SELFLINEAGE: LEARNING THE LONG-TERM CONSEQUENCES OF SELF-MODIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A persistent self-modification can have consequences that appear only when later tasks or edits reuse it. We introduce SELFLINEAGE, a minimal descendant-aware history for harness-level recursive self-improvement with a frozen foundation model. It records actual edits and outcomes, then adds provenance-backed relations when later consequences become observable. At a fixed decision state, Raw-History and SELFLINEAGE receive the same factual nodes; the intervention changes their explicit lineage structure, without adding search or a learned value model. We specify experiments on EvoHarnessBench: EOG Tools provides the primary controlled setting, Skills tests reusable libraries, and Agents stresses orchestration across expanding specialist pools. Private matched continuations measure the value of selected edits. Factual-access, shuffled-link, and truthful temporal-link controls distinguish later evidence from functional relations. Fixed versus adaptive continuations and live versus frozen link updates test whether the benefit extends to further self-improvement. End-to-end capability, retention, and complete cost determine whether these mechanisms yield useful adaptation. The central question is whether correct descendant relations make subsequent self-modification decisions better.

1 INTRODUCTION

An agent can change the router, verifier, recovery procedure, or context manager that its future versions use. A new verifier may initially cost time yet prevent failures across later tasks; its diagnostic may also enable a subsequent planner revision. The same investment can be wasteful when little compatible work remains. Immediate feedback therefore need not capture the value of a persistent self-modification.

This raises a representation question: given the same observed edits and outcomes, can explicitly connecting later consequences to the edits they reuse or depend on improve the next self-edit? We hold the foundation model, self-edit instruction, retriever, validation gate, and online resource limits fixed. Our primary comparison changes the relational structure presented to the agent, with identical historical nodes admitted at a common decision checkpoint.

SELFLINEAGE records immutable persistent edits, links later outcomes through typed, provenance-backed relations, and retrieves this history for the next self-edit. The frozen agent chooses one change or NO-EDIT. Each outcome payload appears once, even when several edits reference it. Temporal succession remains context; observational links do not assign causal credit.

Our core evaluation uses EvoHarnessBench's public EOG Tools streams (Ke et al., 2026). External capabilities change by stage while the agent modifies its internal harness using adaptation experience. Stage tests remain measurement only. We assess capability, retention, cost, and selected edit value in same-state interventions. EOG Skills tests reusable libraries; EOG Agents tests orchestration as specialist pools expand, including a qualified 15-stage stress stream. Factual-access, temporal-link, and fixed-versus-adaptive continuation tests examine each step from historical evidence to further self-improvement.

The contribution combines persistent self-edit records, verified functional use, time-valid retroactive evidence, and matched-fact decision tests. Broad memory and harness baselines assess practical

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

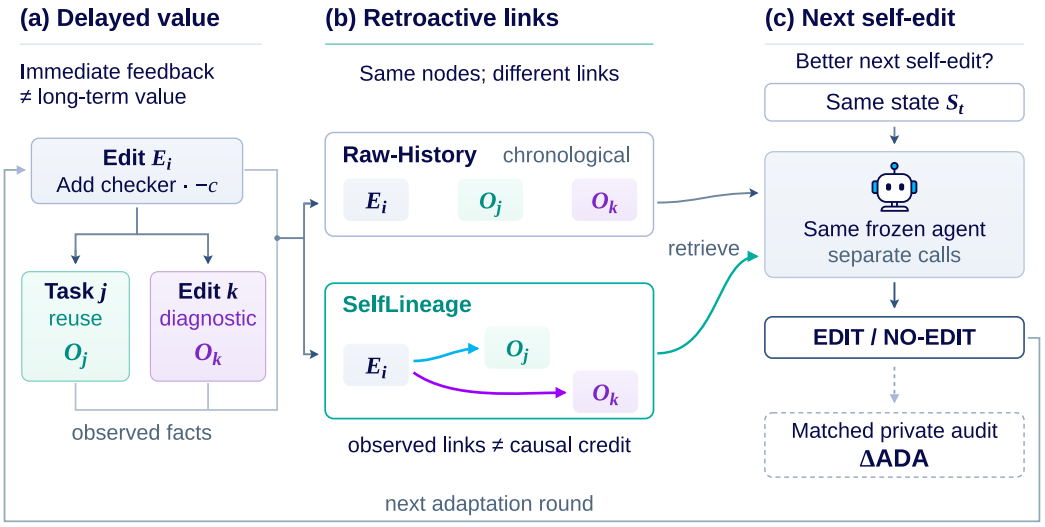

Figure 1: **SELFLINEAGE connects delayed consequences to the next self-edit through a shared history.** (a) Later task reuse (O_j) or a subsequent edit's use of a diagnostic (O_k) reveals consequences of edit E_i. (b) Raw-History and SELFLINEAGE share factual nodes; SL adds verified edges, including adverse evidence. (c) Separate calls to the same frozen agent select the next edit. Solid arrows close the online loop; the dashed private audit measures paired selected ADA (Equation 9) without feedback to that loop. Shuffling changes assignments over the same nodes; admission, instruction, and gate remain fixed.

