LONG-HORIZON INSURANCE CREDIT: MEASURING PREVENTIVE ACTIONS IN LONG-HORIZON TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents can sometimes make themselves more reliable by taking preventive actions, such as saving a checkpoint or writing down useful information. These actions may look useless when everything goes smoothly, but they matter a great deal if the agent later loses context, forgets earlier work, or encounters a failure. This makes preventive actions hard to evaluate: credit signals evaluated only under clean conditions can miss this value, while simple fault-injection tests can also give a misleading picture. We introduce LONG-HORIZON INSURANCE CREDIT to estimate the value of these actions. It measures how much an action helps when failures occur, then subtracts what the same action changes when nothing goes wrong. To estimate this quantity fairly, we compare matched runs driven by the same sampled fault tapes. In a controlled setting where the true answer is known exactly, insurance credit identifies useful preventive actions accurately (Kendall's $\tau = 0.92$), while standard clean-run methods miss them. We then test it on three tool-use benchmarks: ToolMaze, BFCL, and ALFWorld. We find that a single checkpoint can be extremely valuable, but current LLM agents usually fail to benefit from it because they rarely check the saved workspace after losing context. When the system makes them look once, much of the benefit returns, with insurance credit ranging from 6.4 to 20.8 percentage points relative to a length-matched placebo on ToolMaze and BFCL. This suggests that the main bottleneck is discovery rather than use: the agents can often exploit saved information once they look, but they rarely choose to look on their own. In this regime, system-level recovery design appears more reliable than relying on larger models alone.

1 INTRODUCTION

An agent pursuing a long-horizon goal in a fallible environment can spend effort protecting its future self: saving a checkpoint before a risky edit, writing down information it may need later, or preserving a fallback path (Risko & Gilbert, 2016; Packer et al., 2023). We call these *preventive* actions. They are central to reliable long-horizon behaviour, but they are also unusually hard to evaluate (Yan et al., 2026b;a). For the clean-neutral actions studied here, the action does not change clean task success relative to its control; its additional protective value is revealed under later faults. This motivates evaluating credit under more than one continuation condition. Credit assignment relates decisions to subsequent outcomes (Harutyunyan et al., 2019); for an unused checkpoint (Elnozahy et al., 2002) that leaves clean success unchanged relative to its control, the clean action effect is zero. A natural response is to inject failures and ask whether the checkpoint helps (Zhu et al., 2026; Wang et al., 2026). But naive fault injection can also be misleading, because the injected condition may change the trajectory in ways unrelated to the artifact itself. In our first uncontrolled test over 176 runs (GPT-4.1-mini, 11 tasks), checkpointing appeared to improve task success by $18.2\,\mathrm{pp}$ relative to the control arm (Appendix F). Yet a fake checkpoint of the same length, containing no useful information, improved success by exactly the same amount, showing that the apparent gain was not evidence that the checkpoint itself was helping (Appendix F). We therefore propose LONG-HORIZON INSURANCE CREDIT, which estimates the value of preventive actions by comparing matched branches under faults and under clean conditions. LONG-HORIZON INSURANCE CREDIT compares an action such as saving a checkpoint to a cost-matched control that does not provide real protection, measures how much more often the agent succeeds when faults occur, and then subtracts any benefit the action has when nothing goes wrong (§3). What remains is

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.

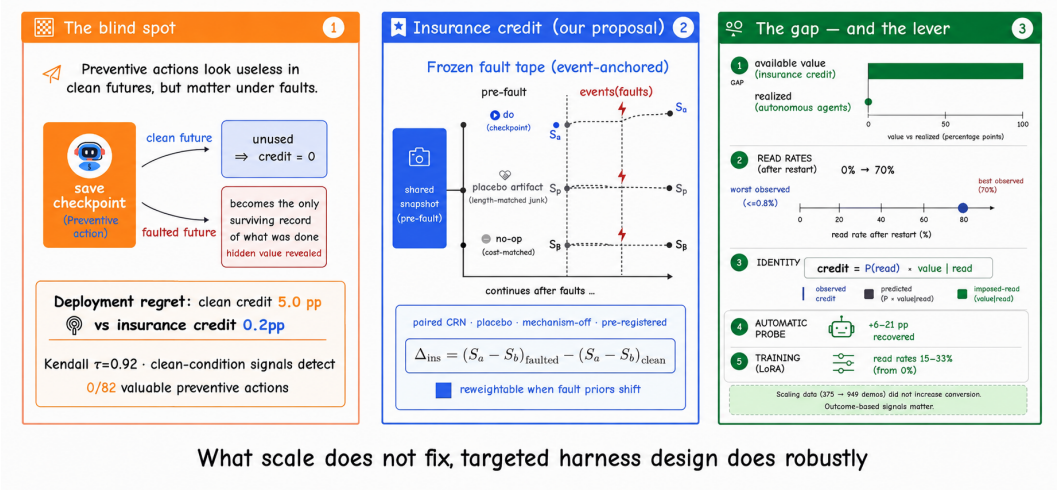

Figure 1: **Overview.** **(1)** Clean-condition comparisons miss the protective value of clean-neutral preventive actions. **(2)** Insurance credit compares action and placebo under shared fault tapes, then subtracts the clean effect. **(3)** Reference policies expose available recovery value; LLM agents often need the workspace to be surfaced. The schematic summarizes C2 and BFCL availability; C3 yields $+45.8$ pp (Table 2). Read-rate products are illustrative, subject to the conditions in §6.

the change in the action's effect under faults relative to clean conditions. To make this comparison fair, we run matched pairs of episodes with shared randomness (Glasserman & Yao, 1992) and include placebo controls, so that unrelated differences are not mistaken for the value of the preventive action itself (§3). Applying LONG-HORIZON INSURANCE CREDIT reveals a striking gap between available value and realized value. In our environments, a single checkpoint is highly valuable if the agent both needs it and knows to look for it after losing long context: a simple recovery policy gains 45.8 to 100 pp in task success from one checkpoint (Tables 2 and 3). Current LLM agents collect almost none of this value because they rarely inspect the workspace they themselves wrote (Tables 2 and 3). In ToolMaze, imposing a single read yields insurance credit of 15.7–20.3 pp relative to placebo across the four models tested, while a separate workspace-off check returns 0.0 pp for Qwen3-8B (Table 2). The main bottleneck is therefore discovery rather than use: once the information is found, the agents can often exploit it, but they rarely choose to look in the first place (§6). In the controlled WORKSPACEPROBE setting, stating in the tool description that the workspace survives restarts leaves the read rate at 0–3%, whereas placing the same information in the restart notice raises it to 67–97% (Table 6).

Contributions.

- **An estimand and a protocol.** We define LONG-HORIZON INSURANCE CREDIT, state the conditions for unbiased paired estimation when branches diverge structurally, and build an estimator with placebo controls and event-anchored fault tapes; these controls expose the same $+18.2$ pp false positive that appeared in our initial uncontrolled evaluation (§3; Appendix F).
- **A blind spot, validated against exact ground truth.** On BASINDAG (160 tasks, exact credit), clean counterfactual credit and pass@k detect $0/82$ valuable preventive actions, whereas LONG-HORIZON INSURANCE CREDIT detects all $82/82$, with 0.2 pp deployment regret versus 5.0 pp (§4).
- **A large value that current agents leave uncollected.** On three public benchmarks (Zhu et al., 2026; Patil et al., 2025; Shridhar et al., 2020), a checkpoint is worth up to $+100$ pp to a policy that uses it, yet the tested agents collect little of that value; a workspace-off evaluation provides a supporting mechanism check (§5).
- **A mechanism and two remedies.** The main bottleneck is discovery rather than use: the discovery ladder shows that both read rates and credit increase when the workspace is

surfaced at the point of need (Table 6). A harness probe after context loss yields $+6.4$–$20.8\,\mathrm{pp}$ (§6); LoRA (Hu et al., 2021) fine-tuning on 375 demonstrations raises the read rate from 0 to 15–33%, although the resulting success gains are seed-sensitive and do not improve with more data (§7).

