

Similarity Drowns Intent: Three-Trace Sagas and Reinstatement Recall for Provenance in Agentic Software Construction

Rama Annaswamy
Independent Research

July 25, 2026

Abstract

Flat similarity retrieval fails *provenance* questions — “why does this code look like this?” — in a characteristic way: it returns restatements and re-askings of the question instead of the conversations where the decision was made. We study these questions over an agent’s own recorded history, organized as *sagas* — joint traces of *intent*, *deliberation*, and *artifact* — and present *reinstatement recall*, a seed-conditioned multi-hop retrieval algorithm that is LLM-free and sub-100 ms warm.

Across two within-operator corpora from unrelated codebases, reinstatement improved ground-truth session coverage over one-shot kNN by +53% and +47% at equal budget under pre-registered gates, with zero per-query losses in those coverage runs; blind cross-vendor judge panels preferred it on both corpora (instructions disclosed as hypothesis-aligned). On graded provenance gold derived from the operator’s own ratification behavior (sealed pre-registered origin recall, dual-vendor dialog-act extraction, external ship-event ledgers), reinstatement wins on origin-MRR (+37%), $nDCG@10$ (+29%), and graded recall (+25%) — point estimates on 12 mapped queries, without interval estimates — while both systems miss the verified origin conversation on over half the queries, locating the open headroom. On this decision-graded gold a channel ablation reverses the coverage-proxy attribution: code-graph spread — the session $\leftrightarrow$ file co-edit trace, not the AST symbol graph — carries the provenance signal, as a within-grid ranking on decision-graded gold rather than a causal attribution. A five-arm sham-controlled follow-up on the same sealed gold supplies the missing intervals and sharpens that claim in two directions: the co-edit channel’s marginal contribution over the otherwise-full stack is +0.017 origin-MRR (95% paired-bootstrap CI [0.000, +0.050], lower bound exactly zero) and +0.026 $nDCG@10$ (CI includes zero) on the 12 mapped-origin queries of 20 — directionally consistent, not separable at this power; and an AST-structural expansion arm is indistinguishable from both the base arm (identical origin-MRR on 12 of 12 mapped queries) and from its own degree-preserving edge-shuffle sham ($\Delta nDCG@10$ +0.0002, CI [−0.0132, +0.0139]), so the structural graph carries no detectable ranking signal on this corpus. What that

layer is measured to deliver is integrity rather than ranking: 98.1% witness closure, 86.4% internal binding, zero drifted edges, and visible abstention. Evaluating the system surfaced a second finding: *self-indexing evaluation contamination*, in which the system ingests the evaluation dialogue itself and echoes of the questions displace the origin conversations being sought. A controlled dose-response experiment from a session-zero snapshot shows five scripted re-ask cycles capturing half of naive retrieval’s top-10, with query-echo defenses repairing 72% of the echo occupancy (sham control null). A blinded three-arm behavioral benchmark (16 sealed re-litigation tasks) shows conversation memory roughly halving reason-quality failures on settled decisions relative to a no-memory arm — while reinstatement and one-shot kNN tie exactly in this single-hop regime. A subsequent pre-registered cross-source benchmark then tested the multi-hop scope directly: no R/K separation on 27 sealed questions (an underpowered null — both arms fail most multi-hop questions), and a decisive reversal on 396 exact receipt-lookup queries, where hybrid kNN+FTS beats the walk 0.813 to 0.581 ($p \approx 10^{-19}$). Ablation traces the reversal to the echo defenses themselves: the verbatim-overlap demotions that repair contamination invert into anti-lookup on receipt-class queries — the walk scores 12 points below its own seed channel. Contamination defense and provenance lookup compete for the same signal, arguing for query-class routing rather than one retrieval policy; a pre-registered ratification-weighting hypothesis halted at $\rho \approx 0$ across three extractor generations, and that negative result directly forbade an inference-based staleness design in the deployed system, yielding an asserted-state resolution ledger instead. We argue that frozen pre-evaluation snapshots, shared index builds, and echo defenses are necessary controls for any self-recording agent memory — and offer direct evidence the discipline works: a fourth contamination vector (hook-recursion self-transcripts comprising 84% of a frozen corpus) was discovered post-hoc, and both gates replicated on a scrubbed corpus the same day. The constructive conclusion the results support is settlement: provenance signals that exist as machine receipts (80% of commits carry exact-key receipts in session transcripts) should be consumed as typed joins, not semantic neighborhoods. External and multi-operator validation remain open.

Keywords: episodic memory, agent memory, retrieval, provenance, temporal context model, MCP

1 Introduction

In *Programming as Theory Building*, Naur argued that source code is not the primary artifact of software work. The real artifact is the *theory* in programmers’ heads — the web of intentions, constraints, and rejected alternatives that explains why the code has the shape it does. When those people leave or forget, the theory is lost, and later maintainers inherit a brittle surface that no longer answers “why” (Naur, 1985).

Agentic software construction sharpens Naur’s problem. The “programmer” is no longer a stable human team but a transient human–agent collaboration stretched across many sessions, subagents, and tool traces. Intent is expressed in prompts; deliberation unfolds as reasoning and tool calls (including dead ends and sidechain chatter); artifact materializes as file-level edits. The theory that would justify a later design decision is distributed across those three traces and is *more* perishable than classical human theory-building: sessions end, context windows compact, subagent transcripts fragment into separate files, and the human who held the intent may never re-open the originating conversation.

Most retrieval systems for coding agents treat memory as a flat bag of text: embed utterances, retrieve top- k by cosine similarity, and hope the right chunk appears. For *lexical* or *local* questions (“where is `integrity_check` defined?”), this often works. For *provenance* questions — “why is the integrity check cached?”, “why was this API shape chosen?” — flat nearest-neighbor search fails on the corpora we study in a predictable way. The query is textually closest to agent restatements, tool-mechanic chatter, or later sessions that *discuss* the decision, not to the human-origin conversation where the decision was made. Similarity drowns intent.