competitiveness; Raw, Temporal, and Shuffled isolate the relational hypothesis. Neither feedback memory nor dynamic linking alone is claimed as new.

2 RELATED WORK

Self-modifying agents. SICA edits itself (Robeyns et al., 2025); DGM archives self-modifications (Zhang et al., 2026a), and HGM uses descendant performance to guide expansion (Wang et al., 2026a). Harness evolution also uses observability, hierarchy, and code search (Lin et al., 2026; Zhou, 2026; Lee et al., 2026). Dream-RSI replays discovery history to refine exploration (Zheng et al., 2026); EvoMem retrieves mutation advice across program-search runs (Volkov et al., 2026). Our mechanism study fixes the edit policy and intervenes on the representation of prior consequences.

Feedback and context optimization. Self-Refine revises outputs (Madaan et al., 2023); Reflexion stores feedback reflections in episodic memory (Shinn et al., 2023); ExpeL and ReasoningBank distill reusable experience (Zhao et al., 2024; Ouyang et al., 2026). TextGrad propagates textual feedback (Yuksekgonul et al., 2024), GEPA evolves prompts (Agrawal et al., 2026), and ACE maintains contextual playbooks (Zhang et al., 2026b). Feedback memory without weight updates is established prior work.

Structured memory and reusable skills. Generative Agents retrieves and reflects on experience (Park et al., 2023); MemGPT manages memory tiers (Packer et al., 2023); A-MEM dynamically links memories by semantic association and updates their attributes (Xu et al., 2026b). MemToolAgent extracts feedback-based tool memories (Er et al., 2026); Voyager maintains executable skills (Wang et al., 2023). Dynamic linking alone is therefore not our novelty. Our narrower contribution combines persistent self-edit records, actual-use functional relations, cutoff-valid retroactive evidence, and matched-fact tests of subsequent edit choice.

Evaluation. AgentBench evaluates interactive decisions (Liu et al., 2024); AgentBoard analyzes multi-turn progress (Ma et al., 2024). EvoHarnessBench supplies evolving external tools, skills,

or specialists (Ke et al., 2026). Keeping these resources outside the writable harness separates persistent internal improvement from task-specific outputs, a central evaluation concern (Wang et al., 2026b).

3 Problem Setup

For the shared self-edit protocol, $A_t = (\theta, X_t, L_{\leq \tau_t})$ at opportunity t: frozen parameters θ, reusable internal harness X_t, and history available by cutoff τ_t. The benchmark supplies external harness $\mathcal{H}_{s(t)}$ for stage $s(t)$. Self-state S_t describes observed failures, component versions, available tool families, and remaining resources. The action is $\mathrm{EDIT}(\Delta_t)$, at most one implemented persistent change, or NO-EDIT.

The writable surface includes internal routing, context, verification, recovery, workflows, reusable skill libraries, and persistent lead-orchestrator policies. Benchmark tools and specialist specifications are external resources, not self-edits. Current-task patches, temporary delegation choices, benchmark manifests, and authoritative graders are outside the self-edit lineage.

A trial produces $X_t^{\Delta} = \mathrm{Apply}(X_t, \Delta_t)$. After static/interface checks, the shared gate evaluates a small paired adaptation probe P_t:

$$\widehat{\delta}_t = \frac{1}{|P_t|} \sum_{x \in P_t} \left[R(X_t^{\Delta}, \mathcal{H}_{s(t)}, x) - R(X_t, \mathcal{H}_{s(t)}, x) \right]. \tag{1}$$

Acceptance deploys the trial; rejection restores the parent. Actual exposure, failed trials, and their costs remain recorded. Missing probe results are unknown, and unapplied proposals have no invented execution outcomes.

