes three unrestricted *free* proposals; Slot B makes one *removal* that deletes the champion's most-used family not yet removed; and Slot C makes one *intensity* proposal that changes one family's usage by about a factor of two. Each candidate's evidence $\mathbf{w}(H)$ follows Eq. (2). Requests are audited from traces rather than trusted: a candidate is a removal if its family set is a strict subset of its parent's, an intensity change if the sets agree and exactly one nonzero coordinate of $\mathbf{u}$ moves by at least 25%, and free otherwise. Candidates with empty predictions on more than 5% of questions count as crashed.

Stopping and backbone. WARM stops once $t \geq 10$ rounds, or $t \geq 8$ with no improvement in the last four. The seed pool $\mathcal{B}$ holds the six best evolved warm-up candidates, preferably at most two per parent, and the backbone H_0 is the best of them.

Warm-up report. NOMINATE receives, per family: (i) *importance*, the mean verdict of audited removals against their parents; (ii) *dose response*, the Spearman correlation between usage and accuracy (observational, not used for ranking); (iii) usage share and oracle-union gain; (iv) a *baseline ladder* of paired differences between seeded harnesses with nested family sets, which measures value from a zero floor; and the frame range and median paired standard error. Blocks (i)–(iii) form $\mathcal{R}_\alpha^{\text{warm}}$. Evidence that cannot be computed is shown as missing rather than zero, which marks under-explored mechanisms without removing any axis. Averaged per removed family, removal contrasts measure each family's contribution; the champion's inflation ω shifts them all alike and so does not distort how families compare. Intensity evidence is observational, so the report uses it to flag directions, not to rank axes.

J.3 Stage II: Nomination, Probe, and Race

Axes and branches. Six templates—*retrieval*, *frame budget*, *read schedule*, *multi-role*, *pixels and detectors*, and *derived state*—fix each axis's permitted families and an on-axis test every branch edit must pass (e.g., a frame-budget edit must change mean frames by at least 20%, and other templates may not change them); one free-form axis has a proposer-written test. A template is pruned only if it touches no family of H_0 and all its ladder effects are negative, keeping at least four. The proposer charters each axis with a mechanism, its rationale, a first edit, a falsification criterion, and a predicted effect. Every branch holds the pool $\mathcal{B}$ and the seeded harnesses, but candidates must descend from H_0 or earlier branch candidates, so all branch verdicts share the backbone that the lattice requires; other pool members may only be mined for mechanisms.

Probe. Each axis produces $r_1 = 3$ candidates, and axes are ranked by their best probe verdict,

$$m_\alpha = \max_{H \in \mathcal{D}_\alpha^{\mathrm{probe}}} \hat{\delta}\big(H, H_0\big), \tag{16}$$

where $\mathcal{D}_\alpha^{\mathrm{probe}}$ is the archive of axis α after the probe. The $k_{\mathrm{r}} = \min\big(k, \max(2, \lceil k/2 \rceil)\big)$ of the k axes with the largest m_α continue (plus at most one tie). Since every axis is probed equally often against the same backbone, the backbone's inflation ω and the maximum's upward bias shift all axes alike and leave the ranking intact. We use the maximum rather than the mean because mechanisms that need compounding edits score near zero at first, and no standard error because probe candidates form one lineage cluster.

Race and admission. Survivors continue to $r_2 = 15$ candidates. With $\delta_h = \hat{\delta}(h, H_0)$ grouped by declared parent into lineage clusters $C_{\alpha,1}, \ldots, C_{\alpha,L_\alpha}$, the effect $\bar{\delta}_\alpha$ is the mean of δ_h, and its standard error comes from 2,000 cluster-bootstrap replicates,

$$\bar{\delta}_\alpha^{(b)} = \frac{\sum_{\ell=1}^{L_\alpha} \sum_{h \in C_{\alpha, J_\ell^{(b)}}} \delta_h}{\sum_{\ell=1}^{L_\alpha} \big|C_{\alpha, J_\ell^{(b)}}\big|}, \qquad \hat{s}_\alpha = \mathrm{sd}_b\big(\bar{\delta}_\alpha^{(b)}\big), \tag{17}$$

with each $J_\ell^{(b)}$ uniform on $\{1, \ldots, L_\alpha\}$, so that edits of one program do not count as independent evidence. An axis is **admitted** if $\bar{\delta}_\alpha > 1.64\, \hat{s}_\alpha$ and $\bar{\delta}_\alpha > 0.5$ pp; with fewer than three clusters it is recorded as unjudged. Because ω lowers every $\bar{\delta}_\alpha$, this test is conservative. The 1.64 multiplier is a one-sided normal reference, not a simultaneous guarantee. At most three axes enter composition, admitted axes first.

J.4 Stage III: Composition and Joint Refinement

Factorial lattice. COMPOSE distills each composed axis $\alpha \in \mathcal{A}_{\mathrm{comp}}$ into one modification d_α, the smallest edit of H_0 that carries the branch's mechanism, and writes all $2^{k_{\mathrm{c}}} - 1$ nonempty cells of Eq. (3), with each d_α byte-identical across cells. With three axes, each modification's main effect contrasts the four cells that contain it with the four that do not. With $x_{\mathcal{U}}(q)$ the correctness of $H_{\mathcal{U}}$ on question q ($x_\emptyset$ for H_0), each cell's effect is $\hat{\delta}(H_{\mathcal{U}}, H_0)$, and for $|\mathcal{U}| \geq 2$ the interaction is

$$I_{\mathcal{U}} = \frac{100}{|Q|} \sum_{q \in Q} \psi_{\mathcal{U}}(q), \qquad \psi_{\mathcal{U}}(q) = \sum_{\mathcal{W} \subseteq \mathcal{U}} (-1)^{|\mathcal{U}| - |\mathcal{W}|}\, x_{\mathcal{W}}(q), \tag{18}$$

over the questions Q shared by all cells, with standard error $100\, \mathrm{sd}_q\, \psi_{\mathcal{U}}(q)/\sqrt{|Q|}$; computing it per question retains the dependence through H_0. The pre-registered gate selects among H_0 and the cells of successive prefixes of the admission order (H_0 if no axis is admitted).