The failure is easy to exhibit on a live system. Asked “why did we drop Qdrant?” on the corpus studied here, one-shot kNN’s top hit — at cosine similarity 0.984 — is the transcript of the user asking that same question earlier the same day; ranks two and three are further echoes of the question. The same query under reinstatement recall surfaces the migration analysis, the memory-limit incident preserved in a subagent transcript (“the Qdrant vector database had no memory limits set”), and the months-earlier session in which the replacement storage engine was actually built — a conversation that never contains the words “dropped Qdrant.” A retrieval system that answers a why-question with a recording of the question is not a memory; the rest of this paper is about what to build instead, and about how easily measurement itself reproduces this failure mode.

This paper makes two measured contributions and one design contribution, in that order. Measured: (1) *reinstatement recall*, a seed-conditioned multi-hop retrieval algorithm for provenance questions, evaluated against one-shot kNN on two corpora under pre-registered gates and blind cross-vendor judging; (2) *self-indexing evaluation contamination*, a reproducible failure mode of self-recording memory systems, with defenses that measurably repair it. Design: the *three-trace saga* — intent, deliberation, artifact as an organizing structure for agent memory. We are explicit about the boundary: mechanism attribution moved as the gold improved — raw channel counts against file-touch coverage credited the seed-conditioned semantic blend, and the controlled ablation against decision-graded gold (which we treat as authoritative) credited code-graph spread — and a third movement arrived with interval estimates: a five-arm sham-controlled grid on the same sealed gold leaves the within-grid ranking standing while showing that the co-edit channel’s marginal contribution is not separable from zero at 12 mapped-origin queries of 20, and that an AST-structural arm is indistinguishable from a degree-preserving edge-shuffle sham. Within-grid ranking survives; causal separation does not. We fix the term here once: throughout this paper *code-graph* names the session↔file co-edit trace (`code_evolution`) and nothing else — the AST symbol graph (`code_nodes` / `code_edges`) has never been read by the measured

retrieval path, and its measured contribution is provenance integrity, not provenance ranking. Source-factored use of the three traces (weighting intent vs. deliberation vs. artifact channels separately) is implemented in storage but not yet ablated; the saga is the architecture the results motivate, not one they isolate. We implement the system as claude-self-reflect (CSR): a single Rust binary with local MiniLM embeddings (384-dim, FastEmbed), HNSW vector search, SQLite storage, ast-grep code graph construction, fourteen MCP tools, and six Claude Code lifecycle hooks that capture sessions as they happen. The provenance path is reinstatement recall (`csr_why`): seed with one-shot kNN, blend query and seed context for a second hop, spread activation over a session-file code graph, hop one step along episode chains, then fuse and cap.

We evaluate on dogfooded multi-hop questions about CSR’s own development history, with pre-registered proceed/iterate/kill gates. The spike passes; production reimplementaion is faithful but lower under same-day corpus contamination. That discrepancy is itself a finding: self-indexing memory systems exhibit an observer effect that evaluation design must confront.

2 Related Work

Temporal context and reinstatement. Howard and Kahana’s Temporal Context Model (TCM) proposes that episodic recall is driven not only by item-item similarity but by reinstatement of a drifting context representation that bound items together at encoding (Howard & Kahana, 2002). Our context-blend step — forming $0.65 \cdot q + 0.35 \cdot s$ from query vector q and seed vector s , renormalizing, and re-searching — is a deliberate computational analogue: retrieval reinhabits the contextual neighborhood of a retrieved memory rather than matching the literal cue alone. The Context Maintenance and Retrieval (CMR) model extends TCM with source and organizational structure (Polyn, Norman, & Kahana, 2009). CMR motivates treating intent, deliberation, and artifact as distinct but contextually bound sources; our Phase-1 system still fuses them into a single blend rather than source-factored reinstatement, which we flag as future work.

Relevance feedback in information retrieval. Our context-blend step has a clear IR ancestor: Rocchio’s relevance feedback (Rocchio, 1971) and its pseudo-relevance variants expand a query toward the centroid of top-ranked documents and re-search — formally close to our $q' = \text{renorm}(0.65q + 0.35s)$. We claim no novelty for vector blending itself. The differences that matter here: PRF re-searches once from a *centroid* of assumed-relevant documents to improve topical recall, while reinstatement walks *per-seed* — each seed spawns its own blended re-query plus graph and episode expansions — and optimizes a different objective (origin-of-decision, not topical relevance). Recent evaluation work sharpens the contrast: Deng (2026), stress-testing agent-memory retrieval under entity collision with pinned BM25 floors and paired-bootstrap intervals, measures lexical PRF (RM3) as a *null* on intent-style memory queries — “no PRF expansion over the lexical channel substitutes for a dense encoder on intent-style queries” — and finds it regresses some strata via query drift. That drift, PRF’s canonical failure mode, reappears in self-indexing agent memory in an aggravated form: the corpus continuously ingests restatements of the very queries being asked, so drift is toward the system’s own echoes. The query-echo defenses of Phase 1.5

(verbatim-echo demotion, echo-aware seed selection) are, in IR terms, drift countermeasures specialized to self-recording corpora, and we position them as a contribution to that older literature as much as to agent memory. A dense per-seed centroid-PRF baseline under our shared reranker remains the fair head-to-head and is queued in the ablation grid.

Rational analysis of memory access. Anderson and Schooler’s rational analysis argues that human memory’s sensitivity to recency and frequency mirrors a Bayesian estimate of future need given environmental statistics (Anderson & Schooler, 1991). This supplies a theoretical warrant for *learned decay*: weighting retrieval by logged access patterns rather than fixed similarity alone. CSR previously logged retrieval only on the prompt-submit hook path; Phase 1 adds MCP-path logging so usage-weighted reinstatement becomes possible once logs accumulate (Phase 2, deferred).