For a common parent, comparator c, matched future adaptation/harness stream ω, fixed neutral improver, and independent private endpoint evaluator U, define

$$\mathrm{ADA}^{(h)}_{\Delta,c} = \mathbb{E}_{\omega} \left[U(X^{\omega}_{\Delta,h}) - U(X^{\omega}_{c,h}) \right]. \tag{2}$$

This *Audited Descendant Advantage* measures the value of a selected action after h further opportunities under a specified continuation policy and budget. It is a research estimand, computed in isolated branches. Neither ADA nor stage-test or external-evaluation outcomes can enter online history. The primary question is whether the same frozen agent selects higher-value actions when shared historical facts are supplied with correct lineage relations.

4 Method: SelfLineage

SelfLineage changes the history read by the self-improver. Its operations are factual recording, retroactive linking, and contextual retrieval. The schedule, edit instruction, action space, and gate are common across mechanism conditions.

4.1 Delayed Consequences and the Role of Representation

An infrastructure edit with setup cost $c > 0$ and benefit $b > 0$ on each of n compatible future uses has illustrative net value $nb - c$. Its immediate probe can be identical across situations with opposite long-term values. Reuse can also enable another edit, or introduce maintenance costs and regressions. Remembering only the immediate result discards these later observations.

Our primary baseline already sees those observations. Let V_t denote the admitted factual edit and outcome nodes, E_t^{time} their chronological organization, and E_t^{lin} the verified lineage relations:

$$G_t^{\mathrm{Raw}} = (V_t, E_t^{\mathrm{time}}), \qquad G_t^{\mathrm{SL}} = (V_t, E_t^{\mathrm{time}} \cup E_t^{\mathrm{lin}}). \tag{3}$$

If edges are reconstructible from shared provenance, their serialization adds no information for an unrestricted ideal reader. The hypothesis concerns a bounded model: explicit relations may make distant dependencies easier to use, while incorrect links, salience, and context cost may hurt. Benefit therefore requires measurement, not an information-gain theorem.

4.2 IMMUTABLE EDIT AND OUTCOME NODES

An actual persistent edit receives a base record

$$E_i = (S_{t_i}, \Delta_i, r_i, C_i, \nu_i), \qquad O_j = (\nu_j, \tau_j^{\text{obs}}, y_j, p_j), \tag{4}$$

where r_i contains immediate validation observations, C_i the paid costs, and ν_i the parent/trial version and changed-artifact identifiers. An outcome node O_j identifies its execution version ν_j, observation time, measured payload y_j, and provenance p_j. State, edit, contemporaneous rationale, outcome, and version facts are sealed. Later lifecycle events and edges append without rewriting them.

Lifecycle events distinguish static rejection without exposure, isolated testing, deployment, rollback, and supersession. Rejected static proposals have no execution descendants; reverted trials retain their actual negative observations and earlier uses. No-edit and unapplied proposals remain in the decision/cost ledger.

An outcome payload is stored and rendered once, even when multiple edits reference it. This preserves shared ancestry without repeating evidence or multiplying reward. Current-task solutions and benchmark-provided additions never become internal self-edit nodes.

4.3 RETROACTIVE CONSEQUENCE LINKING

A *runtime-reuse* relation requires a trace showing that downstream computation actually uses the edited component's content, interface, or surviving artifact. Loading or reading an artifact without using its content is insufficient. An *evolution-dependency* relation requires traceable use of an earlier edit's diagnostic, interface, tool, or representation in constructing a later self-edit. Temporal succession, component presence, and unsupported actor explanations establish neither relation.

Let t_i be the opportunity of edit i and $t(o)$ the opportunity index of a later outcome's execution. The observation window spans the next H opportunities, including no-edit opportunities:

$$\mathcal{A}_H(o) = \{i : 1 \le t(o) - t_i \le H\}. \tag{5}$$

Unlike an executed-edit count, this clock does not extend the window when the agent abstains. The starting setting is $H = 3$, with $H = 5$ as a sensitivity analysis; the private continuation horizon h is registered separately.

For a proposed typed link $\ell = (i, j, \rho, p_{ij})$, let a_ℓ be the latest availability time of its outcome, supporting provenance, and link creation. It is visible at cutoff τ only if

$$i \in \mathcal{A}_H(O_j), \quad a_\ell \le \tau, \quad \text{Contains}(\nu_j, \nu_i, p_{ij}), \quad \text{Verified}_\rho(p_{ij}). \tag{6}$$

Containment requires actual exposure to the relevant edit/artifact version. For evolution dependency, it may be a traced surviving diagnostic path rather than continued installation of the original code. An edit's own immediate probe remains a base observation. Discovering a relation later cannot make it visible to an earlier decision.