Joint refinement and final selection. MERGE runs 10 rounds of three candidates from a ledger holding all cells, so the proposer may build on any of them. $H^\star$ is the validation arg max of this ledger; only then are the seeded harnesses and the MERGE validation front evaluated once on the held-out split.

Table 17: **Settings of VIDEOHARNESS-SRSI.** The free-only warm-up control (Section 4.2) differs only in the warm-up slots (5 / 0 / 0). Here $k_{\mathrm{r}} = \min\big(k, \max(2, \lceil k/2 \rceil)\big)$ for k probed axes.

Setting	Value	Setting	Value
Warm-up slots A / B / C per round	3 / 1 / 1	Probe survivors (tie slack)	k_{r} (+1)
Warm-up chunk; min. / max. rounds; patience	2; 8 / 10; 4	Gate z_0 / floor (pp) / min. clusters	1.64 / 0.5 / 3
Class-1 / free-form templates; min. kept	6 / 1; 4	Composed axes (cells)	≤ 3 (≤ 7)
Branch candidates per round; r_1; r_2	3; 3; 15	MERGE rounds $\times$ candidates	10×3

J.5 BUDGET AND CONFIGURATION

With T warm-up rounds, k, k', and k_{c} probed, raced, and composed axes, and $R = 10$ MERGE rounds, a seed evaluates

$$B_{\mathrm{total}} = 5T + k\,r_1 + k'(r_2 - r_1) + \big(2^{k_{\mathrm{c}}} - 1\big) + 3R \tag{19}$$

candidates, excluding re-scored seeded harnesses. With $T \leq 10$, $k \leq 7$ (hence $k_{\mathrm{r}} \leq 4$), $k' = k_{\mathrm{r}}$, and $k_{\mathrm{c}} \leq 3$, this is at most $50 + 21 + 48 + 7 + 30 = 156$ candidates per seed. We report this as the budget bound; the tie slack of the probe cut-off can add one raced axis ($r_2 - r_1 = 12$ candidates) only when two axes tie exactly at the cut-off. The three Video-MME runs evaluated 138–141.

K ABLATION: INSTRUMENTED WARM-UP

This section details the warm-up ablation of Section 4.2. Its conclusions are:

- **Same validation, better held-out.** Both arms reach the same validation accuracy, yet every full-arm run beats every free-only run on held-out data, for the warm-up backbone ($+1.85$ pp) and the deployed system ($+2.8$ pp).
- **More controlled evidence.** The full arm yields $2.7\times$ as many removal contrasts and far fewer undefined family estimates.
- **Attribution.** A within-run replay attributes 28% of the backbone gap to selecting structured candidates; the rest is consistent with slot results informing free proposals.

K.1 PROTOCOL

Both arms run the complete VIDEOHARNESS-SRSI pipeline on Video-MME, three runs each. Per warm-up round, the *full* arm proposes three free candidates (Slot A), one family removal (Slot B), and one intensity change (Slot C); the *free-only* arm spends the same five evaluations on free proposals. Everything else, including seeds, splits, caches, isolation, stopping rule, and later stages, is shared. Because slots are instructed rather than enforced, we classify every warm-up candidate from its trace with the audit of Appendix J.2; counts below are behavioral. Arms are compared by exact one-sided permutation tests over the $\binom{6}{3} = 20$ assignments of runs, whose smallest attainable p-value, 0.05, requires complete separation; with three runs per arm, all results indicate direction rather than precise effect sizes.

K.2 RESULTS

Warm-up. Table 18 lists per-run results. The free-only arm climbs faster early (71.18 versus 69.26 after six rounds; Figure 13a), but the full arm catches up in rounds 7–9, and the arms end at 71.70 and 71.85. In the full arm, 8 of 11 champion replacements are structured (3 removals, 5 intensity changes), against 5 of 13 in the free-only arm, and all three full-arm backbones are intensity changes with structured parents. Deliberate contrasts are also cleaner: frames move by more than 10% in 23% of the full arm's removals, versus 77% of the structured candidates that arise unrequested in the free-only arm.

Evidence. The full arm yields 11.7 versus 4.3 removal contrasts per run ($p = 0.05$) and 13.0 versus 9.3 intensity contrasts ($p = 0.15$), giving 14 importance estimates covering six of the seven families versus 8 covering five, and leaving 5 versus 11 of the 21 family–run coverage increments (seven

Table 18: **Per-run results of the warm-up ablation** (Video-MME). Slots: behavioral counts of warm-up candidates. Families with an importance estimate (Imp.) or an undefined coverage increment (Und.). Backbone: warm-up champion; deployed: validation argmax of joint refinement. Val, Test in %; Fr.: frames per validation question. †Stopped after round 8 (plateau rule).

	Slots			Families		Backbone			Deployed		
Run	A	B	C	Imp.	Und.	Val	Test	Fr.	Val	Test	Fr.
Full-1	27	11	12	6	2	73.56	69.87	50	77.11	71.07	110
Full-2	26	9	15	3	1	71.11	68.40	55	73.33	70.00	141
Full-3	23	15	12	5	2	70.44	68.58	38	74.67	69.16	66
Mean	25.3	11.7	13.0	4.7	1.7	71.70	68.95	48	75.04	70.07	106
Free-1	38	2	10	2	4	73.11	66.58	44	74.00	66.31	63
Free-2†	27	8	5	3	4	71.33	66.62	65	72.44	67.16	81
Free-3	34	3	13	3	3	71.11	68.09	59	74.89	68.40	86
Mean	33.0	4.3	9.3	2.7	3.7	71.85	67.10	56	73.78	67.29	77