Programming as theory building, and the software-engineering lineage. Naur’s thesis frames provenance recall as the product problem: the goal is not merely to find similar text but to reconstitute the theory that produced the code (Naur, 1985). In agentic settings, that theory lives disproportionately in the *intent* trace — human prompts and the human-agent dialogue that fixed constraints — while deliberation and artifact record how and what, respectively. Joint indexing without a path back to intent recreates theory loss at machine speed. The question is empirically grounded in the software-engineering literature: rationale (“why was it done this way?”) ranks among the hardest information needs developers report (Ko, DeLine, & Venolia, 2007), and intent/rationale questions dominate developers’ hardest-to-answer questions about code (LaToza & Myers, 2010). The classical responses — design-rationale capture from IBIS onward (Kunz & Rittel, 1970; Moran & Carroll, 1996), architecture decision records (Nygard, 2011), and issue-to-commit traceability recovery (Wu et al., 2011) — all require humans to *author or curate* the rationale record. Agentic construction changes the economics: the rationale record now writes itself as a side effect of doing the work, and the open problem moves from capture to retrieval — which is where this paper sits.

Agent memory systems. The established systems closest to this work, in increasing order of relevance. A-MEM (Xu et al., 2025) builds Zettelkasten-style linked memory notes, but its links are constructed at write time and retrieval remains one global cosine kNN — no expansion at query time. Zep/Graphiti (Rasmussen et al., 2025) maintains a bi-temporal entity/fact knowledge graph over conversations and business data with hybrid semantic+BM25+traversal search; it answers “what was true when,” over entities rather than code symbols. HippoRAG 2 (Gutiérrez et al., 2025) is the strongest graph-diffusion precedent — seed nodes then Personalized PageRank over an OpenIE triple graph — but diffuses over static QA corpora, not self-recorded agent history. MAGMA (2026) is the nearest neighbor: four orthogonal graphs including LLM-inferred *causal* edges, an intent-aware router that up-weights causal edges for “why” queries, and beam-search traversal. MAGMA’s causal graph is entailment over dialogue events (LoCoMo), not AST-anchored code artifacts; its traversal is beam search from rank-fused anchors rather than per-seed reinstatement; and, like every system above, nothing in it guards against the retrieval surface being flooded by echoes of its own recorded queries — the failure mode this paper measures and defends against. Deng (2026) contributes evaluation methodology rather

than a system (entity-collision stratification, pinned BM25 floors, bootstrap intervals) and is discussed under relevance feedback above; we adopt its discipline as the standard our queued ablation grid should meet.

Concurrent 2026 preprints. Three concurrent works occupy immediately adjacent space; the positioning below is based on close reading of the full texts. ACT-Up (Thomson & Lebiere, 2026) is not an agent-memory system but a cognitive-modeling toolkit: a working-memory and spreading-activation module for the ACT-Up architecture, combining base-level decay with spreading activation from working-memory context and validated against human serial-recall data (conditional response probability fits, $r^2 > 0.9$, on the Klein-Addis-Kahana paradigm). Its recall loop is genuinely iterative — “preliminary retrievals continue to be sources of activation, creating a robust chain of contextual information to prime subsequent recall” — which makes it mechanistic *grounding* for our reinstatement walk rather than a competing system: it validates the recall dynamics we borrow, on human data, with no agent, corpus, or code anywhere in scope. MRMS (Li & Shi-Nash, 2026) is architecturally the nearest neighbor: a multi-resolution memory substrate whose typed memory graph carries provenance-relevant edges (*supports*, *contradicts*, *supersedes*, *derived-from*) and whose retrieval pipeline follows kNN with typed graph expansion. It differs on four axes: interaction traces are undifferentiated (no intent/deliberation/artifact taxonomy), there is no code or artifact awareness, its evidence-attribution task is single-hop claim-to-trace selection rather than multi-hop “why” chains, and its evaluation is 800 deterministically generated synthetic tasks with, by its own description, “no private data, production data, external API, or LLM judge” — a complementary regime to the production-corpus evaluation pursued here. E-mem (Wang et al., 2026) preserves *uncompressed* episodic context — raw dialogue/document segments held by per-segment assistant agents — retrieved by multi-pathway activation with an iterative refine-and-query loop, and is the one system of the three with real multi-hop evaluation (LoCoMo, HotpotQA). But its stream is a single undifferentiated token sequence: no artifact or code trace, no human-intent typing, no code graph, and its multi-hop questions are conversational QA, not code provenance. Across all three: none jointly indexes intent+deliberation+artifact, none is code-graph-aware, and none evaluates provenance “why” questions on a production corpus. This paper’s contribution is therefore joint three-trace indexing, a specified and measured reinstatement algorithm, and a methodological finding about observer effects in self-indexing systems — not a claim of exclusive novelty in episodic agent memory writ large.

3 System Design

3.1 Architecture overview

CSR is a single Rust binary (*csr-engine*) that replaces an earlier Python/Docker/Qdrant stack. At evaluation time the corpus comprised over 70,000 indexed chunks across roughly 530 conversations, together with stored reflections, an AST-derived code graph, and episode anchors; exact per-snapshot chunk counts appear in the Method section, where they document contamination states. Embeddings are local (MiniLM, 384 dimensions via FastEmbed); search uses HNSW (*hnsw_rs*, less than 1 ms p95 for unconstrained

queries); persistence is SQLite (`rusqlite`). AST analysis via `ast-grep` populates `code_nodes` / `code_edges`, linking conversations to files and functions they touched. A `code_evolution` table records the session↔file timeline used both for graph spreading and for ground-truth construction in evaluation. Three layers are distinct here and are never collapsed in what follows: (i) the AST symbol tables `code_nodes` / `code_edges`; (ii) the session↔file co-edit ledger `code_evolution`; (iii) symbol→conversation attribution. Only the co-edit ledger is read by the retrieval walk’s graph-spread step (Step 3 below); the AST tables supply definition-versus-name-only labeling, line anchors, and visible abstention on the consumer surfaces; attribution is the layer a user consumes when asking who introduced a symbol, and is measured on its own terms rather than by retrieval metrics (Limitations, Deferred work). The system exposes fourteen MCP tools and six Claude Code hooks (`SessionStart`, `UserPromptSubmit`, `PostToolUse`, `Stop`, `PreCompact`, `SessionEnd`) that capture sessions continuously.

