ASSET OR EXPENSE? KEEPING THE BOOKS OF SELF-EVOLVING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving agents persist experience as artifacts such as playbooks, memories, or skill libraries, and are evaluated by the accuracy gained while the task stream runs. We ask what the deployer owns when evolution stops. We introduce a *ledger* that tags every LLM call by phase, content-addresses the artifact before measuring it, and fills four accounts: frozen held-out value, build cost, per-task carry cost, and the lifetime of the backbone the artifact depends on. A recognition gate, fixed before the closed-model runs, refuses to compute depreciation for an artifact whose value is not shown to be positive. Instantiated inside the AgentStream suite with its methods, tasks, and prompts unchanged (54 runs, 8 backbones), no artifact passed the gate; one was significantly harmful after multiple-comparison correction (GPT-5.5's own ACE playbook, -24 points on Tau2-telecom) and two more before correction. On the column with the largest reported online gain (GPT-5.4, Tau2-telecom), both the online gain we reproduce ($+0.04$) and the frozen value ($+0.02$, 95% CI $[-0.06, +0.11]$) are indistinguishable from zero, and on that model a length- and vocabulary-matched, syntax-destroyed copy of the playbook performs comparably. The costs, by contrast, are certain: even with 96% of prompt tokens served from cache, carrying the playbook raises per-task execution cost by 48% and overtakes the one-off build cost after about 67 tasks; online rewriting adds about one cache miss per task. Against the alternative of upgrading to the successor model, the artifact would need at least 7.7 points of value to break even. By these books every artifact we measured is an expense, and we state the four measurable conditions under which one would be an asset. We release the ledger, the gate, and a dataset of dated snapshot lifetimes (12–21 months for 2025 releases).

1 INTRODUCTION

A self-evolving agent does more than solve the task in front of it. After each task it reflects, and it writes the result of that reflection somewhere persistent: a playbook of strategies (ACE (Zhang et al., 2026)), a memory graph (A-Mem (Xu et al., 2025); see also Park et al., 2023; Packer et al., 2024; Wang et al., 2024c), a bank of distilled reasoning (ReasoningBank (Ouyang et al., 2026)), a skill library (AutoSkill (Yang et al., 2026)), or a revised harness (Zheng et al., 2026). The literature evaluates these methods by streaming tasks past the agent and reporting the accuracy gained along the way (Figure 1a; Wei et al., 2026; Wang et al., 2026; Ren et al., 2026). The most systematic such evaluation is AgentStream (Yan et al., 2026), a suite of six agentic benchmarks, five self-evolving methods, and three streaming scenarios. It finds the gain small on average ($+1.4$ points over 45 configurations), concentrated in a few benchmark columns, gated by model capability, and non-monotone in model strength.

That evaluation answers one question, whether accuracy rose while the agent evolved, and leaves a second one untouched: when the stream ends, what does the deployer own? The persistent artifact is either an asset, something built once that keeps paying, or it is part of the running expense of a particular way of prompting a particular model. A stream-level accuracy number cannot tell the two apart; a bill can. An asset can be frozen, audited, transplanted to the next model, and amortized. An expense cannot, and an artifact mistaken for an asset is a form of technical debt (Sculley et al., 2015). No evaluation we know of draws the distinction. Token logs (Zheng et al., 2026) and cost-controlled comparisons (Kapoor et al., 2024) exist, but none freezes the artifact, attributes cost to the phase that incurred it, or gives the artifact a horizon.

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.

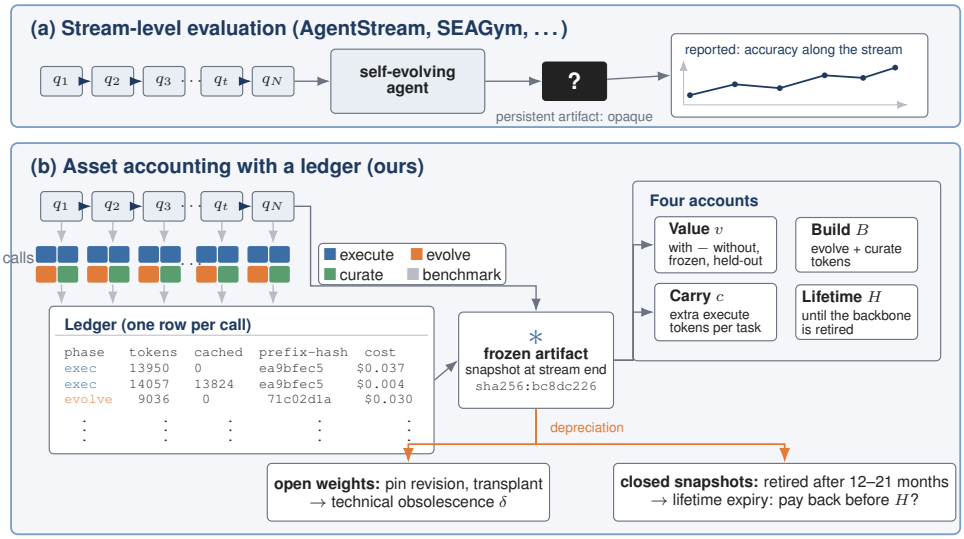

Figure 1: Stream-level evaluation vs. asset accounting. **(a)** Existing evaluations stream tasks past a self-evolving agent and report accuracy along the way; the persistent artifact is never separated from the act of evolving. **(b)** The ledger tags every call with its phase, freezes and content-addresses the artifact at the end of the stream, and fills four accounts: value (frozen, held-out, paired), build, carry, and lifetime. Depreciation is then measured along the two axes on which models are actually deployed: transplant to a pinned open-weight backbone, or retirement of a closed snapshot.

This paper treats the artifact as an asset and keeps its books (Figure 1b). Four accounts describe it: the *value* it adds when frozen and injected on held-out tasks, the *build* cost of the calls that produced it, the *carry* cost it adds to every later call, and the *lifetime* of the backbone it was built on. Depreciation, the loss of value over time, then has two sources, and they correspond to the two ways models are deployed. Open weights can be pinned to a revision, so the loss that matters is *technical obsolescence*: how much of the value survives a change of backbone. Provider-hosted models cannot be pinned, and vendors have retired their recent dated snapshots after 12 to 21 months, so the loss that matters is *expiry*: whether build and carry are recovered before the forced migration. Both are governed by one horizon, the earlier of a forced replacement, when the vendor retires the snapshot, and a voluntary one, when a better model appears; open weights only remove the forced term. Pieces of this picture exist (asymmetric transplant of evolved artifacts, Zheng et al., 2026; Ye et al., 2026; Gao et al., 2026; cost-controlled evaluation, Erol et al., 2026; the price of breaking a prompt cache, Lumer et al., 2026); the bookkeeping that joins them does not, and it begins with a question depreciation presupposes: whether the frozen artifact is worth anything at all.

We instantiate the ledger inside AgentStream without changing its methods, benchmarks, or prompts, run it across six open-weight and two closed-model rounds, and read the books. The contributions rest on evidence of unequal strength and are listed in that order. (1) *Method*: the ledger, content-addressed freezing, and a recognition gate that refuses to compute depreciation for an artifact whose value is not shown to be positive, together a protocol that any evaluation of persistent artifacts can reuse (Section 2). (2) *Measurement*: across eight rounds no frozen artifact showed significant positive value, one was significantly harmful after multiple-comparison correction and two more before it, while the cost accounts are exact (Sections 4 and 5). (3) *Economics*: a pay-back rule with a break-even value, a dataset of vendor snapshot lifetimes, and the condition under which the lifetime binds (Section 5). (4) *Exploratory observations*, labeled as such: the artifact changes how long the agent talks, and a placebo with its syntax destroyed reproduces most of its footprint. Section 6 reads the results through accounting, economics, organizational learning, and meta-science; Section 8 states each threat to validity with its bias direction.

2 Accounting for a Persistent Artifact

To tell an asset from an expense one needs what an accountant needs: an object with an identity, costs attributed to the activities that incurred them, a value measured on its own, and a horizon

over which to recover it. We borrow the vocabulary deliberately. Accounting standards recognize an internally generated intangible only when future benefit is probable and its cost is measurable, expense everything spent before that point, amortize the recognized asset over its useful life, and test it for impairment when circumstances change (International Accounting Standards Board, 2004b;a).

Terms. A *column* is one benchmark×model cell of AgentStream; a *round* is one build plus frozen measurement on a *source* backbone and, where run, a *target* backbone inheriting the artifact; *carry* is the recurring execute-phase cost the artifact adds per task; *churn* is a cache miss caused by rewriting it; *label-free* means the evolve step sees only the agent's own trajectory and self-generated feedback, never the ground-truth outcome (ACE's "helpful/harmful" counters come from its own reflector).

Four accounts. An asset A built by a self-evolving method on backbone m is described by *value* v (accuracy gain on held-out tasks when A is injected), *build cost* B (tokens spent in evolve/curate calls while producing A), *carry cost* c (extra execute-phase tokens per task when A is present), and *lifetime* H, the amortization horizon: $H = \min(H_{\text{forced}}, H_{\text{voluntary}})$, the earlier of the vendor's retirement of m and the deployer's own switch to a better model. Open weights can be pinned to a revision, so $H_{\text{forced}} = \infty$; the voluntary term applies to both.