2 Related work

Counterfactual credit for agent actions. The closest line of work evaluates an action by replacing it with a counterfactual and rolling out the remainder of the episode under a frozen continuation policy. Recent examples include PCCC (Meng, 2026), C3 (Chen et al., 2026), CRAFT (Meng & Chen, 2026), and Causal Agent Replay (Shah, 2026). Our distinction is the fault-conditioned, clean-subtracted estimand. Our clean-counterfactual baseline evaluates action effects without subsequent faults; in the clean-neutral oracle setting, it detects $0/82$ valuable preventive actions (§4). Classical counterfactual credit (Harutyunyan et al., 2019; Meulemans et al., 2023), prefix-scoring by weaker continuers (Zhou et al., 2026), Monte-Carlo pass@k (Che et al., 2026), and calibrated process rewards (Li et al., 2026) are related outcome-based signals with the same clean-condition blind spot here.

Credit for memory and state-writing actions. A second nearby literature studies memory updates and other state-writing actions. Memory-R2 (Yan et al., 2026b) re-rolls from a shared memory state, and HiMPO (Yan et al., 2026a) scores memory updates by the task information they preserve. Our focus is narrower: we compare action effects under a declared distribution of subsequent faults with their clean-condition effects. The distinction is this insurance-credit profile.

Robustness benchmarks. Another relevant line of work studies robustness at the episode level rather than the action level. ToolMaze (Zhu et al., 2026), AgentNoiseBench (Wang et al., 2026), and MAS-FIRE (Jia et al., 2026) inject faults and report overall degradation; Khanal et al. (2026) formalise reliability estimands observationally; Recovery-Bench (Tan et al., 2025) replays corrupted states. These benchmarks are useful for asking whether an agent remains reliable under perturbation, but they do not attribute that robustness to a specific earlier action. That is the gap we target here. Empirically, we build on ToolMaze as one of our external environments.

Option value and preservation. At a broader conceptual level, empowerment (Mohamed & Jimenez Rezende, 2015), attainable-utility preservation (Turner et al., 2020), recovery RL (Thananjeyan et al., 2021), and action-robust RL (Tessler et al., 2019) all formalise keeping futures open or performing well under perturbation. These frameworks operate at the level of policies or general objectives. Insurance credit is narrower: it is an action-level quantity tied to a particular success event and a declared fault distribution.

3 Insurance credit

Definition. Fix a history h, a candidate action a, and a cost-matched control b. Let ν be a distribution over continuation policies and κ a distribution over exogenous fault tapes whose events are scheduled *after* the action. Write

$$S(a;\kappa) = \mathbb{E}_{\pi\sim\nu,\ \xi\sim\kappa}\left[Y \mid do(a), h, \pi, \xi\right] \tag{1}$$

for the induced success probability, with $Y \in \{0,1\}$. The total effect and the insurance credit are

$$\Delta_{\mathrm{tot}}(\kappa) = S(a;\kappa) - S(b;\kappa), \qquad \Delta_{\mathrm{ins}}(\kappa) = \Delta_{\mathrm{tot}}(\kappa) - \Delta_{\mathrm{tot}}(\varnothing), \tag{2}$$

where $\varnothing$ denotes the clean condition. Insurance credit measures how the action–control effect changes under faults relative to clean conditions. Our primary object is the *profile* of Δ_{ins} over fault types and severities; any scalar summary integrates that profile against a probability distribution w over deployment fault conditions,

$$\Delta_w = \int \Delta_{\mathrm{ins}}(\kappa)\,\mathrm{d}w(\kappa). \tag{3}$$

Relation to fault-augmented value. For fixed ν and a probability distribution w, define the value in an MDP augmented with fault stochasticity by

$$\tilde{Q}_w(h,x) = \mathbb{E}_{\pi\sim\nu,\,\kappa\sim w,\,\xi\sim\kappa}\big[Y \mid do(x), h, \pi, \xi\big], \qquad \Delta_w = \tilde{Q}_w(h,a) - \tilde{Q}_w(h,b) - \Delta_{\mathrm{tot}}(\varnothing). \tag{4}$$

The insurance-credit summary equals the fault-augmented value difference only when the action–control pair has zero clean-condition effect. We therefore carry $\tilde{Q}_w$ as a baseline throughout. The profile can be re-weighted over measured fault conditions when w changes (Table 7).

Estimation with keyed tapes. For each pair (t, s) of task and pair seed, we execute both branches from a common snapshot with the branch action *imposed* — $do(a)$ rather than a choice — under a single frozen fault tape $\xi_{t,s}$ drawn before either branch runs. This gives the paired plug-in estimator

$$\widehat{\Delta}_{\mathrm{ins}} = \frac{1}{|P|}\sum_{(t,s)\in P}\big(Y^a_{t,s,\kappa} - Y^b_{t,s,\kappa}\big) - \widehat{\Delta}_{\mathrm{tot}}(\varnothing), \tag{5}$$

where $Y^x_{t,s,\kappa}$ is the outcome under action x and tape $\xi_{t,s}$, and $\widehat{\Delta}_{\mathrm{tot}}(\varnothing)$ is the mean action–control contrast in clean runs, using the same target task weights. The pooled estimator targets the corresponding average of task-specific insurance credits. This is a common-random-number construction (Glasserman & Yao, 1992). Sharing ξ affects the covariance between branch outcomes,

$$\mathrm{Var}(Y^a - Y^b) = \mathrm{Var}\,Y^a + \mathrm{Var}\,Y^b - 2\,\mathrm{Cov}(Y^a, Y^b), \tag{6}$$

which is the source of the 89% variance reduction reported in §4. The crucial implementation detail is how tapes are keyed. We bind them to *designated semantic events* — for example, the first post-fork call of a named task tool, if it occurs — rather than to a global call index.

That detail matters. In our first implementation, faults were keyed to the global post-fork call index. Because an efficient recoverer makes *fewer* calls, that convention silently shifted faults onto its task-critical calls, changing measured checkpoint credit from $+13.5\,\mathrm{pp}$ under semantic keying to $-12.5\,\mathrm{pp}$ under index keying. The keying rule is therefore part of the fault intervention being measured. Semantic anchoring specifies the intended event more clearly, but does not guarantee identical realized exposure when branches skip tools (Appendix A).

Controls. Each comparison includes four controls. We use a cost-matched no-op (token and tool-call matched within $\pm 10\%$), a *placebo* artifact of matched length with no usable content, a non-fault stratum that checks the clean action effect, and a mechanism-off ablation in which the artifact exists but cannot be read. Success is the strict conjunction

$$Y = \mathbf{1}\{\text{tools complete, cleanly}\} \cdot \mathbf{1}\{\text{no duplicated side effect}\} \cdot \mathbf{1}\{r \geq \tfrac{1}{2}\}, \tag{7}$$

with r the retained share of the task's world-information steps. The judge was calibrated on clean controls and audited on 96 traced episodes. Blind re-annotation of a stratified 100-episode sample gives Cohen's $\kappa = 0.82$ $[0.70, 0.92]$, with 98% agreement in the wipe stratum. The audit did not reveal a clear branch imbalance in disagreements, but symmetric misclassification can still attenuate effect sizes (Appendix B).