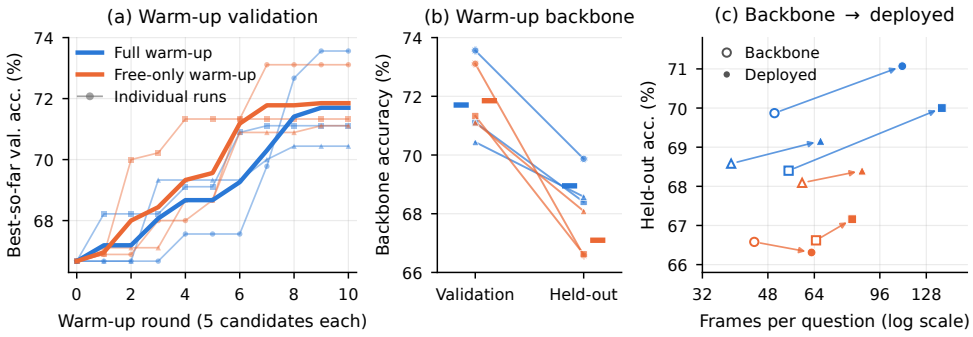


Figure 13: **Warm-up ablation on Video-MME** (thin lines, markers: runs; thick lines, bars: arm means). (a) Best-so-far validation accuracy (round 0: best seeded harness). (b) Warm-up backbones. (c) Held-out accuracy vs. frames per question, backbone (hollow) to deployed system (filled).

families × three runs) undefined ($p = 0.05$). Retrieval shows the difference most clearly: each full run removes it four times (mean -7.15 pp), whereas the free-only arm has a single retrieval-free candidate. Removals are not always frame-matched: the largest drop (-23.11 pp) also cut frames from 80 to 16.

Held-out contrasts. **Every full-arm backbone and deployed system beats every free-only one on held-out data**: 68.95 versus 67.10 for backbones and 70.07 versus 67.29 for deployed systems (both $p = 0.05$; Welch $t = 2.73$ and 3.40), while validation does not separate the arms ($p = 0.65$ and 0.25). Frames, which each harness chooses, are the main confound (Figure 13c), but regressing held-out accuracy on arm and $\log_2$ frames leaves arm effects of $+2.0$ and $+2.2$ pp ($t = 2.38$ and 2.55, three residual degrees of freedom).

Attribution. Because all full-arm backbones are structured, part of the gap could come from selecting structured candidates. Replaying each full run's selection on its free candidates only yields substitutes within 0.67 pp of the actual backbones on validation but lower on held-out data in all three runs, so **selection accounts for $+0.52$ pp, or 28% of the backbone gap**. The substitutes still average 1.33 pp above the free-only backbones. This remainder is consistent with slot results informing free proposals through the archive—8 of the full arm's 64 free proposals cite them, and the Full-3 backbone cites a removal result and a dose curve over retrieval calls—but three runs per arm cannot separate this channel from run-to-run variation (standard error about 0.7 pp).

Table 19: **Information-isolation ablation on LongVideoBench** (446 validation / 891 held-out questions, identical to Table 1). Accuracy (%) of each run's validation argmax, mean ± standard deviation over three seeds.

Setting	Val	Held-out
VIDEOHARNESS-BRSI feedback (Multimodal)	$71.00_{\pm 3.21}$	$65.43_{\pm 1.18}$
VIDEOHARNESS-BRSI + isolation (Blindfolded)	$71.22_{\pm 0.34}$	$66.93_{\pm 0.81}$
VIDEOHARNESS-SRSI	$71.1_{\pm 2.0}$	$\mathbf{67.3}_{\pm 1.0}$

L ABLATION: INFORMATION ISOLATION

This section separates the contribution of information isolation from that of the rest of VIDEOHARNESS-SRSI on LongVideoBench. Its conclusions are:

- **Isolation carries most of the gain.** Restricting the VIDEOHARNESS-BRSI proposer to content-free feedback raises held-out accuracy from 65.43 to 66.93%, about 80% of the way to VIDEOHARNESS-SRSI (67.3%), at the same validation accuracy.
- **Each step narrows the gap.** The validation–held-out gap falls from 5.57 (VIDEOHARNESS-BRSI feedback) to 4.29 (with isolation) and 3.8 points (VIDEOHARNESS-SRSI).

Protocol. We reuse the 33% ladder of the feedback study (Appendix F.1), whose validation and held-out questions are identical to those of the LongVideoBench partition in Table 1 (446 / 891 questions). Within this ladder, the Multimodal arm gives the proposer the full access of VIDEOHARNESS-BRSI, and the Blindfolded arm keeps the same loop but shows only per-question correctness and behavioral counters, the content-free feedback of VIDEOHARNESS-SRSI (Appendix J.1); it is thus VIDEOHARNESS-BRSI with information isolation and nothing else. Each arm runs three seeds of 30 iterations with three candidates, with perception tools disabled, and deploys each run's validation argmax. The VIDEOHARNESS-SRSI row is the LongVideoBench entry of Table 1. Two differences qualify the second step: VIDEOHARNESS-SRSI evaluates more candidates per seed (at most 156, Appendix J.5, versus 90), and the VIDEOHARNESS-BRSI row here is the 30-iteration Multimodal arm rather than the 50-iteration run of Table 1. The first step, from Multimodal to Blindfolded, changes only the proposer's feedback.

Results. As Table 19 shows, the three settings reach nearly the same validation accuracy (71.0–71.2%), so the differences lie entirely in transfer. Withholding question content raises held-out accuracy by 1.50 points over VIDEOHARNESS-BRSI feedback and shrinks the gap by 1.28 points, consistent with Finding 2: the proposer can no longer tailor harnesses to the validation questions, so less of its validation progress is instance-specific fit. The remaining stages of VIDEOHARNESS-SRSI add 0.37 held-out points and shrink the gap by another 0.5 points; given the budget difference above and three seeds per setting, we read this second step as directional.