A checker used by a later task supplies runtime evidence; a planner revision consuming its diagnostic supplies evolution evidence. Failures follow the same rule as successes. Each observational link has `causal_claim=false`: coexisting edits and external changes preclude assigning an outcome's full value to one ancestor. Links supply no learned reward. Open windows remain maturing, and incomplete run-end windows censored; neither is filled with zero.

4.4 MATCHED RETRIEVAL AND RELATIONAL CONTROLS

The retriever uses observable failure patterns, internal component/interface compatibility, external harness stage, and remaining budget/horizon. A deterministic base-field score suffices:

$$s(S_t, E_i) = \sum_{f \in \mathcal{F}} w_f \, \text{sim}_f(S_t^{(f)}, E_i^{(f)}), \qquad I_t = \text{TopK}_i \, s(S_t, E_i). \tag{7}$$

Weights, tie-breaking, k, and field truncation freeze before confirmation. Neither ADA nor a descendant reward enters ranking. The initial k range is three to five.

At a common checkpoint, all three relational conditions admit identical edit/outcome nodes through a deterministic base-field rule, independent of edge correctness and symmetric for adverse evidence. Admission uses the locked tokenizer and a common relation allowance. Any overflow removes the same lowest-priority node from every rendering; representation-specific truncation is prohibited.

Raw-History serializes the admitted edits, outcomes, and provenance chronologically without an explicit edge list. SELFLINEAGE adds verified edges between those same IDs. Shuffled-Link permutes endpoint assignments with a registered seed, preserving the nodes, relation-type counts and format while disrupting correctness. Its edges are experimental placebo annotations, never verified online evidence. All three have the same cutoff and token ceiling; payloads occur once per packet. Exact counts and unavoidable token-length differences are reported.

Direct uses current state without past memory. Immediate retrieves past edits through their immediate observations. These are auxiliary comparisons; neither substitutes for the same-facts Raw-History baseline. Histories can diverge after different online actions, so node identity is asserted within each matched checkpoint intervention, not across independently evolved trajectories.

4.5 THE SHARED EDIT LOOP AND ITS COST

The frozen model receives the common instruction I_{edit} and representation-specific packet $\mathcal{P}_t^{(v)}$:

$$a_t \sim \pi_\theta(\cdot \mid S_t, \mathcal{P}_t^{(v)}, I_{\text{edit}}). \tag{8}$$

It chooses no-edit or implements one change. The fixed gate requires static/interface validity and $\widehat{\delta}_t \geq -\epsilon$, with the probe and tolerance $\epsilon \geq 0$ locked before confirmation. A small tolerance permits initially neutral or costly infrastructure without promising non-regression. Rejection restores the parent and records paid work; there is no rescue search.

Algorithm 1 SelfLineage under controlled external harness evolution

Require: Frozen model θ, internal harness X_0, external stage manifests, gate, H, retriever, budgets

```
Initialize immutable node store, append-only edge/lifecycle store, and cost ledger
for each external harness stage do
    Mount the official external capability set; keep it outside the writable genome
    for each scheduled adaptation batch whose resource block fits do
        Run tasks; store actual outcome nodes and current self-state
        Append admissible relations using Equation 6; fix decision cutoff
        Retrieve candidate nodes, admit the common factual block, and render its representation
        The same agent chooses NO-EDIT or implements one persistent edit
        Apply the common gate; retain all real edit, exposure, status, and cost facts
    end for
    Freeze within stage; evaluate held-out tests without history or retrieval updates
end for
Seal final systems and cost ledgers; keep private mechanism audits isolated
```

Recursion proceeds through adaptation information: a self-edit changes later execution, realized outcomes extend its lineage, and that history informs another self-edit. Only the self-improvement session reads the history packet; task execution uses the committed internal harness. Held-out stage tests, oracle metadata, and private continuations never feed this loop.

With at most one edit per opportunity, M outcomes require at most $O(MH)$ relation checks, plus trace processing, retrieval, and serialization. Deduplication does not bound history growth. Online cost includes every task, edit/no-edit, gate, failure, rollback, and memory operation, including verification with no qualifying edge. Private ADA and final tests have separate ledgers. Unsupported relations leave a chronological packet; receipts preserve candidate relations, exclusions, and rendering costs.

5 EXPERIMENTS

Tables 1–4 test system performance, edit choice, components, and further adaptation; Figure 2 examines reuse conditions.