4 Does the estimator recover the truth?

BasinDAG is a family of finite, fully enumerable agent environments in which every action's insurance credit is computable exactly by dynamic programming. Its state includes progress, a checkpoint slot, durable memory, a fallback flag, and corruption and blocking states; continuations follow a rule policy with an action-error rate, and faults include corruption, information loss, and path blocking. The generator produces valuable, useless, and *harmful* preventive actions — for example, checkpointing an already-corrupted state has genuinely negative credit — so a signal cannot score well by always predicting a positive sign. We generate 160 tasks, hold out a fifth of the structural families, and validate the oracle against exhaustive enumeration to 10^{-9}.

Table 1 makes the blind spot precise. One part of the result follows from the oracle setup: these preventive actions do not change true clean success, so their population clean action effect is zero,

Table 1: Signals evaluated against exact ground truth on BASINDAG (160 tasks). "Detects" counts the $n = 82$ preventive actions whose true insurance credit exceeds $5\,\mathrm{pp}$ that the signal scores above $2\,\mathrm{pp}$. The detection column should be read descriptively: the preventive actions considered here leave true clean success unchanged, so their population clean action effect is zero; finite-sample or numerical estimates need not be exactly zero. The two right columns report placebo false-positive counts and deployment regret, respectively. Deployment regret is the loss in success from acting on the signal's ranking.

Signal	Detects	Mean credit	Placebo FP	Deploy. regret
Insurance credit (ours, 16 pairs)	**82/82**	$+11.8\,\mathrm{pp}$	0	$0.2\,\mathrm{pp}$
Fault-augmented Q	82/82	$+11.6\,\mathrm{pp}$	0	$0.01\,\mathrm{pp}$
Clean counterfactual credit	**0/82**	$-0.001\,\mathrm{pp}$	0	$5.0\,\mathrm{pp}$
pass@4	**0/82**	≈ 0	0	$5.0\,\mathrm{pp}$
pass@8	**0/82**	0	0	$5.4\,\mathrm{pp}$

while our estimator is designed to recover a fault-conditioned quantity. What is *not* guaranteed by definition is finite-sample recovery of sign and ranking, including negative-credit templates that a sign-agnostic method could not fake ($\tau = 0.92$), and downstream decision quality. We quantify the latter by deployment regret, defined for a signal σ as

$$\mathrm{Reg}(\sigma) = \mathbb{E}_T\big[\, S_T(a_T^\star) - S_T(\hat{a}_T^\sigma) \,\big], \qquad \hat{a}_T^\sigma = \arg\max_{x \in \mathcal{A}_T} \sigma(x, T), \tag{8}$$

the realized success forfeited by acting on σ rather than the oracle-best action. By that measure, acting on insurance credit incurs $0.2\,\mathrm{pp}$ regret rather than the $5.0\,\mathrm{pp}$ incurred by clean counterfactual credit. The role of the oracle world is to validate the instrument; the substantive claims then come from the real-agent experiments in §5–§7. The estimator itself is well behaved: at sixteen paired rollouts per cell, mean absolute error is 0.008, sign accuracy is 0.92, Kendall τ is 0.92, held-out structural families give 0.91 sign accuracy, compared with 0.92 overall, and keyed-tape pairing reduces estimator variance by 89% relative to independent sampling.

A separate question is what the profile adds over the fault-augmented Q baseline, which nearly matches it in Table 1. The answer is a boundary rather than a claim of dominance. When the deployment fault distribution is known and stationary, the scalar Q is the right object: it wins narrowly on regret (0.01 vs. $0.2\,\mathrm{pp}$) and is simpler to use. The profile earns its keep when that distribution is unknown or misspecified, which is often the relevant deployment case. In that setting, re-weighting the measured profile under shifted fault priors incurs $0.31\,\mathrm{pp}$ mean decision regret, whereas the fixed-prior scalar — which cannot be re-weighted after the fact — incurs $1.23\,\mathrm{pp}$; a minimax rule over the profile over-hedges and does worse ($4.52\,\mathrm{pp}$). Measuring the profile requires no additional rollouts: it is simply the disaggregation of what the scalar already averages.

5 WHAT ARE PREVENTIVE ACTIONS WORTH TO REAL AGENTS?

Environments. We study ToolMaze (Zhu et al., 2026) complexity families C2 and C3 using GPT-4.1-mini, the open-weights Qwen3-8B (Yang et al., 2025), and two Gemini-3-generation models (gemini-3-flash-preview and gemini-3.5-flash). GPT-4.1 is additionally evaluated in the WORKSPACEPROBE ladder (§6). We harden the environment so that information loss has real consequences: truncation removes the *oldest* exchanges, each data source has a session-level request budget, the final answer must contain the requested information, and a restart severity in which everything after the task statement is lost while the workspace survives models a crash-and-resume event. All hardenings were frozen before any pre-registered-model data was collected (Appendix C).

One design choice matters enough to foreground. The possible value of an artifact is bounded by the share of required information it protects. With an early fork, the checkpoint contained only about a fifth of what the task needed, and the measured effect was near zero for every agent *and* for the reference policy — a structural null, which we report. We therefore focus in the main text on the configuration in which the artifact carries a decisive share of the task state (fork after ≈ 0.6 of the shortest valid path) and treat information share as an explicit axis.

Table 2: Insurance credit in the hardened tool-use environment, sharp configuration; paired contrasts vs. length-matched placebo, task-clustered bootstrap 95% CIs. "Reads" = episodes with any workspace access.

Agent	Condition	Insurance credit	Reads	Clean succ.	Tasks / eps
Reference policy	C2, autonomous	$+56.2$ pp $[+44.8, +67.7]$	384/1536	1.00	48 / 1536
Reference policy	C3, autonomous	$+45.8$ pp $[+29.2, +64.6]$	192/768	1.00	24 / 768
GPT-4.1-mini	C2, autonomous	$+0.0$ pp $[-7.5, +7.5]$	**0/640**	0.49	20 / 640
Qwen3-8B	C2, autonomous	-1.4 pp $[-4.2, +0.0]$	**0/1152**	0.24	36 / 1152
Qwen3-8B	C3, autonomous	$+0.0$ pp $[+0.0, +0.0]$	**0/1536**	0.28	48 / 1536
Gemini-3-flash-prev.	C2, autonomous	$+2.3$ pp $[-2.3, +6.9]$	2/714	0.64	23 / 714
Gemini-3.5-flash	C2, autonomous	$+1.0$ pp $[-3.1, +6.2]$	37/768	0.50	24 / 768
GPT-4.1-mini	read imposed	$+15.7$ pp $[+6.7, +24.6]$	436/2144	0.52	67 / 2144
Qwen3-8B	read imposed	$+16.7$ pp $[+4.2, +31.2]$	175/768	0.26	24 / 768
Gemini-3-flash-prev.	read imposed	$+20.3$ pp $[+10.3, +31.5]$	150/638	0.69	23 / 638
Gemini-3.5-flash	read imposed	$+17.2$ pp $[+1.6, +33.3]$	283/1276	0.61	24 / 1276
Qwen3-8B	imposed, ws. off	$+0.0$ pp $[+0.0, +0.0]$	96/384	0.26	12 / 384

"Clean succ." is the no-op branch success rate at fault severity $\rho = 0$ in the same run — the agent's unperturbed competence on these tasks. Entries of $+0.0$ with degenerate intervals $[0, 0]$ arise because every observed paired contrast is zero in the available samples. Gemini episodes lost to API rate limits are excluded as errors (13% worst case, branch-symmetric); other agents are unaffected.