M HELD-OUT QUESTIONS ON UNSEEN VIDEOS

On MomentSeeker, the split assigns whole source videos to one side, so every held-out result on it concerns videos never seen during evolution. On Video-MME, LongVideoBench, and OVO-Bench, the split is over questions, so a held-out question may come from a video that also has validation questions: 702 of 2,250 held-out questions on Video-MME, 303 of 891 on LongVideoBench, and 622 of 979 on OVO-Bench. A harness could thus gain on such questions from properties of videos it was selected on. To check that the comparisons below do not rest on such gains, we re-score them on the *unseen-video* questions only, i.e., held-out questions whose video has no validation question (1,548 on Video-MME and 588 on LongVideoBench). Harnesses and predictions are those of the original evaluations; only the scored subset changes. Intervals are 95% bootstrap intervals over videos (2,000 resamples).

Information isolation. As shown in Table 20, restricting the proposer to content-free feedback raises accuracy on unseen-video questions by 1.64 points, with an interval that excludes zero. The effect of information isolation in Appendix L therefore holds on videos never seen during evolution.

Table 20: **Held-out accuracy (%) on all held-out questions and on unseen-video questions.** Δ: difference on unseen-video questions, with its 95% bootstrap interval over videos.

Comparison	Harness	All held-out	Unseen videos	Δ (unseen)
Information isolation (LongVideoBench, Table 19)	VIDEOHARNESS-BRSI feedback (Multimodal)	65.43	67.12	$+1.64$ $[+0.11, +3.28]$
	VIDEOHARNESS-BRSI + isolation (Blindfolded)	66.93	68.76	
Evolved vs. AKS (Video-MME, Figure 2)	Evolved `medium_augspread` (38 frames)	68.36	68.02	–
	AKS-BLIP (62 frames)	68.40	68.73	-0.71 $[-2.45, +1.16]$
	AKS-CLIP (62 frames)	68.09	68.22	-0.19 $[-2.20, +1.74]$

For AKS rows, Δ is the evolved harness minus that AKS variant.

Evolved versus hand-designed harnesses. On unseen-video questions, the evolved retrieval harness of Section 2.1 stays within one point of both AKS variants, and neither difference is significant, while it uses 38 rather than 62 frames per question. The conclusion that VIDEOHARNESS-BRSI evolves harnesses competitive with this hand-designed method therefore also holds on videos never seen during evolution.

N INFERENCE COST OF EVOLVED HARNESSES

This section details the budget analysis of Section 4.1. We measure the cost of a harness by three per-question counters recorded in its held-out traces: the frames passed to the executor, the executor (VLM) calls, and the tool calls. All values are means over the three seeds of Table 1.

Cost of the deployed harnesses. Table 21 compares the deployed harnesses of both frameworks. VIDEOHARNESS-SRSI spends more frames than VIDEOHARNESS-BRSI on Video-MME, OVO-Bench, and MomentSeeker, and uses more tool calls on Video-MME and LongVideoBench, where its harnesses retrieve and re-inspect evidence before answering. Part of its gain in Table 1 therefore coincides with a larger budget. The relation is not uniform, however: on LongVideoBench, VIDEOHARNESS-SRSI is more accurate with fewer frames, and on MomentSeeker with fewer tool calls.

Table 21: **Cost of the deployed harnesses.** Held-out score and per-question cost, averaged over three seeds.

Benchmark	Method	Test	Frames/q	VLM calls/q	Tool calls/q
Video-MME	VIDEOHARNESS-SRSI	70.1	106.1	2.44	19.3
	VIDEOHARNESS-BRSI	66.5	55.1	2.43	1.6
LVB	VIDEOHARNESS-SRSI	67.3	64.8	2.85	17.5
	VIDEOHARNESS-BRSI	66.6	69.2	2.77	5.0
OVO-Bench	VIDEOHARNESS-SRSI	72.5	47.8	4.91	2.6
	VIDEOHARNESS-BRSI	67.2	27.3	3.57	1.5
MomentSeeker	VIDEOHARNESS-SRSI	23.0	69.8	4.38	11.1
	VIDEOHARNESS-BRSI	19.0	60.3	2.52	14.7

Budget-matched selection. For each benchmark, the budget is the mean frames per question of the three deployed VIDEOHARNESS-BRSI harnesses. In each VIDEOHARNESS-SRSI run, we keep the candidates whose validation traces use at most this many frames per question and deploy the one with the highest validation accuracy, breaking ties by fewer frames. The selection uses no held-out data, so the result is a harness that VIDEOHARNESS-SRSI could have deployed under this budget. Table 22 reports its held-out score and cost next to the deployed VIDEOHARNESS-BRSI harness.

Results. At matched budget, VIDEOHARNESS-SRSI remains ahead of VIDEOHARNESS-BRSI on every benchmark, by $+2.43$, $+0.59$, $+1.22$, and $+2.41$ points, while using fewer frames and at most 0.09 more executor calls per question. Averaged over benchmarks, the matched gain

Table 22: **Budget-matched comparison.** VIDEOHARNESS-SRSI restricted to the frame budget of the deployed VIDEOHARNESS-BRSI harness and selected by validation accuracy, averaged over three seeds. Δ: held-out gain of VIDEOHARNESS-SRSI.

		VIDEOHARNESS-SRSI (budget-matched)			VIDEOHARNESS-BRSI (deployed)			
Benchmark	Budget	Test	Frames/q	Calls/q	Test	Frames/q	Calls/q	Δ
Video-MME	≤55.1	68.93	52.5	2.04	66.50	55.1	2.43	+2.43
LVB	≤69.2	67.19	57.8	2.86	66.60	69.2	2.77	+0.59
OVO-Bench	≤27.3	68.42	25.8	3.64	67.20	27.3	3.57	+1.22
MomentSeeker	≤60.3	21.41	56.0	2.48	19.00	60.3	2.52	+2.41

(1.66 points) is about half of the deployed one (3.40 points). Figure 14 places both frameworks and the fixed baselines on the cost–accuracy plane: on every benchmark, the budget-matched VIDEOHARNESS-SRSI harness lies above the deployed VIDEOHARNESS-BRSI harness at no more frames, and above every fixed baseline. Two conclusions follow. First, VIDEOHARNESS-SRSI converts part of its advantage into a larger budget when that budget pays off on validation, which the deployed harnesses reflect. Second, the remainder is not a budget effect: under the same frame budget, the evidence that VIDEOHARNESS-SRSI gathers still selects more accurate harnesses than VIDEOHARNESS-BRSI does.