3.2 Saga schema (Phase 1 data model)

A *saga* is the joint structure over one unit of work (the term is unrelated to the distributed-transactions Saga pattern; we use it in the narrative sense — the connected story of how a piece of code came to be):

1. **Intent** — human prompts and human-authored dialogue that state goals and constraints.
2. **Deliberation** — agent reasoning, tool-call traces, dead ends, and sidechain/subagent chatter.
3. **Artifact** — resulting code changes, tracked file-by-file and session-by-session in the code graph.

Phase 1 production changes make saga structure explicit in storage:

- **chunks.seq** — import order within a conversation (aligned with the existing chunk index that feeds UUIDv5 chunk ids). Enables future chunk-level temporal reinstatement; not yet used in scoring.
- **chunks.is_sidechain** — true if any source message has the JSONL `isSidechain` flag *or* the conversation id starts with `agent-`. Over-labeling was deliberate: live Claude Code places subagent transcripts in separate `agent-*.jsonl` files, so inline `isSidechain` flags were effectively absent. The `agent-` prefix rule labeled 10,907 of 33,015 backfilled chunks. Labeling is present; sidechain-aware rerank weights are deferred.
- **retrieval_events for MCP** — MCP-path searches are logged with `hook_phase='mcp_search'` and sentinel `session_id='mcp'`. Previously only the prompt-submit hook path logged retrieval, rendering the dominant MCP recall path invisible to any future frequency/recency mechanism. Logging unblocks Phase 2 learned decay; it does not implement it.

Deferred non-goals (explicitly not claimed as done): learned decay weighting; chunk-level temporal reinstatement via `seq`; `SessionStart` saga narration UX; sidechain-aware scoring weights.

3.3 Reinstatement recall (*csr_why*)

The algorithm implements five steps. Defaults: $k = 10$ (max 50), $\text{seeds} = 3$, $\text{blend_query_weight}=0.65$, $\text{graph_boost}=1.10$, $\text{graph_cap_per_seed}=6$, $\text{min_score}=0.20$. No LLM participates in the retrieval path — only vector and graph arithmetic. Warm latency target is less than 100 ms; measured warm latency is 8 ms in production.

Step 1 — Seed retrieval. Compute top-3 nearest neighbors over the query against chunks and stored reflections merged by cosine similarity. This is exactly the current one-shot baseline used by `reflect_on_past` (minus presentation formatting).

Step 2 — Context blend (second hop). For each seed with vector s , form a blended query

$$q' = \text{renorm}(0.65q + 0.35s)$$

and re-run kNN with q' . This is the TCM-style reinstatement step: search from the context associated with the seed memory, not only from the literal cue.

Step 3 — Code-graph spreading activation. From each seed’s session, look up files touched in `code_evolution`, find other sessions that touched the same files, and pull the best-matching chunk (cosine vs. the *original* query) from each neighbor session, with a small activation boost ($\text{graph_boost}=1.10$). Cap at $\text{graph_cap_per_seed}=6$ to limit over-spreading.

Step 4 — Episode chain hop. From the seed session’s episode reflection, follow one step of `prev_episode_id` (chronological chain of session summaries) and pull the immediately prior episode when present.

Step 5 — Fuse, rerank, dedupe, cap. Merge channels, deduplicate, apply min_score , rerank the fused pool under a provenance-aware policy (Phase 1.5, below), and return at most k results as grouped evidence chains citing conversation ids.

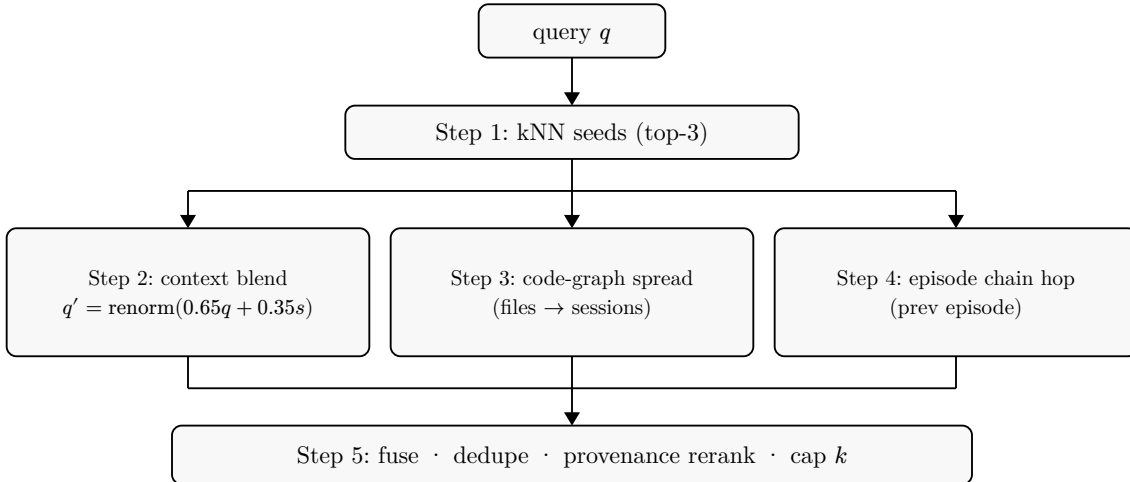


Figure 1: Reinstatement recall. One-shot kNN provides seeds only; three reinstatement channels re-expand from seed context before provenance-aware fusion.