Phase-tagged calls. Every LLM call passes through one wrapper that requires a phase tag as a positional argument (`execute`, `evolve`, `curate`, `benchmark`, `overhead`); a missing tag fails in the caller's stack. A safety-net callback writes a `phase=unknown` poison record for any call that bypasses the wrapper, and a run containing a poison record is invalid. Each record stores server-reported usage (prompt, completion, cached-read, cache-write, reasoning tokens), a hash of the stable request prefix (system messages plus tool schemas), the provider's cache-hit flag, and the dollar cost from the public price sheet. The benchmark apparatus (tau2's user simulator) is recorded under `phase=benchmark` and excluded from asset attribution.

Freezing. Self-evolution normally interleaves solving and learning, so a with/without contrast on the same stream cannot separate "owning accumulated knowledge" from "the act of reflecting after each task". We therefore build A on a fixed set of tasks, take a SHA-256 content-addressed snapshot of every store, and evaluate on a *disjoint* held-out set with evolution hard-disabled: any evolve-phase call under the frozen flag raises. The build set is AgentStream's own 50-task selection (seed 42); the held-out set is the rest of the pool in the same selection order (64 tasks for Tau2-telecom, of which the first 50 were run first and the last 14 added afterwards; Section 5).

The recognition gate. Let s_i^+, s_i^- be task i's score with and without the frozen asset. Value on the source backbone is $r_{\text{src}} = \frac{1}{n}\sum_i (s_i^+ - s_i^-)$ with a task-clustered bootstrap interval (Miller, 2024). Transplanting the same snapshot to backbone m' gives r_{tgt}, and depreciation is $\delta = 1 - r_{\text{tgt}}/r_{\text{src}}$. δ is an impairment test: it is only defined when the denominator is positive, so the protocol fixes a *recognition gate*: *if the lower confidence bound of* r_{src} *is* ≤ 0*, the round is reported as REFUSED and* δ *is not computed*. The rule is asymmetric on purpose, in the spirit of accounting conservatism: an artifact not shown to be worth anything is carried as an expense, not as an impaired asset. It was fixed in the runner before any closed-model run (after the open rounds had been seen) and was not registered on a public platform; the one design change after that point, the 14-task extension of the GPT-5.4 round (C1, Section 5), is disclosed where it occurs. When δ is defined the runner also buckets it (retained, halved, evaporated, toxic); no round in this paper reached that step, so we report r_{src} and, where measured, r_{tgt} descriptively, and we report the *upper* confidence bound of r_{src} as the informative number. Because eight rounds are tested, significance is also reported after Holm correction.

What the gate can decide. Scores are binary per task, so with $n = 50$ the 95% half-width of the paired difference is about 12 points (Figure 6, appendix): the benchmark-wide +1–2 point gain is undetectable at any affordable n, a +13-point gain has power 0.55 (0.62 at $n = 64$) (Card et al., 2020). The gate can recognize an asset only where the reported gain is large; a refusal elsewhere bounds value from above.

Churn. Provider prompt caches match on the rendered prefix. An asset injected into the system prompt is part of that prefix, so every rewrite of the asset invalidates the cache for the next task:

one change of the prefix hash between consecutive calls is one churn event. Under online evolution the first call of every task misses; under a frozen asset it hits from the second task on. Freezing is therefore also the cheaper regime, and the ledger prices the difference from cached-read counts. When the artifact sits in the system prompt and the conversation history dominates the prefix, the amount is bounded at about one uncached prefix per task, so it matters most for short tasks and under a cache-write premium (Section 5).

3 Experimental Setup

AgentStream (Yan et al., 2026) is an evaluation suite: six agentic benchmarks (AppWorld, BFCL, BrowseComp-Plus, HLE, SWE-bench-Verified, Tau2), five self-evolving methods (ACE, A-Mem, ReasoningBank, AutoSkill, Harness), and three streaming scenarios, run from one harness. We instantiate the ledger inside that harness without modifying its methods, benchmarks, task selection, or prompts; only the base models change. The ledger instruments all five methods and every benchmark adapter; the runs reported here cover three benchmark columns (GSM8K, BFCL, Tau2-telecom), two methods, eight backbones, and 54 runs, amounting to 2,212 agent sessions, 51K LLM calls, and 482M tokens (Table 2). Two facts from AgentStream shape the design (Table 4): its mean evolution gain over 45 configurations is $+1.37$ points, and the large gains concentrate in Tau2-telecom (the telecom domain of τ^2-bench; Barres et al., 2025; Yao et al., 2024), a dual-control customer-service environment with an LLM-simulated customer (GPT-5.4 with ACE: 50.0 $\rightarrow$ 63.3); the open rounds use BFCL (Patil et al., 2025) and, as a sanity column outside the suite, GSM8K (Cobbe et al., 2021). Decisions made in advance: the gate, the build/held-out split, the closed-model snapshots, and the placebo design; decisions made after seeing data: the 14-task extension of C1 and the choice of Tau2-telecom after the open rounds (Nosek et al., 2018). Open models are pinned to a revision SHA, closed models to dated snapshots; client-side response caching is disabled everywhere; sampling is explicit or recorded as provider-fixed.

4 Open Weights: What the Frozen Artifact Is

On open weights the backbone can be pinned, so the forced term of the horizon vanishes and value and carry are read in tokens. Six rounds fill the books (Table 3; Figure 6 in the appendix shows the intervals and the gate's power). The value account is read first, then what the carry account says the artifact does instead.

Observation 1. The value account. Two open rounds are significantly negative before multiple-comparison correction, R1 on GSM8K (-0.12 [-0.22, -0.02]) and R5 on gpt-oss-120b (-0.16 [-0.30, -0.02]); neither survives Holm correction over the eight rounds ($p_{\mathrm{Holm}} = 0.22$), and the other four have wide intervals around zero. On R5 the same snapshot transplanted to Qwen3-32B-AWQ gives $r_{\mathrm{tgt}} = -0.14$ [-0.28, 0.00]. All six sit on columns where AgentStream reports no gain distinguishable from seed variance (BFCL: Claude Opus 4.7 86.0 vs. 85.3 with ACE, Gemini 3.1 Pro $+4.7 \pm 13.3$; GSM8K is outside the suite), and with $n = 50$ the gate detects only effects above about 12 points, so these refusals bound value from above; R5 is the one result inside the suite that excludes a positive effect.

Observation 2. What the carry account shows (exploratory). Figure 2 plots, for every round and both sides of every transplant, execute calls per task with and without the same frozen asset. On the backbone that built it the asset usually sits above the diagonal (Qwen3-4B 7.2 $\rightarrow$ 16.9 calls per task on GSM8K, gpt-oss-120b 25.5 $\rightarrow$ 40.1, GPT-5.4 16.7 $\rightarrow$ 20.2); transplanted, the same snapshot lands below it (granite-3.3-8B 45 $\rightarrow$ 16 with the R1 playbook; Qwen3-32B-AWQ 18.7 $\rightarrow$ 7.2 with R5's). "Role" (source vs. target) and "baseline behavior" are confounded here: the target for R1–R4 is granite-3.3-8B, which loops at 27–45 calls per task without any asset. R3-A separates them, since there the *source* Qwen3-8B-AWQ loops without the asset (37.5 calls) and the playbook cuts it to 11.1; the consistent description is that the asset acts on pacing, shortening runs of a backbone that loops and lengthening runs of one that does not; the non-looping GPT-5.5 target is lengthened like its source (17.4 $\rightarrow$ 19.1). A placebo bounds how much of this is content: re-running the GPT-5.4 arm with the same playbook word-shuffled inside every entry (length, vocabulary, section headers, and entry scaffolds preserved; syntax destroyed; one replicate; diamond in Figure 2) gives 17.9 calls per

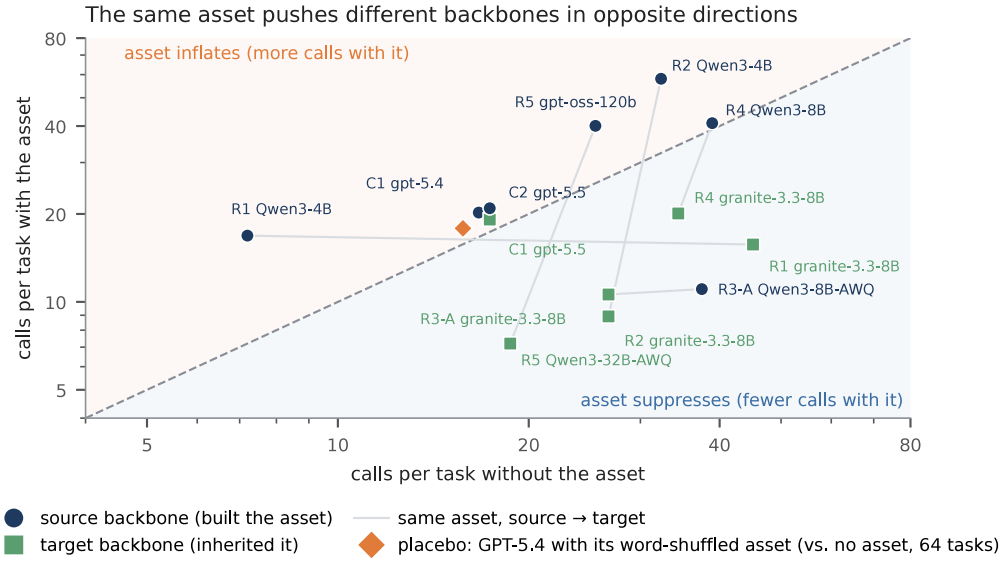

Figure 2: Execute calls per task with vs. without the same frozen asset, one point per round and side (means over 50 tasks; the placebo point and its no-asset reference are over 64 tasks; log axes). Circles: the backbone that built the asset; squares: a backbone that inherited it; grey lines join the two ends of one asset. Orange diamond: GPT-5.4 with a word-shuffled copy of its own playbook (length and vocabulary preserved, syntax destroyed). Source inflation and target suppression are confounded with baseline behavior (the targets loop without any asset); R3-A, a looping source that the asset suppresses, is the informative exception.

task against 15.8 with no asset and 19.4 with the real playbook (64-task means; 50-task means above), at the same prompt tokens per call (15.4K). Most of the footprint follows from the text's presence; the content-specific remainder ($+1.5$ calls per task) is inside the between-task spread. Keyword effects ("single-step" survives the shuffle) are not excluded, and the transplant side was not placebo-tested.