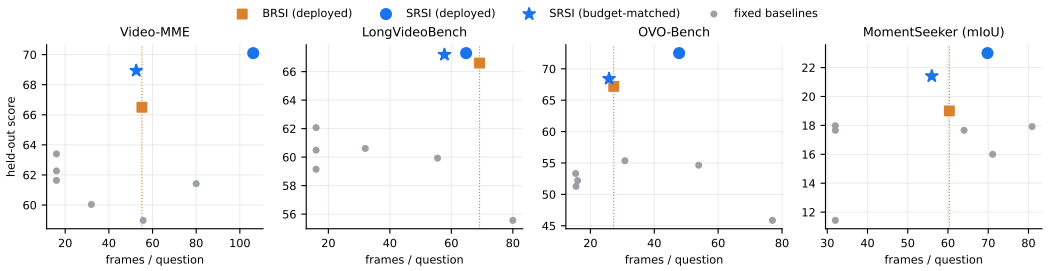


Figure 14: **Held-out score versus frames per question.** Deployed harnesses of VIDEOHARNESS-BRSI (squares) and VIDEOHARNESS-SRSI (circles), budget-matched VIDEOHARNESS-SRSI harnesses (stars), and fixed baselines (gray). The dotted line marks the budget, i.e., the frames per question of the deployed VIDEOHARNESS-BRSI harness.

O FAILURE-MODE DIAGNOSTICS OF VIDEOHARNESS-SRSI

Table 23: **Early-experience trap.** Runs whose champion never exceeds 1.2× the seeds' frame budget (16 frames; 32 on MS), three seeds each; VIDEOHARNESS-BRSI searches the full harness with Multimodal feedback. VM: Video-MME; OVO: OVO-Bench; MS: MomentSeeker.

	VM	LVB	OVO	MS	Total
VIDEOHARNESS-BRSI	1 / 3	0 / 3	1 / 3	2 / 3	4 / 12
VIDEOHARNESS-SRSI	**0 / 3**	0 / 3	**0 / 3**	**0 / 3**	**0 / 12**

Because the warm-up of VIDEOHARNESS-SRSI contrasts the frame budget in every run regardless of earlier verdicts, no VIDEOHARNESS-SRSI run stays at the seeds' frame budget on any benchmark (Table 23), whereas VIDEOHARNESS-BRSI runs do on Video-MME, OVO-Bench, and MomentSeeker.

P PROPOSER PROMPTS OF VIDEOHARNESS-SRSI

This section reproduces the most important proposer prompts of VIDEOHARNESS-SRSI, taken from a Video-MME run with the Qwen3-VL-8B executor and Claude Opus 5 as the proposer. Text

in braces, such as {BACKBONE}, is filled in by the orchestrator at run time with that run's measurements; run constants that the prompts leave as placeholders are filled in with the values of Table 17. We elide long passages with [. . .] and drop Markdown emphasis and non-ASCII symbols; everything else is verbatim. The prompts use internal names that map onto Section 3 as follows: a *family* is one of the seven mechanism families of Appendix J.2; the *backbone* is H_0; *WARM* is Stage I, whose Slots A/B/C are the free, removal, and intensity proposals; a *class-1 slot* is one of the six fixed axis templates and a *class-2* charter is the free-form axis α_{free}; *PROBE* and *RACE* are the $r_1 = 3$ and $r_2 = 15$ schedules; a branch's d_i is its effect $\bar{\delta}_\alpha$; a *cell* is a composition $H_{\mathcal{U}}$; and the *gate* is the admission test of Appendix J.3.

Iteration skill. Every evolution loop of VIDEOHARNESS-SRSI (WARM, PROBE, RACE, and MERGE) runs the loop of Appendix C.1, whose proposer session loads the skill below. In PROBE and RACE, the branch's charter skill replaces the opening sections, and the interface, constraints, and workflow are appended verbatim, identical on every branch. The proposer sees only content-free traces (Section 3.1); the leak guard at the end is enforced on every candidate before it is evaluated.