3.4 Provenance-aware rerank (Phase 1.5)

The fused pool is wide (`min_score=0.20`, ~46 candidates in practice), so rank order inside it matters. Phase 1.5 applies CSR’s existing rerank machinery to the pool under a dedicated Provenance policy, tuned by frozen-corpus evidence rather than by porting `reflect_on_past`’s weights wholesale. Two planned transfers were *rejected* by per-query evidence: demoting tool-mechanic chunks evicted a ground-truth session (edit-heavy chunks are provenance *evidence*, not noise, in this task), and a flat user-authority boost promoted weakly relevant user chunks and user-role compaction summaries over strong evidence. Two defenses were added instead: (1) a **query-echo penalty** (-0.35) on chunks quoting the query near-verbatim — a session that *asks* a question is not the session that *answered* it — paired with echo-aware seed selection ($2\times$ seed over-fetch, non-echo hits preferred as walk seeds), and (2) classifying compaction summaries (“This session is being continued from a previous conversation...”) as scaffold text. Without echo-aware seeding, a re-asked question drowns the walk in its own prior askings: in a live test, the reinstatement pool for a repeated query collapsed to three self-referential conversations; with it, five conversations surfaced with the answer-bearing chunk ranked first.

Implementation note on scoring. Per-conversation scoring uses exact cosine over that conversation’s stored chunk embeddings (conversations average ~136 chunks — microseconds), *not* a filtered HNSW search. Tiny allowed-id-set filters pathologically escalate `hnsw_rs` toward near-full-index scans; exact scan over a single conversation is both simpler and faster.

3.5 Baseline for comparison

Arm A (baseline) is one-shot kNN over chunks+reflections — the production `reflect_on_past` retrieval core. Arm B is the five-step walk above. Both share the same embedding space and storage. A fairness caveat we state plainly: after Phase 1.5, Arm B also carries the provenance reranker and echo-aware seeding while Arm A does not, so post-1.5 comparisons (including the judge panels) validate the full `csr_why` stack rather than the walk in isolation; only the Phase 0 spike isolates the walk. A factored comparison grid — each arm sharing the identical reranker, dedup, and budget, varying only candidate generation — has since been run in reduced form: five arms isolating the expansion channels (no expansion, co-edit spread, AST spread, degree-preserving AST sham, both spreads), with paired-bootstrap intervals, reported below under Evaluation §Sham-controlled channel follow-up and released at `csr-engine/eval-kit/h1/`. The echo-demotion and centroid-PRF arms of the original specification remain unrun.

3.6 Evidence-grounded forgetting (“dreaming”)

The asserted-state ledger established one boundary: retrieval may replay a cited verdict, but it should not infer state from plausible prose. Dreaming extends that boundary from human- or agent-asserted settlement to repository-verifiable evidence. On a background cadence it asks a narrower question: does the code span a remembered claim was bound to still exist in the evidenced form? The name describes background maintenance, not generative replay — no model writes or judges a dreaming verdict.

Witness ledger. `witness_ledger` is the durable substrate. At a commit OID it records the code anchor, a versioned `b3`: BLAKE3 stamp of the anchored span, and the identity needed to join the witness back to a claim or chunk. The ledger is append-only: later observations do not rewrite earlier witnesses; obsolescence and reinstatement are additional facts, so an exact revert restores a prior witness without erasing the interval in which it was absent. The unit is the span rather than the file — the v10 spike found file-level stamps report false drift for unchanged functions whenever neighbouring code changes. Commit OIDs order committed evidence; wall-clock time does not. Dirty-worktree observations are provisional and never become committed ledger facts. Seven repositories have been ledgered; on the integrity gates used for this work, witness closure is 98.1%, internal binding 86.2%, and the drifted count 0. These are substrate measurements, not retrieval-effect estimates.

Deterministic verdicts. Verdicts are computed by witness closure and hash-equality joins, abstention-first: (1) resolve the claim to one ledger identity, else abstain; (2) establish the committed witness’s relation to the evaluated commit by ancestry or content identity, else abstain; (3) span-hash equality in the same hash domain yields `Intact`; (4) absence yields `Vanished` only when the extractor can establish absence without a name collision; a changed span yields `Drifted`, a later evidenced replacement `Superseded{by, receipt}`; (5) a later hash-equal witness may reinstate an earlier belief, while ambiguous renames and $A \rightarrow B \rightarrow A'$ histories remain abstentions. No LLM anywhere in the verdict path, and no semantic claim that changed code is *wrong* — only that an earlier code-bound statement no longer has the same witness.

Consumption: demote and annotate. **Demote** partitions evidence before pagination, so a proved-obsolete item falls behind intact or unclassified evidence — it does not delete the item, stack a second decay penalty, or remove the receipt. **Annotate** leaves the evidence visible and attaches its verdict, witness identity, and commit receipt; live search renders these as `[evolved]/[stale anchor]` annotations carrying the supporting commit. The channels separate ranking policy from the evidential record: demotion can be disabled while annotations remain inspectable. The daemon revisits eligible work on a 6-hour cadence with fair (not newest-first) queue selection — an accumulating import backlog must not starve older witnesses, a failure already observed in the ratification pipeline — and a kill switch disables dreaming without disabling search.

A second evidence-grounded consumer, temporal decay v2, replaces wall-clock age with release ancestry where linkage exists: a conversation whose linked commits shipped several releases ago decays faster than an equally old conversation about still-current code (effective half-life scaled by releases-behind, capped, neutral on any ambiguity — fan-out, multiple roots, missing repositories, or unreleased work all fall back to the wall clock).

4 Evaluation

4.1 Method

We evaluate *provenance recall*: multi-hop questions of the form “why does the code look like this?”, each paired with a ground-truth (GT) target file. GT is the set of sessions in `code_evolution` that historically touched that file. Primary metric **M1** is summed GT-session

coverage across queries at equal budget $k = 10$. **M2** is distinct-conversation diversity in the top-10 (to check that gains are not mere result spraying). **M3** is qualitative side-by-side judging (better / tie / worse per query) plus a junk-rate estimate for B-only results; in the spike it was single-annotator, and it was later re-run as a blind three-judge cross-vendor panel with inter-rater agreement reported (see Evaluation §Blind cross-vendor judging).

Pre-registered gates (set before running):

Table 1: Pre-registered gates.