Observation 3. Carry decomposed. Figure 5 (Appendix G) decomposes the extra prompt tokens per task. On R5 the 5.7K-token playbook costs 318K extra prompt tokens per task, of which 46% is the asset text re-sent on every call and 51% is the 15 additional calls it induces, each carrying the full history. Fifteen R5 tasks hit the 40K context limit against 2 without the asset, so part of R5's negative value may be truncation rather than content; we cannot separate the two. On the closed snapshots the proportions invert (70% text on GPT-5.4) because the strong model adds few calls, and on R3-A the asset shortens the run so the net carry is negative.

5 Closed Models: The Cost of Ownership

On provider-hosted models every account is in dollars and the lifetime is finite. We ran the full protocol on `gpt-5.4-2026-03-05` (C1: reasoning effort medium, provider-default sampling, user simulator the same snapshot, as in AgentStream) and repeated the source side on its successor `gpt-5.5-2026-04-23` (C2), onto which we also transplanted C1's frozen playbook, the migration a deployer faces when GPT-5.4 retires. Table 1 collects the measured accounts; Findings 1–6 read them in turn. C1's 14-task extension and the placebo run carry the run labels C6 and C7 in the appendix.

Finding 1. Frozen value. Stream-level evaluations report the gain accumulated while evolving; a deployer owns what is left when evolving stops. We measured both on one model. Scoring GPT-5.4 with no asset on the 50 build tasks gives the *online* gain of ACE evolving after every task: 0.78 vs. 0.74, $+0.04$ $[-0.08, +0.16]$. Freezing the playbook that same stream produced and scoring the 64 held-out tasks gives its *frozen* value: 0.84 vs. 0.82, $r_{\text{src}} = +0.02$ $[-0.06, +0.11]$ over two independent replicates whose own contrasts have opposite signs (-0.02 and $+0.06$); the first 50 tasks gave $+0.05$ $[-0.04, +0.14]$ and the 14 added tasks moved it toward zero. Two identical runs agree

Table 1: Measured accounts on Tau2-telecom (ACE; public tariff $2.50 / $0.25 cached / $15 per million tokens for GPT-5.4, $5 / $0.50 / $30 for GPT-5.5; every ledger audit valid with zero poison records).

Quantity	Value	Note
Held-out success, with / without frozen asset (GPT-5.4, 64 tasks, 2 seeds)	0.84 / 0.82	$r_{\mathrm{src}} = +0.02$ $[-0.06, +0.11]$; seeds -0.02, $+0.06$; first 50 tasks $+0.05$ $[-0.04, +0.14]$
Build-task success, online ACE / no asset (GPT-5.4)	0.78 / 0.74	online gain $+0.04$ $[-0.08, +0.16]$
Task outcomes agreeing between identical runs, without / with	47 / 48 of 64	sampling noise floor
GPT-5.4 asset on `gpt-5.5`, with / without (49 paired tasks)	0.918 / 0.918	$r_{\mathrm{tgt}} = 0.00$ $[-0.08, +0.08]$; 0.920 over all 50 with-asset tasks; carry $0.103 per task
GPT-5.5's own asset, with / without (49 tasks)	0.673 / 0.918	$r_{\mathrm{src}} = -0.24$ $[-0.39, -0.12]$
Placebo: word-shuffled GPT-5.4 asset, with / none (64 tasks)	0.875 / 0.820	$+0.055$ $[-0.03, +0.14]$; real $-$ placebo -0.03 $[-0.11, +0.06]$; 17.9 calls, carry $0.047
Prompt tokens per execute call, with / without (GPT-5.4, first 50 tasks)	15,342 / 7,814	$\times 1.96$; calls per task 20.2 / 16.7
Cache-served share of prompt tokens, build / without / with	89% / 89% / 96%	frozen asset caches best
Carry c per task, GPT-5.4 (seed 1; seed 2; extension tasks)	$0.056; $0.056; $0.048, $0.064	execute cost $0.172 vs. $0.116
Build cost B (evolve + curate), GPT-5.4 / GPT-5.5	$3.77 / $8.58	$0.075 / $0.17 per task
Churn (online $-$ frozen cache cost) per task, GPT-5.4 / GPT-5.5	$0.0086 / $0.0096	one uncached prefix per task

on only 47–48 of 64 tasks, so 16–17 outcomes flip on sampling alone. In our reproduction both are indistinguishable from zero; the one-sided 95% upper bound on the frozen value is $+0.09$ (two-sided $+0.11$), and the bootstrap mass above the break-even value of 0.077 (Finding 6) is 10%. We did not reproduce the $+13$-point online gain AgentStream reports on this column: our no-asset baseline solves 74% of the build tasks where AgentStream reports 50%, a discrepancy we cannot attribute, which shifts both arms of every paired contrast equally but leaves less headroom (Section 6 returns to the gap). On the same 64 tasks the syntax-destroyed placebo scores 0.875 (real $-$ placebo $= -0.03$ $[-0.11, +0.06]$; one replicate against a two-seed mean): no evidence that the readable content adds value beyond text of the same length and vocabulary.

Finding 2. Carry as a running expense. With 89% (no asset) to 96% (frozen asset) of prompt tokens served from the provider cache, carrying the 7.6K-token playbook raises execute cost per task by 48% ($0.172 vs. $0.116; $0.162 vs. $0.106 in the second replicate; $0.048 and $0.064 on the 14 extension tasks), i.e. $c = \$0.056$, and doubles the prompt tokens per call ($\times 1.96$). Against a one-off build of $3.77 ($0.075 per build task), the cumulative carry overtakes the build after $B/c = 67$ tasks and keeps growing (Figure 4, Appendix G). Thirty percent of the carry is induced calls (20.2 vs. 16.7 per task); the rest is the asset text itself re-sent on every call. The placebo reproduces $0.047 of the $0.056 (84%): carry is priced largely by length. A token-count projection made before any closed run (Table 6) had put carry at $0.21–$0.93 per task, 4–16$\times$ too high because it could not see caching; the accounts are therefore read from server-reported cached tokens.

Finding 3. Churn. Figure 7 (Appendix G) resolves the churn account by call index. Under online evolution the first call of every task is served 2% from cache, because the rewritten playbook is new to it, and the second call is already at 90%; under a frozen asset the first call is at 97% because the asset is unchanged since the previous task. Everything after the first call is cached in both regimes because the conversation history, not the asset, dominates the prefix. Churn is therefore about one uncached prefix per task in this configuration (asset in the system prompt, long multi-turn tasks): $0.0086 per task on GPT-5.4 and $0.0096 on GPT-5.5, 15% of the carry and about 5% of the execute bill. It would scale up for short tasks and under a cache-write premium (Anthropic charges 1.25–2$\times$ on writes).

Finding 4. The stronger backbone. GPT-5.5 is a stronger baseline (0.918 without any asset on the held-out tasks), and its own frozen playbook (3.7K tokens, half the size of GPT-5.4's) harms it: 0.673 with vs. 0.918 without, $r_{\mathrm{src}} = -0.24$ $[-0.39, -0.12]$ ($n = 49$; one session stalled without error and is excluded), the only contrast that survives Holm correction ($p_{\mathrm{Holm}} < 0.001$). Figure 8 (Appendix G) shows where the 24 points go. Thirteen tasks flip from solved to failed and one the other way; all thirteen are MMS-troubleshooting tasks, against a base rate of 51% MMS tasks in the held-out set (binomial $p \approx 2 \times 10^{-4}$), eight with the "easy" customer persona, and in each the conversation with the asset runs longer (median 113 vs. 84 messages), ends with the customer leaving, and fails the environment check. The asset is shorter than GPT-5.4's yet more harmful, so length does