Prompt 1: Iteration skill shared by all evolution loops (excerpt)

```
# Meta-Harness: Full Agentic Framework

Run ONE iteration of evolution on Video-MME. Do all work in the main session - do
NOT delegate to subagents (constraints get lost, and you end up shipping parameter
tweaks with no prototyping).

You do NOT run benchmarks. You analyze results + traces, prototype, and implement
new frameworks. The outer loop benchmarks them on the resident Qwen pool and runs
the selection layer.

## What is being evolved

A complete training-free agentic framework that takes a video + a multiple-choice
question and returns the FINAL answer letter. Everything is on the table at once:

| mechanism                                   | primitive
          |
|---------------------------------------------|-------------------------------------------------|
| frame selection                             | uniform_indices, ctx.pool_indices(),
    ctx.read_frames |
| query-conditioned retrieval                 | ctx.retrieve(model, query, k)
          |
| perception tools                            | ctx.tools.detect / locate
          |
| cheap video signals                         | ctx.tools.shots / motion / similarity
          |
| pixel re-presentation                       | ctx.tools.zoom / crop / montage +
    ctx.look(...)  |
| self-reflection & multi-turn correction     | repeated ctx.vlm_answer conditioned on
    earlier output |
| debate / multi-agent collaboration          | ctx.agent(role, prompt, indices=...)
          |
| working-memory management                   | ctx.memory.note / digest / set_policy /
    forget   |

There is exactly ONE model doing inference: the frozen Qwen3-VL. "Multi-agent"
means several ROLES of that same model. You (the outer agent) write the framework
code; you never answer a question yourself, and neither does any other external
model. No finetuning, no weight updates, no new model downloads, no network.

Goal: maximize Video-MME accuracy. The cost axis is average frames summed over ALL
passes (tool calls and long generations cost too). A deeper or broader pipeline has
to EARN its cost: the frontier is accuracy vs frames, so machinery that buys nothing
is strictly worse than not having it.

## Interface - implement answer(ctx)
[... interface, primitives on ctx, and per-question budgets ...]

## CRITICAL CONSTRAINTS

- Tag each candidate's components with the mechanisms it uses. This is a ledger
  field for later analysis; nothing in the loop reads it back to you.
- You MUST implement 3 new frameworks every iteration, each with a base_strategy -
  the exact parent you built on, taken from evolution_summary.jsonl /
  frontier_val.json, or a baseline. Mix exploitation and exploration.
```

```
- Do NOT abort early, do NOT write "the frontier is optimal", do NOT skip
  prototyping.
- Anti-parameter-tuning: a new candidate must change a mechanism - what triggers an
  action, which evidence is gathered, how rounds/roles/memory/fusion work - not just
  num_frames, k, max_rounds or a threshold. If your candidate differs from its
  parent only by constants, REWRITE it.
- One frozen model only. All reasoning happens inside Qwen3-VL via ctx.vlm_answer /
  ctx.ask_vlm / ctx.agent / ctx.look. No external API, no second model, no
  answering by hand.
- Tool-cost discipline. [...]
- No dataset-specific hints. General patterns are fine ("keep frames where the
  queried object is detected"). Hardcoded knowledge about particular videos or
  questions is not.

### Leakage rules (enforced automatically - a violation REJECTS the candidate)

Every candidate is AST-scanned by leak_guard.py before it is benchmarked. Your
framework gets everything it needs through ctx, so:
- No filesystem, network or subprocess access - no open, pathlib, pandas, requests,
  subprocess, os.environ, no reading logs or the dataset.
- Imports must be on the allow-list [...]. No random (be deterministic).
- No dataset-identity handles: ctx.video_path, question_id, videoID, task_type,
  duration-label routing. (ctx.duration_sec - the number - is fine and useful.)
- No eval/exec/compile/__import__, no hashing question text, no large literal
  lookup tables.

Separately: you must never read the dataset parquet, the test split, other runs'
logs, or private .hstate/ state. The gold answer is not available to you or to
your framework and must not be sought.

## WORKFLOW

1. Analyze: read evolution_summary.jsonl, frontier_val.json, config.yaml, and recent
   val_log.jsonl traces. Look at where a framework flips right<->wrong, which
   families it actually used, and what it spent.
2. Prototype (MANDATORY): a throwaway script in .scratch/. With the stubbed VLM,
   verify the loop terminates, returns a valid letter, exercises the branch you
   intended, and survives the all-empty fallback path.
3. Implement: copy a baseline from agents/ to agents/<name>.py and build the new
   mechanism. Self-critique: is this a genuinely new mechanism, or a parameter
   variant? Then validate the import and run leak_guard.py on it.
4. Write pending_eval.json to the path in the task prompt.
```

Stage I: warm-up slots. In the WARM workspace, the orchestrator appends the addendum below to the iteration skill, which turns each round into the three free, one removal, and one intensity candidates of Section 3.2.1. The prompt only requests the slots; as stated in its last rule and in Appendix J.2, each candidate's slot is audited from its trace. The free-only control of Section 4.2 receives an addendum that assigns all five candidates to Slot A.

Prompt 2: Stage I warm-up slot addendum

```
# WARM PHASE ADDENDUM - five candidates per iteration, in three slots

This warm phase is not only searching for a good framework. It is also
manufacturing the evidence that the next phase needs in order to charter branches.
Those are different jobs and they need different candidates.

| slot                            | count | job                           |
|---------------------------------|-------|-------------------------------|
| A free exploration              | 3     | find a good backbone          |
| B one-family ablation           | 1     | manufacture a control group   |
| C one-family intensity sweep    | 1     | manufacture an intensity range |

Slot A keeps all three of the candidates a normal round would have had. B and C
are added on top, not carved out of A, so nothing about your search for a good
framework is reduced.

## slot A - free exploration (3 per iteration)

Whatever you think is most promising. No constraint beyond the SKILL above.

## slot B - one-family ablation of the current champion (1 per iteration)

Take the current best framework and produce a variant that removes exactly one
```

```
primitive family it uses, keeping everything else byte-for-byte. Pick the family
with the highest usage share that you have not already ablated.
Name it slotB_<family>_off (e.g. slotB_retrieve_off).

Why this slot exists: a family that every candidate uses has no control group, so
any with/without estimator on it returns "too few candidates to test" - and the
next phase is then structurally unable to say anything about it. That is a
property of the estimator, not of the family, and this slot is what fixes it.

You are not trying to make this candidate win. A large drop is the useful outcome:
it is the measurement.

## slot C - one-family intensity sweep (1 per iteration)

Take any framework in the archive and produce a variant that changes how hard one
family is worked - and nothing else. Halve or double the retrieval budget, run the
detector on 3 frames instead of 12, take 2 reflection passes instead of 1. Same
family set, same structure, one dial moved by at least ~2x.
Name it slotC_<family>_<up|down> (e.g. slotC_retrieve_up).

Why: a family that every candidate already uses cannot be measured by
presence/absence, but it can still be measured by dose. This slot is what creates
the intensity range that measurement needs.

## Rules for all five candidates

- All five are implemented and evaluated normally. B and C compete for the
  champion like anything else - they are simply unlikely to win, and that is fine.
- Do not spend the frame budget to make B or C look good. Frame count is a strong
  confound on this task: a B or C candidate that also moves the frame budget
  measures the budget, not the family.
- If the champion has only one family left to remove and you have already ablated
  it, use slot B for a second intensity sweep on a different family and say so in
  the hypothesis field.
- Put the slot letter in the strategy name. The orchestrator audits the slots
  afterwards from behaviour, not from the name (a candidate counts as an ablation
  if its family set is a strict subset of its parent's), so a mislabelled name is
  harmless - but a correct name makes the log readable.
```

Stage II: nomination. NOMINATE is a single proposer call that receives the warm-up report (Appendix J.2) in place of the braced tables and writes one charter per axis: its mechanism, rationale, first edit, falsification criterion, and predicted effect (Section 3.2.2). The axis templates it must fill are given in Prompt 4.