5.1 EVALUATION PROTOCOL

Environments. EvoHarnessBench separates cumulative external capabilities from the internal harness that the agent edits (Ke et al., 2026). EOG Tools is primary, targeting all nine streams after ITSM/CSM/HR qualification; EOG Skills tests reusable libraries across four streams. EOG Agents qualifies on ITSM/HR before a resource-qualified 15-stage enterprise stream, growing from 16 to 33 specialists, or a shorter subset locked before results. Supplied tools, specialists, graders, and splits are immutable. Persistent routing, verification, recovery, context, and authored skills may evolve; current-task outputs do not count as self-edits.

Conditions and budgets. Frozen/no RSI disables persistent updates; Direct edits without history. Immediate, Raw, and SL progressively expose immediate facts, later facts, and verified relations. ReasoningBank-style and A-MEM-style adapt memory to the common editor; MemToolAgent† supplies task memory, and Meta-Harness‡ searches harness code. All share frozen DeepSeek-V4.1-Flash and within-axis ceilings (Xu et al., 2026a); only mechanism arms share the retriever and edit policy. Pilots lock $H = 3$, $k \in \{3, 4, 5\}$, paired run targets, budgets, and thresholds before confirmation.

Sealed measurements. Only official train/adapt experience can update the harness or history. Stage tests freeze a copy; adaptation resumes the pre-test state without scores, transcripts, or failure summaries. Task-level oracle assignments and private continuation outcomes remain evaluator-only. Benchmark-supplied specialist instructions remain visible for execution. Source revisions, eligible streams, and integrity rules are fixed in advance.

5.2 OVERALL ADAPTATION, RETENTION, AND COST

Table 1 gives SL–Raw gains of $8/6/9$ p.p. in Q, $7/4/8$ p.p. in R, and $0.07/0.05/0.08$ AUC on Tools/Skills/Agents. SL exceeds ReasoningBank-style by $6/5/7$ p.p. in Q and A-MEM-style by $7/4/6$ p.p. These comparisons test competitiveness beyond raw recall and generic linking; they do not isolate lineage correctness. Direct–Frozen separates the common self-edit loop from no RSI.

MemToolAgent and Meta-Harness are strongest in different axes; no single comparator defines the full gap. Each named comparison uses matched runs and fixed domain weights. Costs remain separate: new-arm costs are unprovided, so AUC and rankings cannot establish lower expenditure. Lead, specialist, memory, and search calls all count. Unreached Q^* targets are censored; actual cost curves and paired intervals must accompany any efficiency claim.

Table 1: **Overall adaptation** Q/R are success/retention (%); AUC is on $[0, 1]$. †: task memory; ‡: harness search.

	Tools			Skills			Agents		
Method	$Q\uparrow$	$R\uparrow$	AUC↑	$Q\uparrow$	$R\uparrow$	AUC↑	$Q\uparrow$	$R\uparrow$	AUC↑
Frozen / no RSI	28	26	0.28	16	15	0.16	8	7	0.08
Direct	30	28	0.28	18	17	0.16	9	8	0.08
Immediate	32	30	0.30	20	19	0.18	12	11	0.10
Raw-History	34	32	0.30	22	21	0.19	15	13	0.12
ReasoningBank-style	36	34	0.32	23	22	0.20	17	15	0.14
A-MEM-style	35	33	0.31	24	22	0.21	18	16	0.15
MemToolAgent†	39	35	0.34	22	20	0.19	17	14	0.14
Meta-Harness‡	38	34	0.32	24	22	0.21	20	17	0.16
SELFLINEAGE	**42**	**39**	**0.37**	**28**	**25**	**0.24**	**24**	**21**	**0.20**

Tools supplies the primary estimate; Skills tests library invocation and revision. Agents tests persistent routing and handoff under pool expansion. Pool size covaries with stage and difficulty, precluding causal attribution. Fifteen stages do not extend the $H = 3$ opportunity window. Axis-specific effects remain separate.

Matched subsets align Shuffled comparisons. Tools/Skills subsets differ from the main aggregate; Agents shares its enterprise sample and updated centers. Overlap provides no independent confirmation. Routing, handoff, and gold-assignment diagnostics remain separate; effective specialists need not match the oracle. New-cohort success and forgetting retain distinct denominators.

5.3 Do the Same Facts Lead to Better Edit Choices?

Online histories diverge, so Table 2 uses common EOG Tools checkpoints sampled before audit outcomes. Fix parent version, external stage, failures, cutoff, nodes, model, and gate. Implement each selected EDIT/NO-EDIT in isolation; discard treatment packets and continue matched tasks, stages, and resources under a common neutral improver.