The value at stake. We first ask what the environment makes available in principle. To do so, we run a fixed reference policy that behaves as a well-engineered recovery procedure would: on detecting amnesia, it probes the workspace once, reinstates what it recovers, and redoes only what it cannot account for. This is not a learned agent, but an availability control for the question "is there anything here to collect?" The answer is yes. A single checkpoint yields $+56.2$ pp $[+44.8, +67.7]$ on C2 and $+45.8$ pp $[+29.2, +64.6]$ on C3; no-op success falls from 1.00 to 0.41 under restart, and the checkpoint branch recovers most of that loss. The placebo and non-fault strata are exactly 0.000. Crucially, removing the additional tool faults leaves checkpoint gains of $+56.2$ pp on C2 and $+37.5$ pp on C3: substantial recovery value persists under restart alone (Table 5).

The artifact type also matters. A checkpoint that records *which steps ran, with outputs* carries essentially the entire effect, whereas a durable note that records *facts without provenance* carries none: an agent that recovers what it learned but not what it already did simply re-executes, exhausts the session budget, and fails. This is a property of the artifact, not of the reader, and it carries a harness-design lesson.

The value collected. Table 2 contains the paper's central empirical result. In the same environments, under the same frozen tapes and the same judge, the four autonomous models listed have insurance-credit intervals that include or touch zero. The reason is legible. GPT-4.1-mini and Qwen3-8B record no workspace accesses in the autonomous conditions listed in Table 2. The two Gemini-3-generation agents, despite being trained for agentic tool use, probe only rarely ($2/355$ and $37/384$ of post-restart episodes) and do so indiscriminately: their read rates are similar in the checkpoint, placebo, and no-op branches. Low read rates accompany small realized credit, consistent with a discovery bottleneck.

That failure is worth describing carefully. After the restart, the agent does not remember that anything was written: the write was imposed by $do(a)$, and the conversational trace containing it is gone. The problem is therefore not that the agent "forgets its own notes," but that it never entertains the possibility that persistent state might exist. The workspace tool is present throughout, and in §6 even explicitly says that entries survive restarts, yet the agents still do not look. Nor is this simply a competence issue. Table 2 shows that the strongest agents in clean success (0.49–0.64) still read essentially never, the reference policy reads in the identical harness, and under training (§7) reads and clean success rise *together*.

Nor is this an inability to use the artifact once found. Imposing a single read after the restart — one harness-level tool call, with no instruction and no coaching — recovers $+15.7$ pp $[+6.7, +24.6]$

Table 3: BFCL v4 multi-turn (second public lineage), sharp protocol: paired ckpt-vs-placebo credit at $\rho = 2$, task-clustered 95% CIs; "reads" = post-restart episodes with any workspace access.

	autonomous		read imposed	
Agent	credit	reads	credit	reads
Reference policy	$+\mathbf{100.0}$ pp	384/384	—	—
GPT-4.1-mini	$+6.2$ pp $[+2.1, +11.5]^{\dagger}$	3/382	$+7.8$ pp $[+0.7, +16.1]$	612/612
Qwen3-8B	$+0.0$ pp $[0.0, 0.0]$	**0/768**	$+6.4$ pp $[-1.3, +16.0]$	1000/1000
Gemini-3-flash-prev.	$+2.1$ pp $[0.0, +6.2]$	18/381	$+20.8$ pp $[+7.3, +36.5]$	384/384
Gemini-3.5-flash	$+6.4$ pp $[+1.1, +13.8]$	**267/381**	$+17.2$ pp $[+6.5, +30.2]$	379/379

†The positive contrast coexists with a low read rate; see Appendix D for mechanism-off details. Clean success (no-op, $\rho = 0$): reference 1.00, mini 0.42, Qwen 0.34, g3fp 0.54, g35f 0.48.

for GPT-4.1-mini and $+16.7$ pp $[+4.2, +31.2]$ for Qwen3-8B against the length-matched placebo. For the Gemini agents, the same imposed read yields $+20.3$ pp $[+10.3, +31.5]$ (gemini-3-flash-preview) and $+17.2$ pp $[+1.6, +33.3]$ (gemini-3.5-flash). A separate Qwen3-8B workspace-off evaluation returns $+0.0$ pp, and a matched workspace intervention on 23 ToolMaze tasks increases the checkpoint–placebo contrast by 17.4 pp $[+4.3, +32.6]$ (Appendix A), providing a more direct mechanism check.

Additional public lineages. To test whether this picture is specific to ToolMaze, we port the full harness — $do(a)$ branches, frozen tapes, placebo, mechanism-off, and strict judge — to BFCL v4 multi-turn (Patil et al., 2025), a stateful API benchmark with exact state verification and a task distribution disjoint from ToolMaze's. Here the cost of restart is re-executing *irreversible* calls, which the state judge catches natively (protocol in Appendix D). The main picture replicates and sharpens: $+100$ pp is available to the reference policy (placebo exactly 0), GPT-4.1-mini and Qwen read in only 0.8% and 0% of post-restart episodes, and imposed-read point estimates range from $+6.4$ to $+20.8$ pp. BFCL adds two further observations. First, reading *junk* is not obviously free: under imposed read, the mini model's placebo-vs.-no-op contrast is -6.2 pp $[-12.5, -1.0]$ on the initial 24 tasks, attenuating to -3.2 pp $[-7.7, +0.6]$ at 39. Second, discovery is environment-dependent: gemini-3.5-flash, at $\leq$ 10% reads on ToolMaze, reads 70% on BFCL and realizes $+6.4$ pp $[+1.1, +13.8]$. The instrument registers progress where progress exists.

ALFWorld (Shridhar et al., 2020) plays a different role. It replicates availability exactly ($+100$ pp to the reference policy over 56 games, placebo 0), but it also marks the boundary of the phenomenon. There, completed work is cheaply re-observable from the world itself, so re-exploration can substitute for the artifact and the informative stratum collapses from 56 to 5 games. On that stratum, GPT-4.1-mini reads $0/80$ autonomously, and an imposed read yields $+35$ pp $[0, +75]$ (supplementary). Taken together, the three lineages locate where the gap matters most: when completed work is *not* cheaply re-readable from the environment — destroyed information in ToolMaze, invisible irreversible state in BFCL — preventive artifacts carry large recoverable value. Where the world itself remains an easy source of recovery, that gap relaxes (Appendix D).

6 Why is nothing collected? A discovery ladder

If agents can use artifacts once they read them but almost never read them, then the deficit sits at discovery. We test that hypothesis in a setting designed to remove benchmark complexity. WorkspaceProbe uses three deterministic data tools, each of which answers once per session. The agent must report all three facts, and a session restart erases the conversation while leaving the workspace as the only surviving source. The task is deliberately simple, `read_workspace` remains available throughout, and the five intervention levels vary only *where* the affordance is surfaced.