Prompt 3: Stage II NOMINATE (excerpt)

```
# NOMINATE - fill the fixed axis slots, then add your own

The warm phase is over. Its job was not to find the best framework - it was to
produce a shared backbone and enough measured history to charter the branches that
come next.

Your task is not to write a framework, and it is not to decide which axes are
worth exploring. The axis set is largely decided for you. Your task is:
1. for every class-1 slot below, write the mechanism that slot will try on this
   run, and write that branch its own harness (SKILL.md);
2. add up to 1 class-2 charter of your own - a composite or mechanism this
   vocabulary has no slot for;
3. commit to a numeric prediction for each.

After you finish, every charter gets a short blind branch of 3 candidates. The
ones that survive elimination continue to 15, and only then is an effect size
with a standard error computed and gated.

## Why the slots are fixed, and why you cannot skip one

An earlier version of this phase let the proposer choose the axes freely. Its
recorded failure mode was reading a null union-ceiling increment as evidence of no
headroom and excluding that family - when a null there means only that every warm
candidate used the family, so there was no control group to measure it against.

increment: null means UNTESTABLE, never REFUTED. [...] a slot cannot be skipped,
so a family cannot be dropped for being unmeasurable.

You are not being told which axis is good. You are being told that the question of
```

```
which axis is good is settled by the PROBE and RACE phases that follow, on this
run's own measurements - not by an argument made here.

## What the warm run measured

Backbone: {BACKBONE} at {BACKBONE_VAL}% on {N_VAL} val questions.
Families it already uses: {BACKBONE_FAMILIES}.
Paired noise floor for a single candidate on this run: {PAIRED_SE}.

### Family share, and the union-ceiling increment
{FAMILY_TABLE}
### IMPORTANCE - what the ablation slots measured (removing the family from the champion)
{IMPORTANCE_TABLE}
### HEADROOM - dose-response among candidates that use the family
{HEADROOM_TABLE}
> It is observational - it is read off candidates that were free to change several
> families at once, so it is confounded by construction [...]. Do not rank axes on it.
### The free baseline ladder
{LADDER_TABLE}
> They measure importance from a zero floor, not headroom on top of your backbone.
> [...] treat the ladder rows for those families as uninformative.
### Frames
{FRAMES_NOTE}
> Frame budget is the strongest single confound on this task: an axis that raises
> accuracy by also raising frames has measured the budget, not the mechanism.
> Every class-1 template except frame_budget forbids moving it, and frame_budget
> exists so that the compute axis is priced openly instead of being smuggled into
> another one. Honour that in the mechanism you write.

## The class-1 slots you must fill

For each slot, families_allowed, candidate_test and forbidden are fixed and printed
in full below - copy them into your entry. What you actually write is the
mechanism: the specific thing this branch should try on this backbone, this run.
Read candidate_test before you write the mechanism. It is the definition of "on
axis" that every candidate on that branch is measured against, so a mechanism that
cannot satisfy it is a branch that measures nothing.

{VOCAB_TABLE}

## Output contract

Write logs/warm/nomination.json:
{"axes": [{
  "name": "retrieval_span_walk", "axis_class": 1, "vocabulary_id": "retrieval",
  "families_allowed": ["retrieve"],
  "candidate_test": "<copied VERBATIM from the vocabulary template>",
  "charter": "one paragraph: what this branch may change and what it may not",
  "mechanism": "the specific thing to try on THIS backbone, and why this one",
  "why_this_axis": "cite a number from the warm run above",
  "what_would_falsify_it": "what pattern over ~15 candidates would kill it",
  "first_move": "the exact first diff, small enough to implement in one iteration",
  "expected_effect_pp": 0.8,
  "frame_neutral": true }]}

Rules the contract check enforces - a violation costs you a retry:
- every class-1 slot listed above appears exactly once. You may not skip one.
- at most 1 entry with "axis_class": 2. [...] you write all fields - including your
  own candidate_test, since there is no template to take one from. Write it the way
  the class-1 tests above are written: what a diff must provably do for the branch
  to count it as on-axis.
- expected_effect_pp is a real prediction in [-5, +5], and they must not all be the
  same number. It is scored against the measured d_i afterwards.
- frame_neutral states whether the mechanism can raise accuracy without raising the
  frame budget.

And write one harness per axis at logs/warm/axis_skills/<axis_name>/SKILL.md [...].
Every SKILL is written by you in one session, but the branches are blind to each
other. A SKILL that names another branch voids that blindness and is rejected by
the contract check. Write each one as if it were the only branch.
```

Stage II: axis templates. The six class-1 templates of Section 3.2.2 fill {VOCAB_TABLE} in Prompt 3. Each fixes the families a branch may touch, the on-axis test its every edit must pass, and what it must not change; only the mechanism is left to the proposer. Only frame_budget may change the number of frames, so that the frame budget, the strongest confound on these benchmarks, is measured on its own axis.