Using the no-edit parent comparator in Equation 2, the primary contrast is

$$\Delta V_{\mathrm{SL-Raw}} = \mathrm{ADA}(\Delta_{\mathrm{SL}}) - \mathrm{ADA}(\Delta_{\mathrm{Raw}}). \tag{9}$$

Raw–Immediate tests access to later facts; SL–Raw tests explicit relations over those facts. SL–Temporal tests specificity beyond chronology; SL–Shuffled tests assignment correctness. Each contrast uses its own complete paired sample. Disagreement D alone establishes no benefit; interpret it with selected-value gain and uncertainty.

Table 2: **Same-state edit-selection contrasts** n/N: run/checkpoint targets or eligibility caps; $\Delta U_0, \Delta V_h$: p.p.; D: action disagreement (%). Brackets are working ranges, not CIs.

Contrast	n	N	ΔU_0	ΔV_h [range]	D [range]
Raw–Immediate	20	200	0	+2 [0, +4]	20 [10, 35]
SL–Raw	**20**	**200**	**0**	**+4** [+2, +6]	**30** [15, 45]
SL–Temporal	≤ 20	≤ 200	0	+3 [+1, +4]	25 [10, 40]
SL–Shuffled	≤ 20	≤ 200	0	+5 [+2, +8]	35 [20, 50]

Zero immediate centers and positive delayed differences illustrate a hypothesis, not an observed rank reversal. Immediate utility and later ADA share the private endpoint manifest; gate probes cannot replace it. E0 adds real-model delayed-benefit and reversal cases; fixtures test instrumentation only, and failed manipulations support no effectiveness claim.

All sampled checkpoints remain in the analysis, including identical choices, rejected edits, and abstentions. Implementation costs and remaining budgets are retained. Checkpoints cluster within their source runs; repeated continuation seeds reduce noise but do not create independent evolution replicates. Disagreement-only case studies are exploratory.

5.4 Which Components and Contexts Explain the Gain?

Table 3: **Component ablations** ADA and ΔSL are in p.p.

Axis / metric	SL full $H = 3$	Raw no links	Temporal	Shuffled	No relation labels	No adverse facts	Longer $H = 5$
Tools: ADA	6	2	3	1	5	4	6
Tools: ΔSL	ref.	−4	−3	−5	−1	−2	0
Skills: ADA	3	1	2	0	2	2	3
Skills: ΔSL	ref.	−2	−1	−3	−1	−1	0

Table 3 reports registered Tools/Skills audits; ΔSL is variant minus matched full SL. A poor shuffled score can reflect misinformation; functional specificity requires gains over both Raw and truthful Temporal. Removing adverse facts changes evidence as well as structure. The unchanged $H = 5$

center establishes no equivalence: longer H expands link eligibility while holding private horizon h fixed. Every ablation reruns full SL on its eligible checkpoints; overlapping audits provide no independent confirmation.

Reuse, remaining horizon, bottleneck presence, maintenance cost, and obsolescence define strata before ADA is read. Figure 2 separates reuse-specific ADA, overall selected value, and NO-EDIT frequency. A benefit confined to high reuse narrows the operating regime; repeatedly choosing costly infrastructure after reuse disappears challenges the mechanism. A sign reversal is possible, not required. Blinded trace audits separate actual use from visibility, link precision from missing coverage, and ineffective reasoning from failed control exposure.

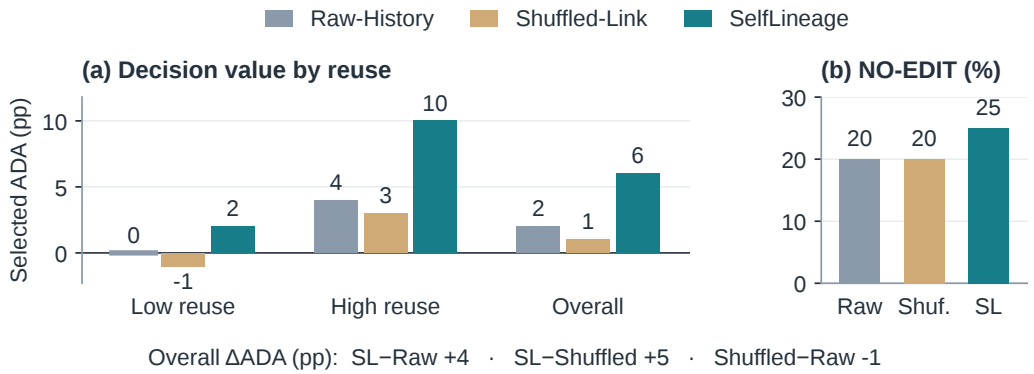

Figure 2: **Decision value across reuse conditions.** EOG Tools examples: (a) selected ADA, with equal stratum weights for the overall center; (b) NO-EDIT frequency. All arms share facts, cutoff, model, and gate.