Figure 2 shows how discovery and credit change across interventions (full ladder table in Appendix D). Let $D = (Y^a_\kappa - Y^b_\kappa) - (Y^a_0 - Y^b_0)$ be the paired difference-in-differences, let R indicate that at least one of these four rollouts reads the workspace, and write $p_R = P(R)$. The law of total

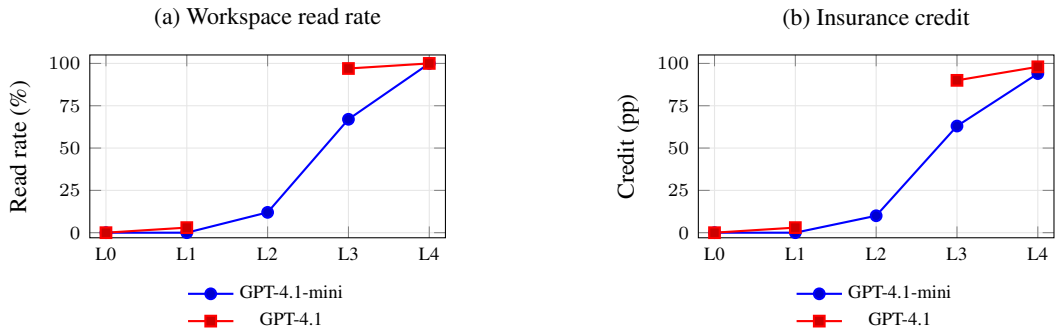

Figure 2: WORKSPACEPROBE discovery ladder. **(a)** Workspace read rates. **(b)** Insurance credit. L0: tool list only; L1: persistence stated in the tool description; L2: workspace mentioned in the task; L3: workspace mentioned in the restart notice; L4: read imposed by the harness. The unreported GPT-4.1 L2 condition is left blank.

expectation gives

$$\Delta_{\text{ins}} = p_R \, \mathbb{E}[D \mid R] + (1 - p_R) \, \mathbb{E}[D \mid \neg R]. \tag{9}$$

The second term is zero only if unread artifacts have no effect on the paired contrast, and the pair-level probability p_R is not automatically the episode-level read rate reported in Tables 2–6.

The ladder supports a discovery bottleneck: credit increases as agents are prompted to inspect the workspace, and the imposed-read rung reaches $94\,\text{pp}$ for GPT-4.1-mini and $98\,\text{pp}$ for GPT-4.1. The placebo contrast is zero at each level. Using imposed-read effects in a read-rate product is therefore best understood as a heuristic, not as a consequence of Eq. (9) alone.

Two contrasts matter most. Stating in the tool description that entries survive restarts — the affordance where it is *defined* — moves the read rate from 0.00 to only $0.00/0.03$. Presenting the same information in the restart notice — the point at which it becomes *needed* — moves it to $0.67/0.97$. The stronger model is no more likely to look unprompted; capability improves responsiveness to a cue, not the underlying disposition to check for persistent state.

The prompt-level lever does not survive realistic complexity. The L3 cue is decisive in WORKSPACEPROBE, so we inject the same hint into the hardened benchmark's restart notice. There, the cue does not induce reading: reads are $0/960$. The reported contrast is $+3.3\,\text{pp}$ $[+0.0, +8.3]$; this residual cannot be attributed to workspace retrieval. Amid a dozen task tools, live quota errors, and a much busier task state, the agent that follows the cue in a three-tool world never acts on it.

What *does* transfer is one level lower in the stack: the harness itself issuing a single workspace probe after context loss. That intervention yields point estimates of $+6.4$–$20.8\,\text{pp}$ against placebo across the tested ToolMaze and BFCL conditions (Tables 2, 3). The practical implication is that recovery from context loss should be treated as an automatic harness step rather than as model initiative. In this regime, neither a larger model nor a better prompt substitutes for that systems-level design choice.

7 CLOSING THE GAP BY TRAINING

If the gap is behavioural rather than purely capability-limited, then a modest amount of the right supervision might repair it. We test that idea by fine-tuning Qwen3-8B with LoRA on 375 judge-passed demonstrations: 133 recovery episodes (16 replayed from the model's own imposed-read successes and 117 rendered from the reference policy), 140 clean-contrast episodes (no truncation, correctly no read), and 102 coverage episodes (junk artifact, empty workspace, memory-note). Training uses tasks C2 001–048 only, masks loss to post-notice assistant tokens, and leaves the evaluation prompt unchanged.

Table 4 shows that workspace discovery can be learned. For the primary seed, reported read rates rise from zero to $31\%/33\%$ on held-out C2/unseen C3, with fault-condition checkpoint–placebo

Table 4: Recovery SFT, autonomous arm, frozen sharp protocol; C2 049–084 held out, C3 never seen in training.

Model	Evaluation	read rate	ckpt vs. placebo	clean success
Base Qwen3-8B	held-out C2	0 %	+0.0 pp [+0.0, +0.0]	0.18
SFT, seed 17 (primary)	held-out C2	**31 %**	**+16.2** pp [+5.9, +26.5]	**0.33**
SFT, seed 42	held-out C2	17 %	+5.0 pp [+0.0, +11.7]	0.15
SFT, seed 123	held-out C2	15 %	+2.1 pp [+0.0, +6.2]	0.11
Base Qwen3-8B	C3 (unseen)	0 %	+0.0 pp [+0.0, +0.0]	0.28
SFT, seed 17 (primary)	C3 (unseen)	**33 %**	**+16.7** pp [+4.2, +31.2]	0.27
SFT, seed 42	C3 (unseen)	24 %	+0.0 pp [+0.0, +0.0]	0.15
SFT, seed 123	C3 (unseen)	16 %	+0.0 pp [+0.0, +0.0]	0.23

contrasts of $+16.2/+16.7$ pp; held-out C2 clean success also increases. Both additional seeds raise reading to 15–24%, although their recovery gains are less consistent. These fault contrasts alone do not establish an improvement in clean-subtracted insurance credit.

Scaling to 689 and 949 demonstrations does not consistently improve conversion. Recovery upweighting restores reading in one run but not stable recovery; the full scaling grid is in Appendix D. Thus demonstrations can install the disposition to look, while converting access into reliable recovery remains a training challenge. The harness probe provides the directly tested systems intervention.

8 Does measuring this change decisions?

A natural question is whether insurance credit matters beyond measurement. We evaluate that directly using the deployment-regret objective of Eq. (8) on the reference-policy tool-use evaluation (two-fold cross-validation, 96 folds, 48 tasks; Appendix E). With the same random tie-breaking rule for both selectors, acting on insurance credit reaches 0.490 deployment success $[0.362, 0.620]$, against an oracle-best upper bound of 0.615. Clean counterfactual credit reaches 0.193 $[0.148, 0.240]$; its action scores are tied in this setting. Thus measuring fault-conditioned value helps select preventive actions that improve realized deployment success.

9 Discussion and limitations

Main takeaway. Preventive actions can carry substantial recovery value that clean-condition credit misses and agents leave uncollected. Insurance credit measures this value through fault-conditioned, clean-subtracted comparisons. Placebo controls expose the initial $+18.2$ pp false positive, while semantic fault tapes make the intervention explicit (Appendices F and A).

Practical implication. Workspace discovery is an actionable bottleneck in the tested settings. Agents can often exploit saved information once it is surfaced; matched workspace and restart-only controls strengthen the recovery interpretation. Automatic checks of persistent state therefore offer a concrete harness intervention after context loss.

Limitations and scope. The reference policy establishes availability, not achievability by an unconstrained learner. SFT improves discovery more consistently than conversion; extra data alone does not stabilize recovery. Free-text notes and shallow checkpoints provide little value in the tested configurations, and single-trajectory prediction does not transfer from the oracle to real-agent evaluations. The observed gap is largest when completed work cannot be cheaply re-observed; easier environmental recovery reduces the role of persistent artifacts.