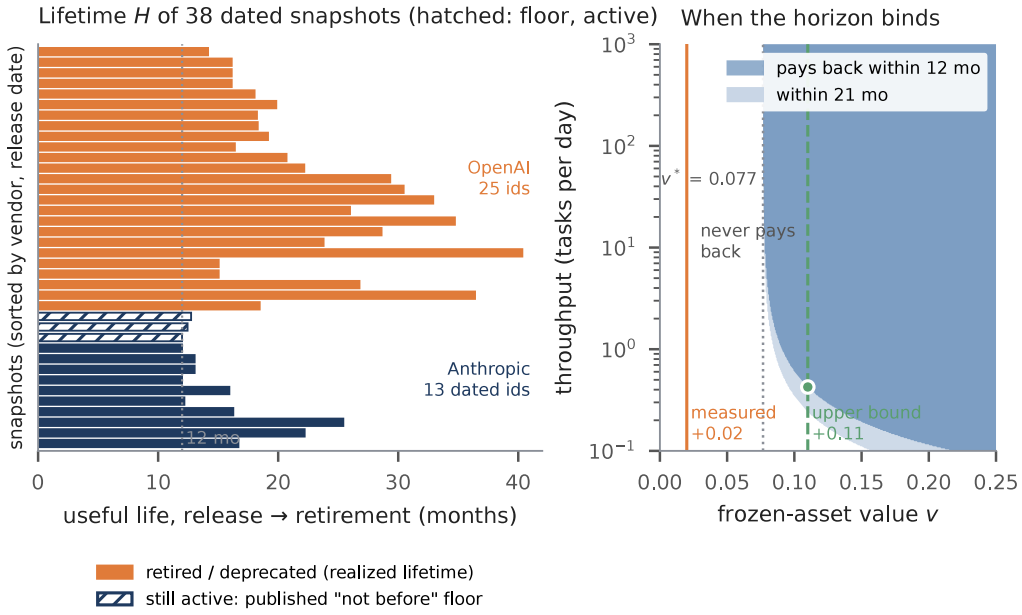

Figure 3: **Left:** useful life of the 38 snapshots in the lifetime dataset that have both a release date and a retirement date or published floor (8 further Anthropic ids have no dated release and are omitted): Anthropic's 2025 snapshots lived 12–13 months, OpenAI's 14–21, older OpenAI snapshots up to 40. **Right:** pay-back phase diagram: shaded where $N^* = B/(vp - c)$ is reached before the horizon H (12 months, the published floor; 21 months, the longest 2025 lifetime) for a value v and a throughput. Below $v^* = 0.077$ the asset never pays back; the measured $+0.02$ lies there, the upper bound $+0.11$ pays back within 12 months above about half a task per day.

not explain this loss; its content does something on this backbone. On the three largest backbones we ran, gpt-oss-120b (-0.16), GPT-5.4 ($+0.02$), and GPT-5.5 (-0.24), the frozen asset's value is never positive, but the three points differ in benchmark (BFCL vs. Tau2) and reasoning effort (medium vs. default) as well as in capability, and we do not read a trend into them.

Finding 5. The retirement event. Transplanting the frozen GPT-5.4 playbook onto `gpt-5.5-2026-04-23` is the migration a deployer faces when GPT-5.4 retires. On the 49 paired held-out tasks the successor scores 0.918 with the inherited asset and 0.918 without (0.920 over all 50 with-asset tasks), $r_{\text{tgt}} = 0.00$ $[-0.08, +0.08]$, while carrying it costs \$0.103 per task (\$0.291 vs. \$0.188; prompt tokens per call double at the higher tariff). By the gate no δ is computed, since the asset never had significant value to lose, but the accounts of the event are complete: no measurable value, positive carry, and a smaller lengthening of the conversation than on the source (17.4 → 19.1 calls per task).

Finding 6. Pay-back. The lifetime account (Figure 3, left; Table 5; transcribed from OpenAI, 2026b; Anthropic, 2026b) records 46 provider snapshots: realized lifetimes of 12.0–13.0 months for Anthropic's 2025 releases and 14.2–20.7 for OpenAI's (up to 40 months for 2023 releases), notice of 60–180 days, and published floors of release $+12$ months. Whether that horizon matters depends on the pay-back time $N^* = B/(v\,p - c)$, where p is the price of one additional solved task obtained instead by moving to a stronger model (an illustration on point estimates whose intervals include zero). Round C2 measures the alternative: upgrading from GPT-5.4 (replicate 1) to GPT-5.5 with no asset raises held-out success from 0.82 to 0.918 and execute cost from \$0.116 to \$0.188 per task, $p = \$0.73$ per additional solved task, so the asset breaks even at $v^* = c/p = 0.077$. The GPT-5.4 asset's $v = 0.023$ is worth $v\,p = \$0.017$ per task against $c = \$0.056$: the margin is $-\$0.039$ at the point estimate ($-\$0.041$ at $v = 0.02$) and $-\$0.056$ at $v = 0$. At the upper bound $v = 0.11$ the margin is $+\$0.024$ and $N^* = 155$ tasks, reached inside a 12-month horizon by any deployment above half a task per day (Figure 3, right). For this asset the sign of the margin binds; H binds only where the margin is thin and the throughput low.

6 Positioning Discussion

The books say expense, not asset. The readings in this section are interpretive hypotheses. An internally generated intangible is recognized as an asset only when future benefit is probable and its cost can be measured; expenditure before that point is expensed (International Accounting Standards Board, 2004b), and a recognized asset is tested for impairment when circumstances change (International Accounting Standards Board, 2004a). The gate is that recognition test and δ the impairment test; "refused" means "expense, not asset". Why does an artifact that costs money and looks harmless fail even the recognition test? Either the reported gain was overestimated or the artifact is the wrong kind of object.

We chose, by design, the column with the largest reported gain out of 45; effects selected for size from underpowered comparisons are inflated and regress on replication (Ioannidis, 2008; Button et al., 2013), so our failure to reproduce $+13$ points on a 74% baseline is what this predicts, and the same literature motivates the gate as a defense against building on a winner's-curse estimate. The shuffled playbook is an *active* placebo in the clinical sense: it matches the side effects while removing the putative active ingredient; preserving vocabulary, it bounds rather than isolates the content's contribution. Regression explains an absent gain, not a harmful one; for GPT-5.5's playbook the content itself must be examined.

Why harm: what a label-free store accumulates. Organizational learning says what such a store holds: routines learned from experience without reliable performance feedback are prone to superstitious learning, and organizations rarely retire routines, so procedures accumulate (Levitt & March, 1988; Nelson & Winter, 1982). In the frozen playbook 76 of 106 entries were never marked helpful by ACE's own curator and 33 were marked harmful, yet all are injected on every call; the never-helpful entries hold 74% of the playbook's tokens, so a retirement rule would have cut carry by about \$0.029 per task. Educational psychology says why the sign flips: guidance that helps novices can hurt experts by conflicting with strategies they already have (Kalyuga et al., 2003), which maps onto Observation 2 and onto Finding 4. We do not claim that models have schemas; the prediction is that a store's effect depends on the gap between its advice and the recipient's default strategy. Together they say the playbook stores advice on how *this* backbone paces a conversation, not domain facts; such content is bound to the model that wrote it, and economics has a name for that.

Firm-specific human capital pays only for as long as the employment relation lasts (Becker, 1964), and relation-specific investments expose their owner to hold-up by the counterparty (Williamson, 1985); a vendor retiring a snapshot is that hold-up. In these terms GPT-5.4's playbook retaining no measurable value on its successor while its carry doubles (Finding 5) is the expected fate of specific capital. Domain-fact entries should then survive transplant better than pacing entries. Specific capital on a relation of finite length turns the question into when it is worth building at all.

Capital theory distinguishes physical wear from economic obsolescence: a machine is retired when a newer vintage makes it uneconomic, not when it stops working (Solow, 1960), and when better vintages arrive at a known rate there is option value in deferring irreversible investment (Dixit & Pindyck, 1994). Finding 6 is this logic applied to prompt-resident artifacts: the horizon is the earlier of forced retirement and voluntary upgrade, so it applies to open weights too, and the phase diagram (Figure 3) locates the thin-margin, low-throughput region where the vendor's floor rather than the margin's sign decides; freezing is the option-preserving choice.

What would make it an asset? Each step above names a condition; together they are future work. The artifact must pass recognition: reported frozen on held-out tasks, paired and gated, next to the online gain. It must stop accumulating: a retirement policy for entries, which halves carry on our ledger. It must be general rather than specific: a representation independent of the backbone that wrote it. It must pay back inside the horizon: carry read from server-reported cached tokens, with $v\,p > c$ and $N^* < H$. Recognition and pay-back concern evaluation, retirement and representation design; all four are measurable with the ledger, and none of the artifacts in this paper met the first.

7 Related Work

Persisting experience across episodes goes back to Reflexion (Shinn et al., 2023), Voyager's skill library (Wang et al., 2024a), ExpeL (Zhao et al., 2024), and Dynamic Cheatsheet (Suzgun et al., 2026), ACE's ancestor; earlier, Self-Refine (Madaan et al., 2023) and bootstrapped self-training (Zelikman et al., 2022; Huang et al., 2023; Yuan et al., 2024) improved a model from its own outputs, and intrinsic self-correction without external feedback does not help reasoning (Huang et al., 2024). Surveys (Jiang et al., 2026; Ren et al., 2026) organize methods by what they update; evaluations (Wei et al., 2026; Wang et al., 2026; Xiong et al., 2026; Zhao et al., 2026) score them along a stream. AgentStream (Yan et al., 2026) evaluates five such methods across six agentic benchmarks (including Trivedi et al., 2024; Patil et al., 2025; Jimenez et al., 2024; Barres et al., 2025) and three streaming scenarios. All of these report accuracy along the stream; none prices the artifact or measures it frozen.

Portability has been measured before. SEAGym (Zheng et al., 2026) reports asymmetric cross-backend transplant of evolved harnesses; MCE (Ye et al., 2026) finds transferred ACE-style contexts can fall below the receiving model's baseline; EvoAgentBench (Gao et al., 2026) finds curated content transfers where automatic methods do not sustain gains; AFTER (Belikova et al., 2026) finds multi-model skills transfer better than single-model ones. We add what these lack: a frozen, content-addressed artifact, phase-attributed costs, a gate fixed in advance, and a horizon.