Gate	Criterion
PROCEED	$B \geq A + 25\%$ summed GT coverage and B better on $\geq 50\%$ of queries and junk rate under 30%
ITERATE	Mixed results; allow one weight retune and rerun
KILL	$B \leq A$; write up honestly, thesis dies cheaply

Disclosed circularity. Arm B’s graph-spread step queries the same `code_evolution` table used to define GT, but only via seed sessions found by search: a bad seed yields no graph gain. The design is therefore not circular by construction, but shared plumbing could still inflate graph-channel hits. We therefore require mechanism attribution and M3, not M1 alone.

Attempted independent GT from version control. To break the shared-plumbing dependency entirely, we attempted a second GT construction from VCS metadata: commits in the project repository carry `Claude-Session` trailers naming the web session that produced them, giving a transcript-independent session-to-file mapping. The attempt failed instructively, in two ways. First, coverage: trailers exist on only 60 of 1,096 commits, all within days of writing — months-old decisions targeted by the eval queries predate the convention entirely. Second, and more telling, the bridge from trailer IDs to conversation IDs (locating the trailer text inside indexed transcripts) was itself contaminated by the observer effect this paper documents: the analysis session that ran `git log` to extract trailers thereby imported those trailer IDs into its own transcript, producing spurious mappings to the measuring session. We therefore report no independent-GT agreement number; the honest yield is a protocol recommendation — projects intending to evaluate provenance recall should emit session-to-commit trailers from day one, and should extract them outside the indexed environment.

Corpus. Live CSR development history (dogfooding). Phase 0 spike: morning of 2026-07-15, throwaway read-only binary against 71,762-chunk DB, 12 hand-written queries. Phase 1 production: evening of the same day, `csr-engine eval --provenance`, the same 12-query suite lifted verbatim into the engine, after a full day of development sessions had been imported. The snapshot states used throughout:

Table 2: Evaluation snapshot states.

Snapshot	Chunks	Queries	Contamination state	Index build
Live, morning (spike)	71,762	12 (primary)	pre-eval-dialogue (cleanest)	fresh per run
Live, evening (production)	~73,300	12 (primary)	same-day eval dialogue imported	fresh per run
Frozen eval-frozen-2026-07-15	73,342	12 + 8 (second corpus)	partial (includes spike-discussion session)	one shared build per A/B comparison

Statistical treatment. With 12 and 8 queries, we report per-query outcomes and exact paired sign tests on wins/losses (ties excluded) rather than interval estimates: primary suite 7 wins / 0 losses (two-sided $p \approx 0.016$); second corpus 4/0 ($p = 0.125$, not individually significant); pooled 11/0 ($p \approx 0.001$). Summed coverage weights queries with larger GT sets more heavily; per-query win/tie/loss counts are the macro-level check. Bootstrap confidence intervals, macro-averaged normalized recall, and graded origin-level metrics (origin-MRR, nDCG) require the graded relabeling described under limitations and are queued with the ablation grid.

4.2 Phase 0 spike results

Gate: PASSED.

Table 3: Phase 0 spike results.

Metric	Arm A (kNN)	Arm B (reinstatement)
Summed GT coverage ($k = 10$)	15	23
Relative lift	—	+ 53% (gate +25%)
Per-query outcomes	—	B better 7/12, tie 5/12, worse 0/12
Diversity (distinct conv. in top-10, summed)	75	77

Mechanism attribution of B’s GT-hit result lines:

Table 4: Mechanism attribution.

Channel	GT hits
Context blend	24
Graph spread	9
Episode chain	0

The majority of the win is from the pure-semantics blend hop, not the potentially shared-plumbing graph walk — the key quantitative evidence against circularity as the sole explanation of M1.

Qualitative highlight (Q3: “why is the integrity check cached”). Arm A’s entire top-10 was subagent/tool-mechanic chunks with **zero** GT sessions represented. Arm B’s blend hop surfaced the human-origin conversation (7ecb720) where caching was decided and discussed with the user — the failure mode the thesis predicts: one-shot similarity drowns in deliberation chatter while reinstatement digs back toward intent. B-only junk rate was eyeballed under 20%, mostly looser-but-relevant graph pulls rather than noise.

4.3 Phase 1 production results

The same walk was reimplemented as engine code and registered as the 14th MCP tool (`csr_why`). Definition-of-Done checks passed: migrations idempotent; backfill processed 732 candidate files (182 missing/skipped, reported not silently dropped); 8 ms warm latency against a 100 ms budget; well-formed evidence chains.

Table 5: Production runs.

Run	A	B	Notes
Production eval --provenance (12 queries)	15	18	First production path
Immediate rerun via original spike code path	15	19	Faithfulness check

Production reimplementation is faithful to the spike algorithm (18–19 vs. the spike’s 23 on the same 12-query suite under a dirtier corpus), not a regression to a weaker walk.

4.4 Using the discrepancy as evidence

Spike B=23 (clean, pre-development-session) versus production B=18/19 (post same-day import) is not treated as measurement noise to hide. It is primary evidence for the observer-effect finding developed in Discussion: between morning and evening, transcripts of the development session that *discussed and quoted the eval queries and spike output* were imported by always-on hooks. Self-referential chunks textually outrank true origin conversations for the queries that discuss them. Concretely, the acceptance query about integrity-check caching no longer directly surfaces 7ecb720 as it did in the morning spike.

A frozen snapshot (`eval-frozen-2026-07-15.db`, 4.2 GB, 73,342 chunks) was taken, but even that snapshot includes same-day contamination from the spike-discussion session. Morning pre-development numbers remain the cleaner baseline; future protocol should freeze from a session-zero backup before any evaluation dialogue begins. The necessity of freezing was confirmed directly: across four otherwise-identical live-corpus eval runs during Phase 1.5 development, the chunk count drifted from 73,846 to 73,882 as the session’s own transcripts imported, and per-query numbers changed between invocations.

4.5 Phase 1.5 frozen-corpus results

Both binaries (pre- and post-rerank) were evaluated against clones of the frozen snapshot. Aggregate coverage held — A=15, B=18 before and after — while the distribution moved where the thesis predicts: Q3, the acceptance-failure query (“why is the integrity check cached”) whose ground-truth session had been displaced by contamination, recovered from 0 to 1 GT sessions; Q12 improved 1 $\rightarrow$ 2; Q5 and Q10 each ceded one hit (2 $\rightarrow$ 1, 4 $\rightarrow$ 3). The rerank is therefore best read as a *contamination repair* mechanism, not a coverage amplifier: it restores origin conversations on the queries where self-echo displaced them, at the cost of marginal hits elsewhere.