AI Assistance Disclosure

Large language models were used as a writing aid in preparing this manuscript: drafting and polishing portions of the text, including figure captions. LLMs also assisted with checking mathematical notation and the consistency of theoretical and statistical arguments. They were not used for research ideation, experimental design, implementation, or generating experimental results. All LLM-assisted text was reviewed and verified by the authors, who take full responsibility for the content of the paper. The LLM agents evaluated in §5–§7 are objects of study, not authorship tools.

References

Fengdi Che, Yang Liu, Lei Yu, Meng Cao, Tong Che, Rupam Mahmood, and Dale Schuurmans. Learning with a single rollout via monte carlo pass@ k critic. *arXiv preprint arXiv:2606.25451*, 2026.

Yanjun Chen, Yirong Sun, Hanlin Wang, Jinghan Wang, Xinming Zhang, Xiaoyu Shen, Wenjie Li, and Wei Zhang. Exact is easier: Credit assignment for cooperative llm agents. *arXiv preprint arXiv:2603.06859*, 2026.

Elmootazbellah Nabil Elnozahy, Lorenzo Alvisi, Yi-Min Wang, and David B Johnson. A survey of rollback-recovery protocols in message-passing systems. *ACM Computing Surveys (CSUR)*, 34 (3):375–408, 2002.

Paul Glasserman and David D Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jin Jia, Zhiling Deng, Zhuangbin Chen, Yingqi Wang, and Zibin Zheng. Mas-fire: Fault injection and reliability evaluation for llm-based multi-agent systems. *arXiv preprint arXiv:2602.19843*, 2026.

Aaditya Khanal, Yangyang Tao, and Junxiu Zhou. Beyond pass@ 1: A reliability science framework for long-horizon llm agents. *arXiv preprint arXiv:2603.29231*, 2026.

Jinyuan Li, Langlin Huang, Chengsong Huang, Shaoyang Xu, Donghong Cai, Yuyi Yang, Wenxuan Zhang, and Jiaxin Huang. Process rewards with learned reliability. *arXiv preprint arXiv:2605.15529*, 2026.

Renwei Meng. Policy-conditioned counterfactual credit for verifiable reinforcement learning of long-horizon language agents. *arXiv preprint arXiv:2606.05263*, 2026.

Zibin Meng and Kani Chen. Craft: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2606.29476*, 2026.

Alexander Meulemans, Simon Schug, Seijin Kobayashi, Gregory Wayne, et al. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. *Advances in Neural Information Processing Systems*, 36:68685–68735, 2023.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

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.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Evan F Risko and Sam J Gilbert. Cognitive offloading. *Trends in cognitive sciences*, 20(9):676–688, 2016.

Jaineet Shah. Causal agent replay: Counterfactual attribution for llm-agent failures. *arXiv preprint arXiv:2606.08275*, 2026.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Shangyin Tan, Kevin Lin, Koushik Sen, and Matei Zaharia. Recovery-bench: Evaluating agentic recovery from mistakes. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224. PMLR, 2019.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6 (3):4915–4922, 2021.

Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. Conservative agency via attainable utility preservation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 385–391, 2020.

Ruipeng Wang, Yuxin Chen, Yukai Wang, Chang Wu, Junfeng Fang, Xiaodong Cai, Qi Gu, Hui Su, An Zhang, Xiang Wang, et al. Agentnoisebench: Benchmarking robustness of tool-using llm agents under noisy condition. *arXiv preprint arXiv:2602.11348*, 2026.

Jiangze Yan, Yi Shen, Wenjing Zhang, Jieyun Huang, Zhaoxiang Liu, Ning Wang, Kai Wang, and Shiguo Lian. Himpo: Hindsight-informed memory policy optimization for less-entangled credit in long-horizon agents. *arXiv preprint arXiv:2606.16285*, 2026a.

Sikuan Yan, Ahmed Bahloul, Ercong Nie, Susanna Schwarzmann, Riccardo Trivisonno, Volker Tresp, and Yunpu Ma. Memory-r2: Fair credit assignment for long-horizon memory-augmented llm agents. *arXiv preprint arXiv:2605.21768*, 2026b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuhang Zhou, Yixin Cao, and Guangnan Ye. From correctness to utility: Gain-based prefix evaluation for llm reasoning. *arXiv preprint arXiv:2606.07190*, 2026.

Dongsheng Zhu, Xuchen Ma, Yucheng Shen, Xiang Li, Yukun Zhao, Shuaiqiang Wang, Lingyong Yan, and Dawei Yin. When tools fail: Benchmarking dynamic replanning and anomaly recovery in llm agents. *arXiv preprint arXiv:2606.05806*, 2026.

A FAULT TAPES AND IDENTIFICATION UNDER STRUCTURAL DIVERGENCE

Formal statement. Let $\mathcal{E}$ be the set of anchor events (the first post-fork invocation of the hash-designated task tool), and let a tape ξ map $\mathcal{E}$ (and continuation rounds, for context loss) to fault payloads. Each tape is drawn once from the declared κ and then frozen.

Proposition 1 (unbiasedness for a declared tape intervention). *For each task and fault condition, assume that each branch has the marginal rollout law defining $S(x;\kappa)$, including continuation-policy and environment randomness, and that the clean contrast is estimated without bias using the same target task weights. Then Eq.* (5) *is unbiased for the correspondingly weighted insurance credit. Branches may differ in length, call order, call count, and realized fault exposure.*

Proof sketch. Correct branch marginals give $\mathbb{E}[Y^a_{t,s,\kappa} - Y^b_{t,s,\kappa}] = S_t(a;\kappa) - S_t(b;\kappa)$, regardless of the coupling between branches. Averaging with the target task weights and subtracting the unbiased clean contrast gives the claim by linearity. Sharing a tape changes the covariance and hence the estimator variance; it is not itself a substitute for correct marginal rollout laws. □

This proposition concerns the intervention defined by the declared tape policy. Requiring the same anchor to occur in both branches would be a stronger condition, needed to interpret every pair as experiencing identical realized fault exposure. Semantic keying does not guarantee that condition when one branch skips the target tool; the alignment audit below records such cases.

Construction. For each (task, pair seed) a tape is drawn once, before either branch runs, and frozen. A tape contains at most ρ events. Context-loss events are keyed by continuation round. Tool-level events (silent corruption, timeout, argument slip) are keyed to a *designated semantic event*: a task-path tool is selected by hash, and the fault fires on the first post-fork call of that tool if the branch reaches it, in whichever branch, at whatever call index it occurs. Corruption payloads reuse the benchmark's own ground-truth corrupted outputs where available, so corrupted results are well-formed and realistic. Even pair seeds are guaranteed to contain one context-loss event, so the artifact-relevant stratum retains statistical power while the fault-type profile remains estimable.

Why global call indices fail. Our first implementation keyed tool-level faults to the k-th post-fork call. This is intervention-*variant*: a branch that recovers efficiently makes fewer calls, so its task-critical calls occupy the indices at which the tape fires, while a branch that wastefully re-executes absorbs the same faults on redundant re-reads. Changing the keying rule changed measured checkpoint credit from $+13.5$ pp (semantically keyed) to -12.5 pp (index keyed), while placebo and no-op arms agreed exactly in both cases. These rules specify different fault-exposure interventions when branches make different calls. The comparison therefore shows sensitivity to the fault policy, rather than proving bias of an otherwise valid estimator for a fixed policy. Semantic anchors make the intended target of a tool fault explicit, but they do not force both branches to encounter it.