On cost, cost-controlled evaluation (Kapoor et al., 2024; Erol et al., 2026) prices accuracy per dollar but treats the agent's artifacts as fixed; prefix caching (Kwon et al., 2023; Zheng et al., 2024; Gim et al., 2024) rewards a stable artifact. Don't Break the Cache (Lumer et al., 2026) measures the savings of cache-preserving agent design but treats the prompt as fixed, and practitioner reports (0xAlcibiades, 2026) describe memory injection rebuilding a cached system prompt, the mechanism our churn account prices. Provider tariffs (OpenAI, 2026a; Anthropic, 2026a) and retirement floors (OpenAI, 2026b; Anthropic, 2026b) make the economics concrete; silent drift within a model id (Chen et al., 2023) is the evaluation-side twin.

Finally, self-evolution is the agentic form of test-time training (Sun et al., 2020; Hardt & Sun, 2024); its artifact plays the role parameters play in continual learning (Wang et al., 2024b), where forgetting (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017) is the classic depreciation.

8 Conclusion

By the books kept here, every artifact we measured is an expense, not an asset. The cost accounts are certain: carrying the artifact raises execute cost per task by 48% on GPT-5.4 and overtakes its build cost after 67 tasks, online rewriting adds about one cache miss per task, and the successor inherits the artifact at zero measured value and double the carry. The value account is bounded rather than measured: no frozen artifact showed significant positive value in eight rounds, the frozen value on the most favorable column is at most $+0.09$ (one-sided 95%), the bootstrap mass above the break-even value is 10%, and GPT-5.5's own playbook is significantly harmful after Holm correction. Two results are exploratory: the artifact changes how long the agent talks, and a syntax-destroyed copy reproduces most of its footprint.

Limitations and threats to validity. *Statistical:* with $n = 50$–64 the gate detects only effects above about 12 points (no evidence of absence); C1 has two replicates whose contrasts have opposite signs, every other round one replicate and every backbone a single build, so a harmful playbook could be one unlucky trajectory (overstates harm); the extension of C1 was decided after the 50-task result (direction unknown). *Construct:* the placebo preserves vocabulary, so keyword effects are not excluded (understates what the content contributes); "label-free" is instantiated by ACE and one ReasoningBank round (narrows the claim). *Confounds:* R5's loss coincides with context-limit truncation (overstates harm); the three large backbones differ in reasoning effort and benchmark (no trend is claimed); our no-asset baseline solves 74% of the build tasks where AgentStream reports 50% (understates the online gain). *External:* one method, three columns, one vendor (narrows generality).

AI USE STATEMENT

The authors conceived the research question and the asset-accounting framing, defined the four accounts and the recognition gate, designed the experiments, chose the benchmarks, models, and rounds, interpreted the results, and verified every number reported here against the archived ledgers; they take full responsibility for the contents. Within that design an LLM coding assistant (Claude, Anthropic) was used, under the authors' direction and review, for the following subtasks: implementing the ledger, freeze mode, snapshot format, runner, and analysis scripts; launching and monitoring runs; producing the figures and tables from the archived ledgers; and drafting text and captions from the archived ledgers, which the authors rewrote and edited. LLMs are also the object of study: the self-evolving methods, the agents, and the tau2 user simulator are all LLM-driven, as described in Section 3. No LLM-generated content was used as a source of factual claims without verification against the cited sources or the ledgers.

REPRODUCIBILITY STATEMENT

The ledger, freeze mode, snapshot format, runner, and analysis scripts are released; every run ships a manifest (pinned model id or revision SHA, sampling, cache regime, task-id hash, and the repository commit at run time) and an append-only SQLite ledger whose audit must pass before the run is used. The commit recorded in the manifests predates the ledger code: the runs were made from a working tree ahead of it, and the code that ran is the released archive, which contains that tree. Closed-model runs record the dated snapshot ids used (`gpt-5.4-2026-03-05`; `gpt-5.5-2026-04-23`), the public tariff version, and the user-simulator snapshot. The figures are regenerated from the archived ledgers by the released scripts.

REFERENCES

0xAlcibiades. Honcho auto-injected context rebuilds cached system prompt every N turns, invalidating KV cache on every prefix-caching backend. GitHub issue #13631, NousResearch/hermes-agent, https://github.com/NousResearch/hermes-agent/issues/13631, 2026. Opened 2026-04-21; accessed 2026-09-26.

Anthropic. Prompt caching. https://platform.claude.com/docs/en/build-with-claude/prompt-caching, 2026a. Read 2026-09-22.

Anthropic. Model deprecations. https://platform.claude.com/docs/en/about-claude/model-deprecations, 2026b. Model lifecycle page; read 2026-09-22.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Gary S. Becker. *Human Capital: A Theoretical and Empirical Analysis, with Special Reference to Education*. National Bureau of Economic Research / Columbia University Press, 1964.

Julia Belikova, Rauf Parchiev, Evgeny Egorov, Grigorii Davydenko, Gleb Gusev, Andrey Savchenko, and Maksim Makarenko. Managing procedural memory in LLM agents: Control, adaptation, and evaluation, 2026. URL https://arxiv.org/abs/2606.23127.

Katherine S. Button, John P. A. Ioannidis, Claire Mokrysz, Brian A. Nosek, Jonathan Flint, Emma S. J. Robinson, and Marcus R. Munafò. Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14(5):365–376, 2013. doi: 10.1038/nrn3475.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9263–9274. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.745. URL https://aclanthology.org/2020.emnlp-main.745/.

Lingjiao Chen, Matei Zaharia, and James Zou. How is ChatGPT's behavior changing over time?, 2023. URL https://arxiv.org/abs/2307.09009.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Avinash K. Dixit and Robert S. Pindyck. *Investment under Uncertainty*. Princeton University Press, 1994.

Mehmet Hamza Erol, Batu El, Mirac Suzgun, Mert Yuksekgonul, and James Y Zou. Cost-of-pass: An economic framework for evaluating language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 34103–34127, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/399d54cce01e782192b3d7ecbe607a96-Paper-Conference.pdf.

Xingze Gao, Chuanrui Hu, Hongda Chen, Pengfei Yao, Zhao Wang, Yi Bai, Zhengwei Wu, Yunyun Han, Xiaofeng Cong, Jie Gui, Yafeng Deng, and Teng Li. EvoAgentBench: Benchmarking agent self-evolution via ability transfer, 2026. URL https://arxiv.org/abs/2607.05202.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In P. Gibbons, G. Pekhimenko, and C. De Sa (eds.), *Proceedings of Machine Learning and Systems*, volume 6, pp. 325–338, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/file/a66caa1703fe34705a4368c3014c1966-Paper-Conference.pdf.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 54625–54640, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/f02f1185b97518ab5bd7ebde466992d3-Paper-Conference.pdf.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1068. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.67. URL https://aclanthology.org/2023.emnlp-main.67/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32808–32824, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8b4add8b0aa8749d80a34ca5d941c355-Paper-Conference.pdf.

International Accounting Standards Board. IAS 36 impairment of assets. IFRS Foundation, https://www.ifrs.org/issued-standards/list-of-standards/ias-36-impairment-of-assets/, 2004a. Issued by the International Accounting Standards Committee in 1998, revised by the IASB in 2004; consolidated text accessed 2026-09-26.

International Accounting Standards Board. IAS 38 intangible assets. IFRS Foundation, https://www.ifrs.org/issued-standards/list-of-standards/ias-38-intangible-assets/, 2004b. Issued by the International Accounting Standards Committee in 1998, revised by the IASB in 2004; consolidated text accessed 2026-09-26.

John P. A. Ioannidis. Why most discovered true associations are inflated. *Epidemiology*, 19(5): 640–648, 2008. doi: 10.1097/EDE.0b013e31818131e7.

Pengcheng Jiang, Jiacheng Lin, Zhiyi Shi, Zifeng Wang, Luxi He, Yichen Wu, Ming Zhong, Peiyang Song, Qizheng Zhang, Heng Wang, Xueqiang Xu, Hanwen Xu, Pengrui Han, Dylan Zhang, Jiashuo Sun, Chaoqi Yang, Kun Qian, Tian Wang, Changran Hu, Manling Li, Quanzheng Li, Hao Peng, Sheng Wang, Jingbo Shang, Chao Zhang, Jiaxuan You, Liyuan Liu, Pan Lu, Yu Zhang, Heng Ji, Yejin Choi, Dawn Song, Jimeng Sun, and Jiawei Han. Adaptation of agentic AI: A survey of post-training, memory, and skills, 2026. URL https://arxiv.org/abs/2512.16301.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/edac78c3e300629acfe6cbe9ca88fb84-Paper-Conference.pdf.

Slava Kalyuga, Paul Ayres, Paul Chandler, and John Sweller. The expertise reversal effect. *Educational Psychologist*, 38(1):23–31, 2003. doi: 10.1207/S15326985EP3801_4.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter, 2024. URL https://arxiv.org/abs/2407.01502.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Barbara Levitt and James G. March. Organizational learning. *Annual Review of Sociology*, 14: 319–338, 1988. doi: 10.1146/annurev.so.14.080188.001535.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf.

Elias Lumer, Faheem Nizar, Akshaya Jangiti, Kevin Frank, Anmol Gulati, Mandar Phadate, and Vamse Kumar Subbiah. Don't break the cache: An evaluation of prompt caching for long-horizon agentic tasks, 2026. URL https://arxiv.org/abs/2601.06007.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H. Bower (ed.), *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989. doi: 10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations, 2024. URL https://arxiv.org/abs/2411.00640.