Per-query results for the primary suite on the frozen snapshot (a later independent index build of the same corpus, illustrating both the distribution and the documented ± 1 ANN build variance — A=14, B=17 on this build vs. 15/18 above):

Table 6: Per-query GT-session coverage, primary suite, frozen snapshot (independent index build).

Query	GT size	A	B	Query	GT size	A	B
Q1	6	2	2	Q7	3	1	3
Q2	2	2	2	Q8	0	0	0
Q3	6	0	1	Q9	0	0	0
Q4	0	0	0	Q10	6	3	3
Q5	3	2	1	Q11	1	1	1
Q6	3	2	3	Q12	2	1	1

Three queries (Q4, Q8, Q9) have zero reachable GT on the frozen snapshot (their `code_evolution` rows post-date it or the target has no file mapping), and B’s one per-query loss here (Q5) is the echo-defense trade documented above. The distribution is the honest shape of the result: B’s advantage concentrates in a minority of queries where multi-hop reach matters (Q3, Q6, Q7), with the rest tied — consistent with a mechanism that rescues hard cases rather than uniformly inflating scores.

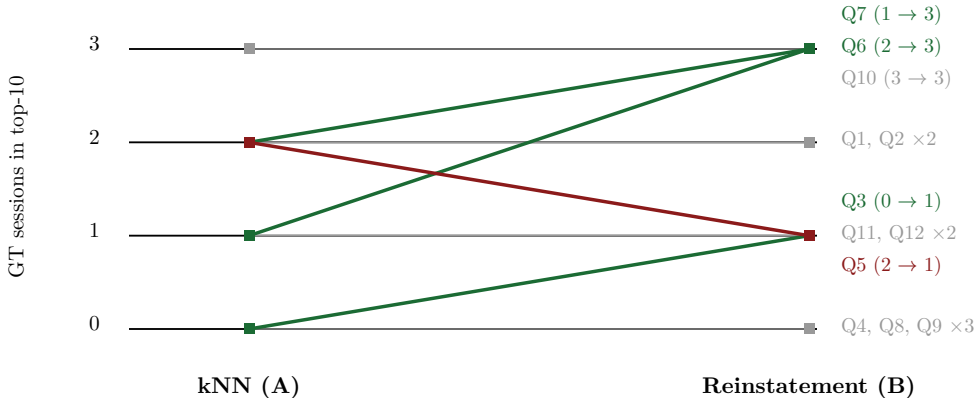


Figure 2: Per-query ground-truth coverage, one-shot kNN (A) vs. reinstatement (B), 12 primary-corpus queries on the frozen snapshot with one shared index build. Rising lines are gains (Q3, Q6, Q7), the falling line is the single regression (Q5, one GT session displaced within budget), flat lines are ties; coincident lines are annotated with their multiplicity. Aggregate: A=14, B=17.

A known remaining gap: for decisions predating the current corpus era (e.g., a major stack replacement), asker sessions still win the top group over the true origin conversation. Chunk-level echo demotion is insufficient when *entire conversations* consist of re-askings; conversation-level echo exclusion and population of a *supersedes* relation are the queued responses.

4.6 Second corpus: within-operator cross-project replication

The corpus is not CSR-only: the same engine continuously indexes every Claude Code project on the machine — 15 projects, 562 conversations, ~74,700 chunks at the time of writing — of which CSR’s own development history is a minority share. The largest corpus family is an unrelated product line (a consumer mobile app, its marketing-campaign operations, and an analytics command center: TypeScript/React/Expo, Meta and PostHog APIs, video production), contributing $3.4\times$ more code-graph rows than CSR itself (613 vs. 180 `code_evolution` rows; 78 vs. 20 sessions; 229 vs. 41 files).

We wrote 8 new provenance queries against this second corpus — authentication-migration decisions, caching-architecture choices, analytics discrepancies, video-composition structure — using the same GT construction (sessions in `code_evolution` touching the target file) and the same frozen snapshot, probed through the production MCP interface with cross-project scope. Coverage: **A=15, B=22 (+47%)**, with B better on 4 queries, tied on 4 (one query missed GT on both arms), and worse on none. The lift on a TypeScript/marketing-operations corpus (+47%) closely replicates the lift on the Rust/systems corpus (+53%), evidence that reinstatement recall’s advantage is not an artifact of CSR describing itself.

The blind three-judge cross-vendor panel (same protocol as below) was run on the second corpus as well: reinstatement judged better on 5 of 8 queries, tied on 3, **and worse on none** — **no judge cast a single vote for the baseline on any query**, with 4 of 8 verdicts unanimous. Fleiss κ is low here (0.19) for a well-understood reason: with zero

baseline votes, the only disagreement dimension left is whether a reinstatement win counts as a tie, and κ collapses under such extreme category prevalence (observed agreement 0.67). Judge rationales again converged on origin-surfacing: the baseline drifted into a neighboring project on the video-composition query while reinstatement stayed on the correct project and surfaced the sync-bug root cause.

Two disclosures: query authors knew the GT target files (as in the primary suite), and the corpus — while a different codebase, language, and work domain — is still one operator’s history, so multi-operator generalization remains open.

4.7 Blind cross-vendor judging (M3, multi-rater)

To replace the spike’s single-annotator M3, we re-ran side-by-side judging with three independent LLM judges from three model families (Claude Sonnet, Grok 4.5, Gemini 3.1 Pro), each invoked headlessly with an identical packet. Evidence was generated fresh for the purpose: a clone of the frozen snapshot served by a single engine process (one index build, so both arms share it), probed over the production MCP interface — Arm A via `csr_reflect_on_past`, Arm B via `csr_why`, both at $k = 10$, all 12 queries. Judges saw the two outputs per query as anonymized sides X and Y with side order randomized per query, were told to judge content rather than formatting, and were asked which side better surfaces the *origin* of the decision, penalizing re-askings and eval-harness echoes.