Alignment audit. For every pair we log the realized fault sequence (type, anchor, position) of both branches and report the fraction of pairs whose sequences differ. Under semantic anchoring this is $19/288$ for the open-weights model (all cases where one branch never calls the target tool); such pairs are retained and reported, never silently dropped. They violate event-presence invariance, so the full-sample contrast must not be described as an identical-realized-exposure comparison. It targets the declared tape policy, including branch-dependent exposure; its unbiasedness remains conditional on the marginal-law and weighting assumptions of Proposition 1.

Matched workspace and restart-only controls. The matched Qwen3-8B comparison uses tasks 001–024 except task 019, which did not reach the fork. Workspace-off tasks 001–012 come from the original run and 013–024 from the addendum. The arms share tasks, pair-seed labels, fault-tape rules, fork rule, judge and serving configuration; pre-fork prefixes and continuations are sampled independently. The clean condition is identical in both arms (no restart and no workspace reads), so we pool its per-task checkpoint–placebo contrast across arms. On–off intervals bootstrap the task-level difference directly with 2000 resamples. Checkpoint, placebo and no-op successes are 9/46, 1/46 and 1/46 with workspace access and 0/46 in every branch without it. This off-arm failure floor limits attribution of a precise informational share, while the matched contrast supports dependence on workspace availability.

Table 5: Reference-policy checkpoint gains under restart alone. Effects are percentage points with task-clustered 95% CIs. The last column is the paired change from the original restart-plus-tool-fault condition to restart only. Clean checkpoint–placebo contrasts are zero.

Family	Original effect	Restart-only effect	Paired change
C2	$+56.2$ [$+44.8, +67.7$]	$+56.2$ [$+41.7, +70.8$]	0.0 [$-9.4, +9.4$]
C3	$+45.8$ [$+29.2, +64.6$]	$+37.5$ [$+20.8, +58.3$]	-8.3 [$-20.8, +2.1$]

Table 5 separately evaluates the reference policy with tool faults removed. Both C2 and C3 retain positive checkpoint gains under restart alone. Paired changes compare the original condition and restart-only condition on matched tasks and seeds; they are not an additive decomposition of interacting fault mechanisms.

B JUDGE SPECIFICATION AND AUDIT

Blind second annotation. A stratified sample of 100 traced episodes (10 per source arm $\times$ judge verdict, spanning both models' autonomous and imposed-read arms and the SFT hold-out) was re-annotated by an independent annotator not involved in the project, with judge verdicts hidden and order shuffled, using the written three-part rubric. Agreement with the strict judge: 91% raw, Cohen's $\kappa = 0.82$, bootstrap 95% CI $[0.70, 0.92]$ ("almost perfect" on the Landis–Koch scale). In the wipe stratum ($\rho = 2$), where all headline contrasts are computed, agreement is $43/44$ (98%). The nine disagreements are 7 judge-strict (human success, judge failure) vs. 2 judge-lenient, concentrated in clean-condition answer-faithfulness edge cases, and balanced across branches (checkpoint 2, memory 3, placebo 2, no-op 2; credited 4 vs. control 3) — providing no clear evidence of branch-differential error in this sample. This does not establish that the contrasts are unbiased: even equal false-positive and false-negative rates across branches can attenuate success-rate differences.

Specification (v2.2). An episode counts as a success iff (i) some valid path's tool multiset has been executed with *clean* successes — fault-injected results, even well-formed ones, do not count, so a silently-corrupted call must be re-executed cleanly to register; (ii) no irreversible (Action-category) tool has been executed more times than the path requires, making duplicated side effects a failure; and (iii) the final message carries the information the task asked for: at least half of the *assessable* world-information steps must be reported. A step is assessable if it retains a high-reportability fact after filtering out echoes of the task statement (facts guessable from the prompt prove no retention) and status-like strings; a step counts as retained if *any* of its salient facts appears in the answer, with numeric and datetime format variants matched.

Calibration and audit. Thresholds were calibrated exclusively on clean control (no-op branch) episodes, before any branch comparison was examined. We audited 96 traced episodes: the tool-side verdict was consistent with an independent recount in $96/96$; on the answer side, an earlier version produced systematic false negatives on steps whose surviving facts were details competent answers omit (secondary prices, airport names), which motivated the assessability rule; residual failures ($\approx 18\%$ of clean controls) are dominated by genuinely vague answers ("All tasks completed successfully."). Audited false negatives have similar counts across severities (13 vs. 12 cases), and the same judge is used across branches. Equal error rates are not guaranteed by using the same judge; even symmetric errors can attenuate contrasts. Beyond the automated audit, a stratified review sheet (32 cases balanced over severity $\times$ verdict, with the final answer and the graded facts displayed side by side) plus additional sampled episodes — several dozen verdicts in total — were re-examined manually by an author; no verdict was judged incorrect. Every stored episode in the paper was re-judged offline with the same final implementation, so no dataset mixes judge versions.

C PROTOCOL, PRE-REGISTRATION AND AMENDMENTS

The protocol was frozen before any pre-registered-model episode and amended four times; each amendment predates the data it governs and is logged with its trigger. **v1.0**: initial design (early fork,

recent-first truncation, per-tool quotas). **v1.1**: duplicated-side-effect failures and a neutral truncation notice, after diagnostics showed the uncontrolled $+18.2\,\mathrm{pp}$ was placebo-shared. **v2.0–2.2**: the information-carrying answer requirement; oldest-first truncation — the realistic direction, and the one under which a fork-time artifact protects anything at all; session-level data-source budgets, after per-tool quotas proved evadable through alternative providers; and the judge assessability rule. **v2.3**: semantic fault anchoring (Appendix A). The sharp configuration (late fork, session-restart severity) was registered before the arms it governs were run; the shallow configuration is retained and reported as a null with a stated structural cause: an artifact protecting a fifth of the required information cannot flip a strict completion verdict, so no behavioural effect is measurable there — for any agent, including the reference policy.

Primary endpoints (paired Δ at the deployment cell against the cost-matched no-op; clean-cell equivalence within $\pm 3\,\mathrm{pp}$; placebo within $\pm 3\,\mathrm{pp}$; mechanism-off decay $\geq 70\%$) and the statistical unit (the task; cluster bootstrap over tasks) were fixed in advance. Kill conditions — placebo matching the checkpoint, effects confined to a single hand-tuned fault type, cost mismatch explaining the effect — were declared in advance; the first of them fired twice during reconnaissance, correctly, on runs this paper reports as false positives rather than findings.

D SECOND LINEAGE (BFCL), LADDER TABLE, AND SCALING DETAILS

BFCL port. Tasks are BFCL v4 multi-turn-base entries whose post-fork remainder contains ≥ 3 calls including one irreversible call (39/200 eligible; 24 sampled, stratified by involved API classes). The scripted prefix executes the ground-truth path to 0.6; the branch action is imposed by $do(a)$; the frozen tape wipes at the first continuation round (the crash lands on the artifact write, so the artifact carries the decisive share — BFCL remainders are irreversible-call-dense, and any unprotected window forces a duplicated side effect in every branch) and anchors one tool fault to a hash-chosen ground-truth tool. Wipe semantics: user turns survive (a user can re-send requests); assistant actions and results do not. Success = exact state match against the ground-truth oracle AND no irreversible call executed more often than the ground truth requires; judge agreement was verified manually on the smoke block before any pre-registered arm. The mechanism-off ablation replays the autonomous arm with the workspace returning empty; GPT-4.1-mini's autonomous contrast is $+6.2\,\mathrm{pp}$ at 0.8% reads, while the measured mechanism-off contrast is $+0.0\,\mathrm{pp}$ $[-6.5, +6.5]$. These summaries do not identify the informational share: the read rate is episode-level, and the mechanism-off interval is broad. A matched on/off contrast is needed for that attribution.