Richard R. Nelson and Sidney G. Winter. *An Evolutionary Theory of Economic Change*. Belknap Press of Harvard University Press, Cambridge, MA, 1982.

Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018. doi: 10.1073/pnas.1708274114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1708274114.

OpenAI. Prompt caching. https://platform.openai.com/docs/guides/prompt-caching, 2026a. Read 2026-09-22.

OpenAI. Deprecations. https://platform.openai.com/docs/deprecations, 2026b. Model lifecycle page; read 2026-09-22.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/980ea04d23d1f6908964eba2a74afe45-Paper-Conference.pdf.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

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*, UIST '23. Association for Computing Machinery, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

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. URL https://openreview.net/forum?id=2GmDdhBdDk.

Zhe Ren, Yimeng Chen, Dandan Guo, Guowei Rong, Tonghui Li, R. B. Xiong, Qingfeng Lan, Wenyi Wang, Li Nanbo, Yibo Yang, Mingchen Zhuge, and Jürgen Schmidhuber. Self-improvements in modern agentic systems: A survey, 2026. URL https://arxiv.org/abs/2607.13104.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Robert M. Solow. Investment and technical progress. In Kenneth J. Arrow, Samuel Karlin, and Patrick Suppes (eds.), *Mathematical Methods in the Social Sciences, 1959*, pp. 89–104. Stanford University Press, 1960.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.333. URL https://aclanthology.org/2026.eacl-long.333/.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

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. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024b. doi: 10.1109/TPAMI.2024.3367329.

Yuyao Wang, Zhongjian Zhang, Mo Chi, Kaichi Yu, Yuhan Li, Miao Peng, Bing Tong, Chen Zhang, Yan Zhou, and Jia Li. EvoMemBench: Benchmarking agent memory from a self-evolving perspective, 2026. URL https://arxiv.org/abs/2605.18421.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory, 2024c. URL https://arxiv.org/abs/2409.07429.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H. Chi, Chi Wang, Shuo Chen, Fernando Pereira, Wang-Cheng Kang, and Derek Zhiyuan Cheng. Evo-Memory: Benchmarking LLM agent test-time learning with self-evolving memory, 2026. URL https://arxiv.org/abs/2511.20857.

Oliver E. Williamson. *The Economic Institutions of Capitalism: Firms, Markets, Relational Contracting*. Free Press, New York, 1985.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.27. URL https://aclanthology.org/2026.acl-long.27/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Dong Yan, Jian Liang, Dapeng Hu, Ran He, Nicholas Jing Yuan, Qi Zhang, and Tieniu Tan. AgentStream: How well do self-evolving LLM agents perform under streaming tasks?, 2026. URL https://arxiv.org/abs/2608.00155.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. AutoSkill: Experience-driven lifelong learning via skill self-evolution, 2026. URL https://arxiv.org/abs/2603.01145.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Haoran Ye, Xuning He, Vincent Arak, Haonan Dong, and Guojie Song. Meta context engineering via agentic skill evolution, 2026. URL https://arxiv.org/abs/2601.21557.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57905–57923. PMLR, 2024. URL https://proceedings.mlr.press/v235/yuan24d.html.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. doi: 10.52202/068431-1126. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8a94ff6f922d995d7d3f4ebf4143e442-Paper-Conference.pdf.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

Weixiang Zhao, Yingshuo Wang, Yichen Zhang, Yang Deng, Yanyan Zhao, Wanxiang Che, Bing Qin, and Ting Liu. Large language model agents are not always faithful self-evolvers, 2026. URL https://arxiv.org/abs/2601.22436.

Congjie Zheng, Chuanyi Xue, Bin Liang, Jun Yang, and Changshui Zhang. SEAGym: An evaluation environment for self-evolving LLM agents, 2026. URL https://arxiv.org/abs/2606.17546.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 62557–62583. Curran Associates, Inc., 2024. doi: 10.52202/079017-2000. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/724be4472168f31ba1c9ac630f15dec8-Paper-Conference.pdf.

A Run configurations

Every run writes a manifest with the pinned model (HuggingFace revision SHA for open weights; dated snapshot for providers), serving configuration, sampling, cache regime, task-id hash, and the repository commit at run time (which, for the reasons given in the reproducibility statement, predates the ledger code). Open rounds: R1 GSM8K and R2 BFCL on Qwen3-4B (RTX 5070 Ti, vLLM), R3-A on Qwen3-8B-AWQ, R3-B on Qwen3-4B with ACE's playbook budget set to 2,000 tokens (a prompt-only knob; the ledger showed no effect on asset size), R4 ReasoningBank on Qwen3-8B, R5 ACE on gpt-oss-120b (revision `b5c939de`) transplanted to Qwen3-32B-AWQ (revision `0499c3ac`) on one H100 with vLLM 0.29.0 at 40,960 context; targets for R1–R4 are granite-3.3-8B. Table 3 lists the gate outcomes. Closed: C1 ACE on `gpt-5.4-2026-03-05`, reasoning effort medium, provider-default sampling, user simulator `gpt-5.4-2026-03-05` at temperature 0, tau2-bench v0.1.3 telecom, build = AgentStream's 50-task selection (seed 42), held-out = the remaining 64 (the first 50 run first; the last 14 as separate runs, labeled C6, with the same snapshot, settings, and simulator, merged per task); the decomposition run scores the build tasks with no asset. C2 repeats the source side on `gpt-5.5-2026-04-23` and transplants C1's frozen playbook onto it;

that snapshot rejects `reasoning_effort` together with function tools on the chat completions endpoint, so C2 runs at the provider's default reasoning effort (recorded in the manifest), with the same user-simulator snapshot as C1. C7 scores the 64 held-out tasks on `gpt-5.4-2026-03-05` with a placebo snapshot: the frozen C1 playbook with the words of every entry shuffled (seed 7), preserving length (37,262 characters, 5,619 words), vocabulary, section headers, and entry scaffolds; the placebo snapshot records the id of the asset it was derived from, and the run uses C1's settings and simulator with one replicate after a three-task smoke test. Frozen measure runs used four concurrent sessions; build runs are sequential because evolution order is part of the measurement. Before any closed run the runner makes one tool-calling request with the run's exact settings through the ledger, so a parameter the provider rejects only when tools are present fails before a build is paid for.

Table 2: Coverage of the AgentStream suite by the runs in this paper. The ledger instruments every method and benchmark adapter in the suite; cells list the rounds that were run. Scale: 54 runs (30 open-weight, 24 closed), 2,212 agent sessions, 51K ledgered LLM calls, 482M agent-side tokens plus 63M user-simulator tokens on the closed side ($176 at public tariffs). A-Mem, AutoSkill, and Harness are instrumented but not run (cost; A-Mem's embedding model is one more apparatus to pin). Backbones: Qwen3-4B, Qwen3-8B, Qwen3-8B-AWQ, Qwen3-32B-AWQ, gpt-oss-120b, granite-3.3-8B (open); gpt-5.4-2026-03-05, gpt-5.5-2026-04-23 (closed).

Method \ Benchmark	GSM8K	BFCL	Tau2-telecom	AppWorld	BrowseComp-Plus / HLE	SWE-bench-Verified
ACE	R1 (sanity column)	R2, R3-A, R3-B, R5	C1, C2, C6, C7	–	–	–
ReasoningBank	–	R4	attempt (Appendix F)	–	–	–

Table 3: Gate outcomes of all rounds. r_{src} is the paired with−without gain on held-out tasks (50 for R1–R5, 64 for C1, 49 for C2) with a 95% task-clustered bootstrap interval; Holm-corrected bootstrap p-values: C2 < 0.001, R1 and R5 0.22, all others ≥ 0.63; C1's transplant onto gpt-5.5 gives $r_{\mathrm{tgt}} = 0.00$ $[-0.08, +0.08]$.

Round	Benchmark	Source → target backbone (method)	r_{src} [95% CI]	Gate
R1	GSM8K	Qwen3-4B → granite (ACE)	-0.12 $[-0.22, -0.02]$	refused
R2	BFCL	Qwen3-4B → granite (ACE)	-0.12 $[-0.26, +0.02]$	refused
R3-A	BFCL	Qwen3-8B-AWQ → granite (ACE)	$+0.08$ $[-0.06, +0.22]$	refused
R3-B	BFCL	Qwen3-4B, playbook budget 2000 (ACE)	-0.10 $[-0.24, +0.02]$	refused
R4	BFCL	Qwen3-8B (ReasoningBank)	-0.02 $[-0.12, +0.08]$	refused
R5	BFCL	gpt-oss-120b → Qwen3-32B-AWQ (ACE)	-0.16 $[-0.30, -0.02]$	refused
C1	Tau2-telecom	gpt-5.4-2026-03-05 → gpt-5.5-2026-04-23 (ACE; 2 seeds, $n = 64$)	$+0.02$ $[-0.06, +0.11]$	refused
C2	Tau2-telecom	gpt-5.5-2026-04-23 (ACE, own asset)	-0.24 $[-0.39, -0.12]$	refused (negative)

B Reference tables

Table 4: Where AgentStream reports self-evolution gains (ACE, Isolated scenario; values are accuracy %).