Prompt 4: The six fixed axis templates (class-1 slots)

```
retrieval            families: retrieve
  WHICH frames the answering pass is shown.
  candidate_test: The diff MUST change how the indices handed to ctx.read_frames /
    ctx.vlm_answer(indices=...) are produced, on the path to the returned letter, so
    that reverting the diff provably changes which frames the model sees.
  forbidden: changing the total frame budget (num_frames) relative to the backbone.

frame_budget         families: retrieve
  HOW MANY frames, and how they are spread across a question.
  candidate_test: The diff MUST change the number of frames read per question (mean
    over val) by at least 20% relative to the parent, either globally or by a
    runtime rule, while keeping the parent's selection rule byte-for-byte.
  forbidden: changing which retriever, ranking or selection rule is used.

read_schedule        families: multiturn
  The pass graph AND the rule that turns pass outputs into a letter.
  candidate_test: The diff MUST change at least one of: (a) which ctx.vlm_answer /
    ctx.ask_vlm passes run or what they are conditioned on at runtime, or (b) the
    function from collected pass outputs to the returned letter (abstention,
    tie-breaks, thresholds, option elimination).
  forbidden: changing the total frame budget; changing which frames are selected.

multi_role           families: roles
  More than one answering persona, and how they are adjudicated.
  candidate_test: The diff MUST issue at least two passes with materially different
    role prompts and combine them by a rule that depends on their disagreement.
  forbidden: changing the total frame budget.

pixels_and_detectors families: perceive, render
  What image is built and shown, and the object geometry that decides it.
  candidate_test: The diff MUST either (a) hand ctx.look a pixel array built by a
    composition rule the parent does not have (different tiling/geometry, a crop or
    rescale, several moments joined, an extra plane), or (b) call ctx.tools.detect /
    locate and route a value from that call into the composition or the prompt. In
    both cases the frame list feeding it MUST be copied from the parent.
  forbidden: changing which frames are selected; changing the total frame budget.

derived_state        families: analyze, memory
  Quantities the framework computes about the video or about its own passes, and
  carries forward.
  candidate_test: The diff MUST either (a) compute a quantity from the VALUES
    returned by ctx.tools.shots / motion / similarity that the parent does not
    compute, or (b) contain a ctx.memory.note(...) whose text derives from an earlier
    pass's raw output together with a later prompt that embeds ctx.memory.digest(...)
    or ctx.memory.items(...). In both cases there MUST be a visible data path from
    that quantity into the frame indices or a prompt on the way to the returned
    letter.
  forbidden: changing the total frame budget.
```

Stage III: composition. COMPOSE is a single proposer call after the race. It receives each raced axis's effect $\bar{\delta}_\alpha$ with its lineage-clustered standard error (Eq. (17)) and the gate's verdict, and it builds every cell of Eq. (3) from one fixed modification d_α per axis. The cells are then scored and seed the MERGE ledger (Appendix J.4). The prompt's "15 rounds" refers to the $r_2 = 15$ candidates of each raced branch.

Prompt 5: Stage III COMPOSE (excerpt)

```
# COMPOSE - build the whole lattice, ship only what the gate pre-registered

The {K} branches are finished. Each spent 15 rounds moving exactly one axis of the
same frozen backbone, blind to the others. This step turns them into one framework.

## What the measurement says

Backbone: {BACKBONE} (val {BACKBONE_VAL}), families {BACKBONE_FAMILIES}.
Per-axis mechanism estimate - the mean paired val delta of that branch's candidates
against the backbone, with a lineage-clustered bootstrap SE (same-parent
descendants count as one draw, because they are edits of each other and not
independent evidence):
{AXIS_TABLE}
Admission gate (pre-registered):
{GATE}
```

```
## The rule you are implementing

The gate does not decide what you BUILD. You build the whole lattice. The gate
decides what may later be SHIPPED, which is a separate step you are not performing.
Your job is to make every combination measurable, including the ones the gate
rejected - a rejected axis that turns out to combine well is exactly the thing this
design exists to catch, and a filter can never show it.

### Step 1 - fix one diff per axis, before you write anything

For each of the {K} axes, extract one diff: the smallest edit to the backbone that
carries that axis's mechanism, generalised across that branch's candidates if you
like, but attributable to that axis alone. Write these down first.

### Step 2 - build all {N_CELLS} cells
{LATTICE}
Every cell is backbone + the diffs of its axes, and the diff for an axis must be
byte-identical in every cell that contains it. This is not a style rule. The
lattice is read as a factorial design: I_ab = d_ab - d_a - d_b is only an
interaction if the a in cell_ab is the same a as in cell_a. If a diff has to change
to fit alongside another, that is not a merge - it is a rewrite, and the right move
is to shrink both diffs until they compose without editing each other, then say so.

Singleton cells are not optional. They are the main-effect cells and no interaction
is computable without them. Note that the effect of cell_a is not the same quantity
as axis a's d_i in the table above: d_i is the mean over ~15 branch candidates,
cell_a is the effect of the one diff you chose. The gap between them is real
information - if your chosen diff underperforms the branch mean, you picked the
wrong diff.

### Step 3 - if a cell cannot be built, say which and why

Two diffs may be genuinely incompatible (both replace the same seam). Do not paper
over it: write the cells you can, and in the hypothesis of the smallest affected
cell name the conflict precisely. [...]

## Constraints that are checked

- Complexity budget. Count if/elif branches and named numeric constants in the
  backbone and in each diff you graft. The merged candidate may not exceed
  backbone + sum(diffs). [...] declare the counts in your hypothesis so they are in
  the ledger.
- No rewriting the backbone. If a graft seems to require restructuring the
  backbone, that is evidence the two mechanisms are not composable at this seam:
  say so in the hypothesis, and graft the smaller version instead.
- Every candidate declares base_strategy = the backbone (for the first prefix) or
  the previous prefix.
- Read the on-axis / bespoke on-off axis / fired columns before the d column. [...]
  A branch with more bespoke code off its axis than on it spent its rounds
  somewhere other than its charter, so its d_i is not an estimate of the mechanism
  its name claims - treat that branch's diffs as unlabelled, and say so.

## What to write
[... manifest path and format; every cell lists the axes it grafts ...]
In each hypothesis, state: which axes this cell grafts, which branch candidate each
diff came from, the complexity counts, and what you expect the effect to be - so
that a smaller-than-predicted effect is visible as a failed prediction rather than
absorbed silently. For the multi-axis cells, also predict the sign of the
interaction: do you expect these mechanisms to add, to overlap (sub-additive, they
fix the same questions), or to fight (super-additive against you)?
```