The example SL–Raw gains are $6/2$ p.p. under high/low reuse; the registered stratum contrast requires matched coverage and fixed weights. NO-EDIT rates (25% SL, 20% Raw) describe behavior, not its cause: more abstention is useful only when selected value improves.

Runtime-only and evolution-only controls retain one relation family with shared facts, cutoff, gate, and continuation. With Raw and full SL, they form a factorial on checkpoints eligible for both types. Full-minus-single-family contrasts test incremental value; the interaction tests complementarity separately from Γ_h. Report eligibility and edge counts because removing a type reduces context. Unused allowance buys no extra facts or attempts. These contrasts have no supplied examples and enter the preregistered secondary family.

5.5 Does the Benefit Extend to Further Self-Improvement?

A useful initial edit need not improve further adaptation. Table 4(a) crosses Raw/SL decisions with fixed/adaptive private continuations. Each action is sampled and gated once; its realized state is copied across policies. Fixed continuation disables further edits; adaptive permits the common neutral improver. Future tasks, external stages, and slots are matched. Saved fixed-policy editing budget cannot buy extra task attempts.

Let g_h^c be the paired SL–Raw endpoint gap under continuation c. The further-edit interaction is

$$\Gamma_h = g_h^{\text{adaptive}} - g_h^{\text{fixed}}. \tag{10}$$

The example gives $g_h^{\text{fixed}} = 2$, $g_h^{\text{adaptive}} = 4$, and $\Gamma_h = 2$ p.p., with adaptive costs $0.94/0.90 B_{\text{cont}}$ (SL/Raw). Further-edit value requires paired support for the adaptive gap and interaction, with complete costs. Accepted-edit counts establish neither the interaction nor causal credit for ancestors.

For ongoing feedback, Live-Link reveals newly verified edges after a shared prefix; Frozen-Link retains prefix edges. Both record outcomes, self-edit, and pay for new links; hiding links saves no budget. Table 4(b) pairs $C/B = 0.98$ with $2/2$ p.p. quality/retention gains. Live exposure (40%) measures intervention opportunity, not beneficial-link frequency. Later histories may diverge.

Table 4: **Further-edit value and ongoing feedback** (a) 20 runs, ≤ 200 quadruples; (b) 20 paired forks. Q_h, Q, R, X in %, ADA in p.p.; E: accepted edits; X: new-edge exposure. Costs: $b_c = C/B_{\text{cont}}$ (private), $b = C/B$ (online).

(a) Fixed/adaptive continuation

Initial / policy	Q_h	ADA	E	b_c
Raw / fixed	32	1	off	0.60
SL / fixed	34	3	off	0.60
Raw / adaptive	35	2	1.2	0.90
SL / adaptive	39	6	1.5	0.94

(b) Live/frozen link updates

Policy	Q	R	AUC	b	X
Live	38	35	0.33	0.98	40
Frozen	36	33	0.31	0.98	off
Live–Frozen	+2	+2	+0.02	0	n/a

Interactions use complete matched quadruples; partial cells retain outcomes and costs without supplying a contrast. Live/frozen comparisons require newly rendered edges. Without that exposure, a null cannot distinguish unnecessary updating from an absent intervention.

5.6 Recursive Adaptation Requires Joint Evidence

Recursive adaptation requires agreement across distinct interventions. SL–Raw and SL–Temporal selected-value gains support functional relations in edit choice. The fixed gap g_h^{fixed} tests persistent task benefit; the adaptive gap g_h^{adaptive} and interaction Γ_h jointly test further-edit value. A positive fixed gap with an uncertain interaction supports useful initial edits while leaving improved adaptability unestablished. Shuffled degradation alone may reflect misinformation. System scores assess practical value; matched audits test relational specificity. System gains without a well-supported matched-audit benefit leave their source unresolved. Positive ADA with flat system quality calls for checking checkpoint coverage, available reuse, and downstream execution costs before generalizing the decision-level gain.

Continuous updating has a separate requirement: Live–Frozen gains with verified new-edge exposure. A precise null under adequate exposure weakens that hypothesis; low exposure makes the intervention inconclusive. Reuse-specific gains delimit where relations help. These interpretations require prespecified practical margins and paired run-level uncertainty, preserving nested checkpoints and repeated task identities. Report domain effects, complete costs, coverage, missingness, and censoring with each contrast; secondary tests use the registered Holm family.