Model	BFCL vanilla→ACE	Tau2-telecom vanilla→ACE	AppWorld vanilla→ACE	Mean gain
GPT-5.4	$66.0 \rightarrow 68.0 \pm 12.2$	$50.0 \rightarrow 63.3 \pm 6.1$	$44.6 \rightarrow 39.1$	-0.35
Gemini 3.1 Pro	$58.0 \rightarrow 62.7 \pm 13.3$	$90.0 \rightarrow 90.7$	$41.8 \rightarrow 43.2$	$+2.71$
Claude Opus 4.7	$86.0 \rightarrow 85.3 \pm 1.2$	$80.0 \rightarrow 93.3 \pm 3.1$	$41.2 \rightarrow 47.5$	$+1.75$

Table 5: Useful life of closed-model snapshots in the lifetime dataset (46 ids; vendor lifecycle pages, read 2026-09-22). Lifetimes are release-to-retirement for realized retirements and release-to-floor for dated active ids; notice is announcement-to-retirement for realized retirements. Cells give the median and the interquartile range.

Vendor	Snapshots	Lifetime, all (months)	Lifetime, 2025+ releases	Notice (days)
Anthropic	10 retired, 3 dated floors, 8 undated ids	13.0 [12.0, 25.4]	12.5 [12.0, 13.0]	69 [60, 189]
OpenAI	25 retired or deprecated	19.8 [14.2, 40.3]	18.0 [14.2, 20.7]	184 [92, 458]

C Ledger schema

```
call_events(event_id, run_id, ts_utc, agent, session_id,
task_id, phase, attribution, asset_ids, model, prompt_tokens,
```

Table 6: Open-side projection made before any closed run: the R5 ledger (gpt-oss-120b, BFCL) priced as a GPT-5.4 deployment (per task, 50 tasks; instrumented tokens × exogenous tariff). Superseded by the measured accounts in Table 1.

Quantity	Value
Frozen asset size	5,703 tokens
Prompt tokens per execute call, with / without	11,108 / 4,993
Execute calls per task, with / without	40.1 / 25.5
Carry: extra prompt tokens per task	317,770 (46% injection, 51% induced calls, 3% history)
Carry per task	\$0.213 – \$0.928
Build cost B per task	\$0.077
Churn (online − frozen) per task	\$0.0067

`completion_tokens, cached_read_tokens, cache_write_tokens, reasoning_tokens, usage_source, wall_ms, cache_regime, prefix_hash, cache_hit, outcome, retry_of, error, callsite, extra)`. The table is append-only; the run-end audit refuses a run with any `phase=unknown` record, folds the write-ahead log so the archived file is self-contained, and reports token totals by phase. `extra` carries the per-call dollar cost from the public price sheet and the price-sheet version. Benchmark-side calls that run in a separate process (the tau2 user simulator) are folded in per session from the benchmark's own result files as `phase=benchmark`, attribution `exogenous`. The ledger reconciles with the bill: summed over every closed-model run, it prices \$176.04 for 202.2M tokens, against \$180 and 200.2M tokens on the provider's usage dashboard for the same key and period (29.3K requests there; the ledger holds 13.2K agent-side requests and folds the simulator's requests per session). The gap is three aborted builds recorded before the price table covered their model and a handful of out-of-ledger gateway probes.

D Power

With binary task scores $s_i^{\pm}$ and paired difference $d_i = s_i^+ - s_i^- \in \{-1, 0, 1\}$, the observed standard deviation of d_i across our rounds is 0.40–0.48. With $n = 50$ the standard error of the mean is ≈ 0.064 and the bootstrap 95% interval half-width ≈ 0.12–0.14, as observed (Table 3); with $n = 64$ and two replicates averaged per task it is ≈ 0.08. Detecting a true $+0.13$ at $\alpha = 0.05$ has power ≈ 0.55; a true $+0.02$ would need $n \approx 2{,}000$. The replicates in C1 put a number on the sampling component of that noise: two runs of the same condition at the provider's default temperature disagree on 25–27% of tasks.

E Derived quantities

Table 7: Round C2 (GPT-5.5, own frozen playbook vs. none, 49 held-out tasks): outcome flips by task domain and customer persona. All thirteen losses are MMS tasks (base rate 25/49 = 51%; binomial $p \approx 2 \times 10^{-4}$).

Domain (persona)	n	same	solved → failed	failed → solved
MMS issue	25	11	13	1
Mobile-data issue	13	13	0	0
Service issue	11	11	0	0
Persona of the 13 losses			Easy 8, None 3, Hard 2	(held-out: Easy 18, None 17, Hard 14)

F Recorded failure modes

Two C1 sessions (one no-asset on the first 50 held-out tasks, one with-asset in the 14-task extension) ended when GPT-5.4 returned a message with neither content nor a tool call; tau2 raised `AssistantMessage must have either content or tool calls` and wrote no result. Each session is scored 0 (the customer was not served) and kept in the pair, and the runner now reads scores from the per-session result files, since the framework's in-memory aggregate dropped

Table 8: Provenance of derived quantities. All are computed from unrounded inputs; displayed inputs are rounded.

Quantity	Formula	Inputs	Value
Carry c	execute cost with − without, per task	\$0.1720 − \$0.1160 (seed 1); \$0.1620 − \$0.1060 (seed 2)	\$0.056
Carry relative to execute cost	c / execute cost without	0.056 / 0.116	48%
Pay-back of build	B/c	3.77 / 0.056	67.3 tasks
Upgrade price p	Δcost / Δsuccess, GPT-5.4 → 5.5	(0.1877 − 0.1162) / (0.918 − 0.82)	\$0.7296 per solved task
Break-even value v^*	c/p	0.056 / 0.7296	0.0768
Margin at v	$v\,p - c$	$v = 0.0234$; $v = 0.11$	−\$0.0389; +\$0.0245
N^* at upper bound	$B/(v\,p - c)$	$v = 0.11$	155 tasks
One-sided 95% upper bound on r_{src}	bootstrap quantile	20,000 task-clustered resamples	+0.094
$P(v > v^*)$	bootstrap mass above v^*	same resamples	0.10
Holm-corrected p	Holm over 8 rounds	bootstrap two-sided p	C2 0.0008; R1, R5 0.22
Never-helpful token share	tokens of entries with helpful= 0 / all entry tokens	76 of 106 entries	74%
Carry saved by retiring them	0.70 × c× share	0.70 × 0.056 × 0.737	\$0.029 per task

sessions under concurrency. A second session (C2, no asset) stalled after 15 steps without an error and is excluded rather than scored.

A method that could not be run. We attempted the same protocol with ReasoningBank on `gpt-5.4-2026-03-05`. The build completed (50 tasks, ledger audit valid, \$9.0) but the agent solved 1 of 50 tasks online and made no tool call in 48 of 50 sessions: ReasoningBank's prompt asks the model to write "Memory Item k: I will use this because …" before acting, and GPT-5.4 at reasoning effort medium returns that preamble as its customer-facing message and stops, so the simulator receives it and the environment is never touched. AgentStream reports 46.7% for this method–model pair, so the incompatibility is specific to the released harness or the snapshot; repairing it would mean editing the method's prompts, which this study forbids, so the frozen runs were cancelled and the ledger is archived as a recorded failure.

G What the asset looks like

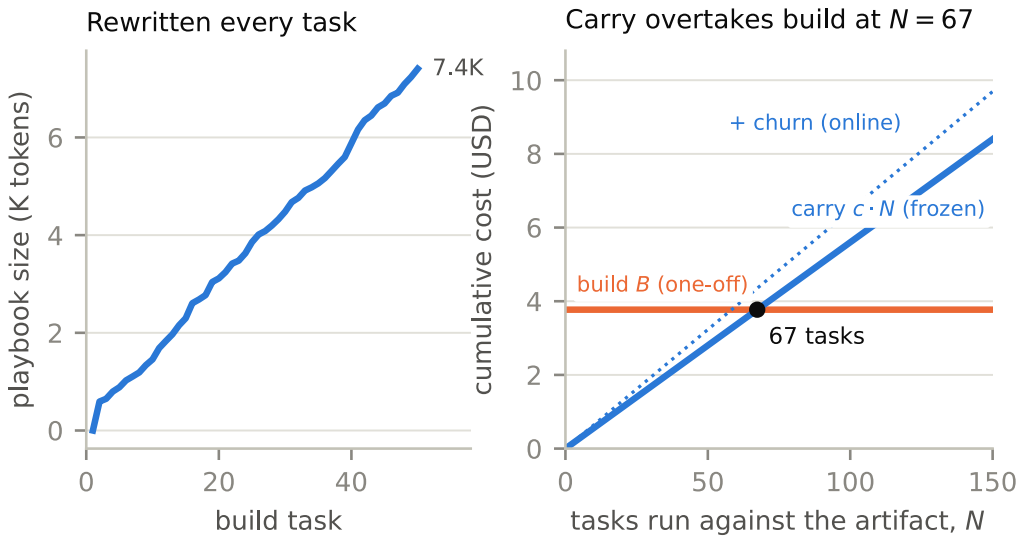

Figure 4: Owning the GPT-5.4 asset. **Left:** the playbook grows to 7.4K tokens over the 50 build tasks (7.6K as rendered into the prompt) and is rewritten after every one of them (50 distinct prefix hashes). **Right:** cumulative cost: the one-off build (\$3.77) is overtaken by the per-task carry (\$0.056) after 67 tasks and the gap widens; the snapshot's published retirement horizon caps how many tasks can ever be run against it.