ALFWorld port (third lineage). This port uses ALFRED TextWorld games with deterministic PDDL, with expert paths cached by one dry rollout per game (121 solvable; the arm pool is the 56 games with 8–16 expert steps). The harm channel is a step budget of $\mathrm{len}(\mathrm{path}) + 2$, and the wipe lands at the first continuation round. The reference policy is therefore *index-based* over the precomputed path: querying the state-based expert at recovery time would leak knowledge an amnesiac agent should not have. Without the artifact, redo-from-zero costs $0.6L$ extra steps and exceeds the budget on every game, so the reference-policy result ($+100\,\mathrm{pp}$, placebo 0, clean 1.00 over 56 games) is structurally informative everywhere. For LLM agents, however, cheap re-observation (`look`/`goto` reveal state) substitutes for the artifact. Under the pre-registered no-op-only selection (clean passes and $\rho = 2$ no-op fails), the informative stratum falls to 5 of 56 games at slack 2 (13 clean-passers at slack 4, 3 informative). On that stratum, GPT-4.1-mini reads $0/80$ autonomously (credit $-10\,\mathrm{pp}$ $[-35, +10]$) and converts an imposed read into $+35\,\mathrm{pp}$ $[0, +75]$, reported as supplementary at five task clusters. The full selection trail (56→13→5) and all pilot episodes are released.

Scaling grid (Exp S). Sizes are the generator's exact-dedup capacities (689, 949) plus the original 375 recipe (which contained near-duplicate demonstrations, an implicit recovery upweighting; recorded before results were seen). Per-seed held-out credits (wipe stratum): 375: $+16.2/+5.0/+2.1$; 689: $+6.5/+5.3/+0.0$; 949: collapse$/+0.0/+0.0$. Median read rates .33/.35/.16. The exploratory "949u" re-weighting (recovery repeated to 31% of the mix) restores reading ($.20 \rightarrow .45$ on seed 42) but not conversion, and collapses clean success on seed 17. The 949/seed-17 run fails every task prefix at a converged loss of 0.003 — behavioural collapse invisible to the loss.

Table 6: WORKSPACEPROBE discovery ladder; paired contrast against placebo (exactly 0.00 at every level for both models).

	Where the affordance is surfaced	GPT-4.1-mini (24 tasks) P(read)	GPT-4.1-mini (24 tasks) Credit	GPT-4.1 (20 tasks) P(read)	GPT-4.1 (20 tasks) Credit
L0	tool list only	0.00	+0 pp	**0.00**	+0 pp
L1	+ description: "survives restarts"	0.00	+0 pp	0.03	+3 pp
L2	+ task statement mentions it	0.12	+10 pp	—	—
L3	+ restart notice mentions it	0.67	+63 pp	**0.97**	+90 pp
L4	read imposed by the harness	1.00	+94 pp	1.00	+98 pp

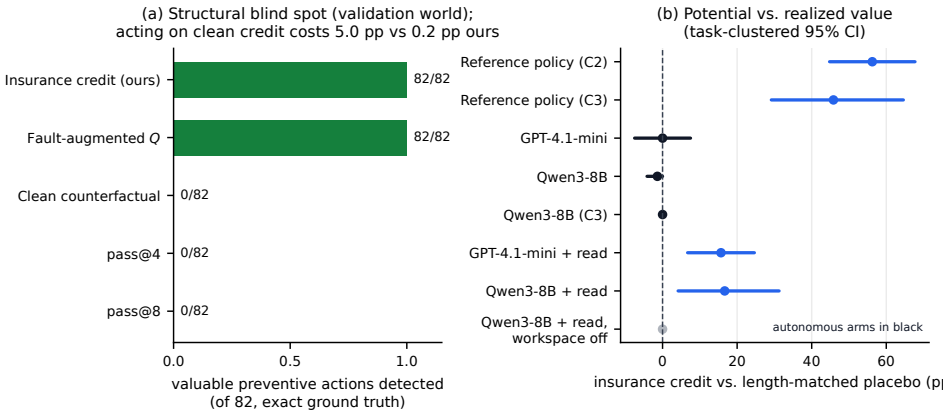

Figure 3: **(a)** With exact ground truth, fault-blind signals assign zero credit to actions genuinely worth +12.2 pp. **(b)** The value is large for a policy that uses the artifact (top two rows), small for the autonomous conditions shown (black; additional models appear in Table 2); an imposed read recovers part of the available value. The workspace-off result displayed here uses a smaller task subset; the separate matched on/off evaluation is reported in Appendix A. Task-clustered bootstrap 95% CIs against placebo.

Table 7: Which credit signal to act on. Decision regret = realized success forfeited by acting on the signal (BASINDAG; misspecification = fault priors shifted after measurement).

Deployment fault distribution	Right signal	Decision regret
Known, stationary	fault-augmented Q (scalar)	0.01 pp
Known, stationary	insurance-credit profile	0.2 pp
Misspecified	profile, adaptively re-weighted	**0.31** pp
Misspecified	fixed-prior scalar	1.23 pp
Misspecified	profile, minimax	4.52 pp

E DECISION-UTILITY EVALUATION

Table 8: Deployment success from acting on each signal (two-fold cross-validation over held-out seeds; 96 folds over 48 tasks; §8).

Selection rule	Deployment success
Oracle best	0.615
Insurance credit, random ties	0.490 [0.362, 0.620]
Clean counterfactual credit, random ties	0.193 [0.148, 0.240]
Insurance credit, first-in-order ties	0.448 [0.312, 0.583]
Clean credit, first-in-order ties	0.052 [0.021, 0.094]
Random	0.193
Do nothing	0.052

F THE SPURIOUS +18.2 pp EFFECT

The first comparison this project ran used the unmodified benchmark, no placebo arm, and independent sampling. Checkpointing appeared to improve fault-condition success by +18.2 pp. Three observations unmade it: the length-matched placebo — persistent junk of equal size — improved success by the same +18.2 pp; no episode in any arm ever read the workspace, so no mechanism could have carried the effect; and the gains were spread across fault types rather than concentrated where an artifact could help. The effect was context flavour plus unpaired sampling noise; re-running the same arms with shared tapes and the truncation notice collapsed every contrast to zero. We were one missing control arm away from publishing a robustness gain that does not exist — the concrete argument for treating the placebo arm and the non-fault stratum as minimum equipment for action-level robustness claims.

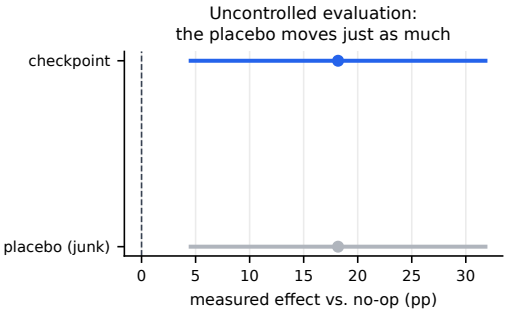

Figure 4: The uncontrolled comparison. Checkpointing appears to help by +18.2 pp; the length-matched placebo moves by the same amount, and no episode in either arm accessed the workspace.