Limitations. Reuse traces establish associations, not causal credit. Missing traces, salience, and misleading links can change decisions under node parity. ADA and Γ_h depend on the parent, continuation policy, horizon, and resources. Enterprise tasks and one backbone limit generality. The continuation experiment freezes internal editing; it does not isolate external expansion as the cause of forgetting. These limits remain even if endpoints improve.

6 Conclusion

SelfLineage combines immutable persistent-edit records, verified actual-use relations, and time-valid retrospective evidence to inform subsequent self-modification. Its hypothesis is that this representation helps a bounded, frozen agent choose more valuable edits from shared historical facts. The proposed EvoHarnessBench study connects broad memory and harness comparisons with same-state decision audits, relation ablations, and controlled continuations. Support requires selected-ADA gains over Raw, specificity beyond truthful temporal and shuffled controls, and resource-aware system improvements. Reuse strata test when relations help; fixed/adaptive continuations distinguish persistent task benefit from further-edit value, and live/frozen links test ongoing feedback. Skills and Agents assess scope across editable substrates.

AI USE STATEMENT

Generative AI coding assistants supported implementation, manuscript drafting and revision, review of mathematical formulations and experimental design, literature searches, reference formatting, and figure preparation. The authors remain responsible for the scientific claims, references, code, and figures.

REPRODUCIBILITY STATEMENT

Algorithm 1 specify factual nodes, time-valid linking, exact-token admission, relation controls, and the online information boundary. The appendices detail EvoHarnessBench adaptation/test isolation, benchmark checkpoint audits, analysis, and cost accounting. Confirmatory reports require exact dataset, model/backend, tokenizer, renderer, stream, gate, and evaluation manifests. The result registry distinguishes pending, unavailable, censored, and measured values.

Artifact release. Release of task or audit artifacts must respect applicable licenses and exclude protected solutions, credentials, and identifying infrastructure details.

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.

Suleyman Armagan Er, Danilo Ribeiro, Yogesh Virkar, Surafel Lakew, Adi Kalyanpur, James Gung, Thomas Delteil, and Arshit Gupta. Memtoolagent: Leveraging memory for tool using agents based on environment and user feedback. *arXiv preprint arXiv:2606.07909*, 2026.

Zixuan Ke, Vaidehi Patil, Haizhou Shi, Yang Li, Ye Liu, Sarath Shekkizhar, Anurag Koul, Jiayu Wang, Xuan Phi Nguyen, Semih Yavuz, et al. Evoharnessbench: Can your agents keep pace with an evolving harness? *arXiv preprint arXiv:2609.04280*, 2026.

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.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, et al. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent. *arXiv preprint arXiv:2504.15228*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Viktor Volkov, Valentin Khrulkov, Andrey V Galichin, Danil Sivtsov, Nikita Glazkov, Olga Volkova, Konstantin Pchelin, Iaroslav Bespalov, Dmitry V Dylov, Petr Anokhin, et al. Evomem: Memory-augmented evolution for code optimization. *arXiv preprint arXiv:2608.10795*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Wenyi Wang, Piotr Piękos, Li Nanbo, Firas Laakom, Yimeng Chen, Mateusz Ostaszewski, Mingchen Zhuge, and Jürgen Schmidhuber. Huxley-g\" odel machine: Human-level coding agent development by an approximation of the optimal self-improving machine. In *International Conference on Learning Representations*, volume 2026, pp. 80356–80386, 2026a.

Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hannaneh Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. Rethinking the evaluation of harness evolution for agents. In *COLM 2026 The 2nd Workshop on Lifelong Agents: Learning, Aligning, and Evolving*, 2026b.

Anyi Xu, B Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, CC Yu, et al. Deepseek-v4. 1-flash: Pushing the limits of kv cache compression. *arXiv preprint arXiv:2609.19969*, 2026a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026b.

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.

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.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Tong Zheng, Xidong Wu, Zheng Zhang, Zhankui He, Chaoyi Zhang, Benjamin Coleman, Ruoqiao Wei, Di Bai, Haolin Liu, Rui Liu, et al. Dream-rsi: Recursive self-improvement through evolving worlds. *arXiv preprint arXiv:2609.14858*, 2026.

Tailin Zhou. Hierarchical self-improvement: A framework for task-specific evolvable agent harnesses. *arXiv preprint arXiv:2608.08466*, 2026.