The frozen GPT-5.4 playbook (37K characters, 7.6K tokens) has 106 entries in three of ACE's seven sections (58 strategies, 34 common mistakes, 14 heuristics; the formula, code, and context sections stay empty on this domain). Each entry carries the counters ACE's curator maintains; 76 of 106 were never marked helpful and 33 were marked harmful at least once, yet all are injected on every call. Verbatim excerpts:

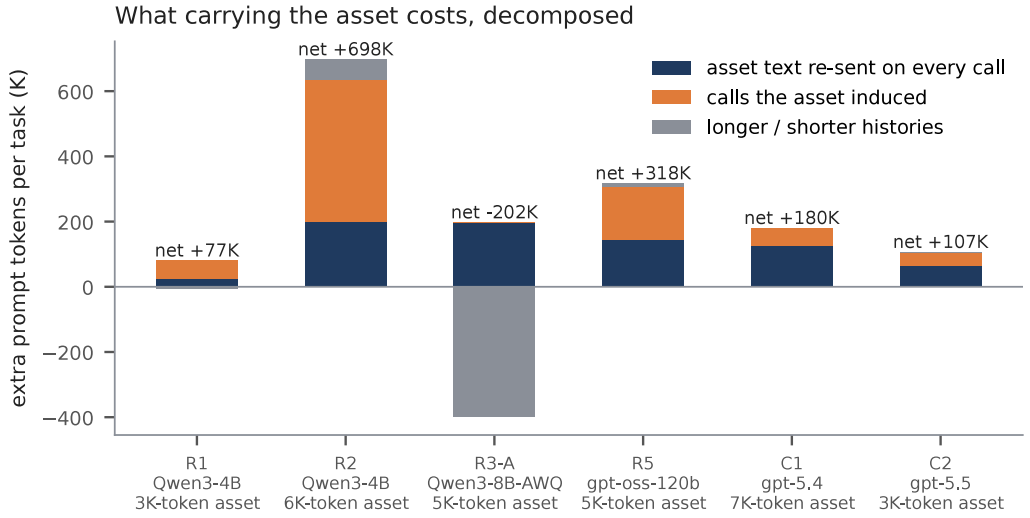

Figure 5: Extra prompt tokens per task caused by carrying the frozen asset, decomposed into the asset text re-sent on every call, the additional calls the asset induced, and the change in history length; one bar per round with the source backbone and the asset size. On weaker open models the induced calls dominate; on the closed snapshots the asset text does; on R3-A the asset shortens the run and the net carry is negative.

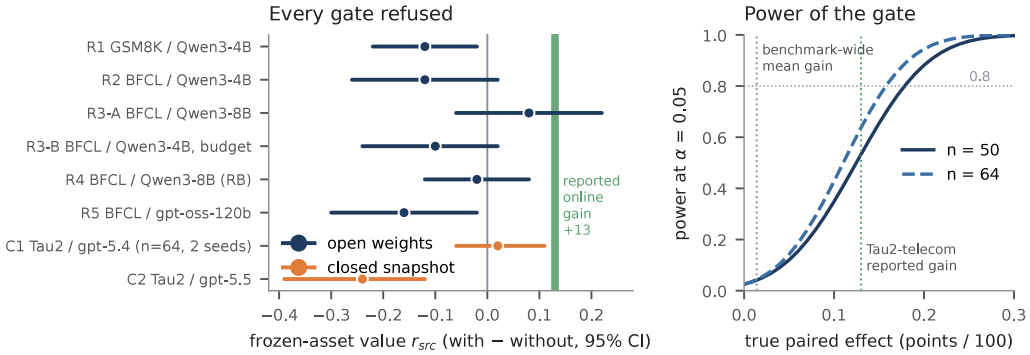

Figure 6: **Left:** frozen-asset value r_{src} (with $-$ without on held-out tasks, 95% task-clustered bootstrap CI) for all eight rounds; the green band is the $+13$-point online gain AgentStream reports on Tau2-telecom. Every round is refused by the recognition gate; only C2 is significant after Holm correction. **Right:** power of the paired gate against the true effect size at $n = 50$ and $n = 64$; the benchmark-wide mean gain is invisible at any affordable n, the Tau2-telecom gain is detectable with power 0.55–0.62.

```
[str-00002] helpful=48 harmful=0 ::  Keep troubleshooting in
strict stage order:  identify customer, determine affected
line, verify account/line status and obvious account-side
blockers, then move to device diagnostics, then validate
improvement with a final check.
[str-00006] helpful=35 harmful=0 ::  If the user says they
can only do one thing at a time, switch to single-step
troubleshooting:  request exactly one action or one
observation per turn, then wait for the result before giving
the next step.
[err-00010] helpful=0 harmful=49 ::  Do not reference
results that have not yet been provided by the user or
returned by a tool.  Strictly ground every conclusion
in currently visible dialogue context and completed tool
```

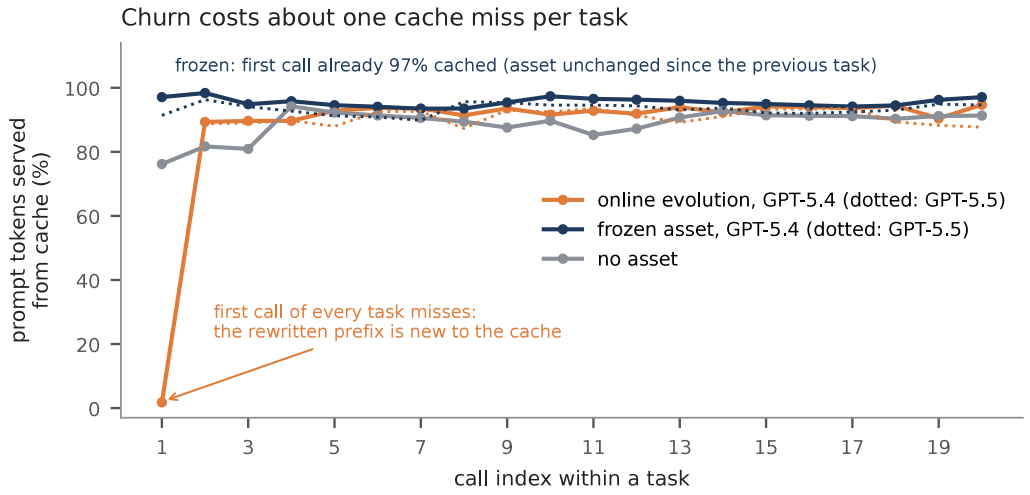

Figure 7: Share of prompt tokens served from the provider cache, by call index within a task, averaged over 50 tasks. Under online evolution (orange) the first call of every task misses because the playbook was rewritten after the previous task; from the second call on the growing conversation is cached whether or not the asset changed. With a frozen asset (navy) the first call already hits at 97%. Churn therefore costs about one uncached prefix per task, on both GPT-5.4 (solid) and GPT-5.5 (dotted).

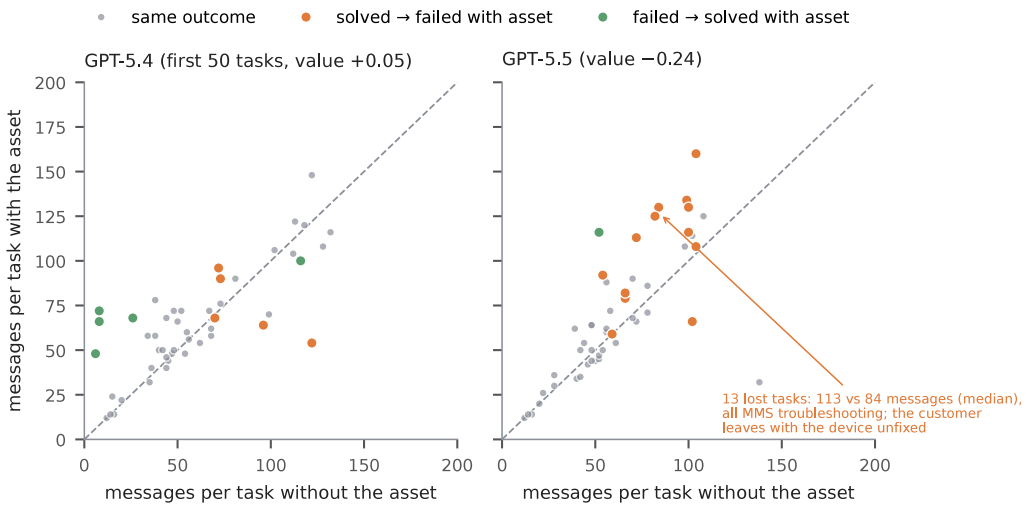

Figure 8: Conversation length per task with vs. without the model's own frozen playbook, colored by outcome flip. **Left:** GPT-5.4 (value $+0.05$ on these 50 tasks): flips are few and show no length pattern. **Right:** GPT-5.5 (value -0.24): the 13 tasks the asset loses all lie above the diagonal: the agent talks longer (median 113 vs. 84 messages), the simulated customer leaves, and the device is never put into the right state.

```
outputs.
[err-00013] helpful=0 harmful=47 ::  Do not perform
unnecessary line-detail checks on unrelated lines after
the affected phone number is known.  Extra lookups increase
the risk of acting on the wrong line and create avoidable
confusion.
```

The content is procedural advice about the order and pacing of the conversation, not domain knowledge the model lacks; this is what "behavior modifier" means concretely, and why the same text can lengthen conversations on a model that would otherwise have moved faster.