Majority verdicts: Arm B better on 8 of 12 queries, Arm A better on 1 (Q9, hooks catch-all policy, where the one-shot arm ranked the origin policy statement higher), ties on 3, no three-way splits. Seven of the twelve verdicts were unanimous across all three judges, in both side orders. Inter-rater agreement was moderate: Fleiss $\kappa = 0.51$ (observed agreement 0.78 against 0.55 expected by chance). Judge rationales independently converged on the paper’s central mechanism — the modal comment against Arm A was that its top ranks were occupied by the evaluation queries themselves (self-echo), while Arm B surfaced implementing sessions and root-cause dialogue. Caveats: the judges are LLMs, not humans; the judging instructions were written by the system’s authors and explicitly instruct judges to penalize re-askings and evaluation echoes — so the panel is blinded to *side* but not to the paper’s preferred failure model (a second panel under a neutral “which side better answers why” prompt, plus a small human-maintainer panel, is the queued remedy); and the packet truncated each side to its first ~4,500 characters.

5 Phase 2 experiments: graded gold, channel ablation, controlled contamination

The three experiments queued in earlier drafts as the next revision’s core have now been executed. Together they (i) replace the file-touch GT proxy with graded provenance gold, (ii) attribute the coverage effect to its mechanism, and (iii) convert self-indexing evaluation contamination from a documented incident into a dose-response phenomenon with a quantified repair.

5.1 Graded provenance gold from ratification behavior (E2)

Gold construction. Rather than retrospective annotation, grades were derived from the operator’s own contemporaneous dialog-acts and ratification behavior, under a protocol frozen before any retrieval ranks were seen. The operator first answered a sealed, pure-memory elicitation for all 20 queries — committed to version control *before* any cueing or rank exposure — yielding strata: 12 origin descriptions recallable from memory, 7 unresolved (excluded from origin-ranked metrics, never soft-matched), and 1 out-of-corpus. Descriptions were then mapped to conversation IDs by metadata only (literal text search over pre-freeze chunks, git dates, external ship events — no embedding retrieval, which would reintroduce circularity). Candidate pools (both arms’ frozen top-10s $\cup$ date-filtered file-touch history; 211 items, 205 reconstructible) were labeled by two independent LLM extractors from different vendors (Grok 4.5 and Claude Sonnet) applying a quote-anchored extractive protocol for DIRECTS / ACCEPTS / REJECTS / RE-ASKS acts; grading used strict two-vendor consensus, conservative on splits (directs agreement 84.8%, Cohen κ = 0.41; splits routed to owner audit). Grade 3 was reserved for sealed-and-mapped origins — extraction can corroborate but never mint an origin. External ratification ledgers (git history across four repositories, npm publish timestamps, a release-train manifest) provide off-corpus acceptance evidence: across 204 extracted conversation-items the operator’s dialogue contained 21 DIRECTS but only **1 explicit ACCEPTS** — operators direct in words and ratify by shipping, so independently timestamped ship events are the only acceptance signal that exists at scale.

Results (12 mapped-origin queries). Reinstatement beats kNN on every graded metric: origin-MRR 0.264 vs. 0.193 (+37%), nDCG@10 0.470 vs. 0.363 (+29%), Recall@10 of grade- ≥ 2 items 0.553 vs. 0.444 (+25%) — direction consistent with the coverage results, now on decision-graded gold. The sobering companion finding: **5 of 12 mapped origins were retrieved by neither arm**, and origin-MRR is zero for both arms on 7 of 12 queries. An owner audit confirmed all five contested mappings correct (map-error reading eliminated); four of the five missed origins date from the corpus’s earliest era — old, sparse origin conversations losing to later, denser sessions. Origin-finding is improved by reinstatement but not solved; the headroom above the winning system is large.

5.2 Channel ablation (E1)

A seven-arm ablation (research harness mirroring the production walk; one process, one shared index build over 73,342 chunks; scored against the E2 gold; conversations outside the graded pool scored zero, a bias *against* exploratory channels) decomposes the walk:

Table 7: Channel ablation on graded gold, one shared index build. Within-grid comparisons only; absolute values are not comparable to the E2 table (different index build).

Arm	origin-MRR	nDCG@10	R _{≥2} @10
kNN baseline	0.243	0.394	0.469
full reinstatement	0.276	0.477	0.559
blend-only	0.269	0.446	0.499
graph-only	0.329	0.555	0.607
episode-only	0.321	0.457	0.470
full — rerank	0.257	0.447	0.563
full — echo-defense	0.234	0.434	0.549

(In-text bridge for readers cross-checking tables: the kNN baseline row here, 0.243 origin-MRR, differs from the 0.193 reported in the E2 comparison because the grid ran on a separate index build; ANN construction is non-deterministic across rebuilds, so only within-grid comparisons are meaningful — the same discipline the contamination section motivates.)

Five findings. (1) *Graph spread is the workhorse*: graph-only beats the full walk on every metric despite the scoring bias against it — co-edit structure, not semantic blending, carries the provenance signal on graded gold. This reverses the spike-era attribution, and we treat the graded result as authoritative: blend dominance was an artifact of file-touch *coverage* gold; on decision-graded gold — which derives from dialog-acts and ship ledgers, independent of the `code_evolution` plumbing the graph channel shares with the coverage GT — graph dominance is not self-confirming. The arm ordering is nonetheless unbracketed: the sham-controlled follow-up below measures the co-edit channel’s marginal contribution over the otherwise-full stack at +0.0167 origin-MRR with a 95% interval whose lower bound is exactly zero, so finding (1) is a within-grid ranking and not a separated causal contribution. (2) *Fusion dilutes*: max-score fusion lets blend-sourced semantic neighbors crowd graph-sourced provenance neighbors out of the final budget; the tuning direction is a larger graph share, not more channels. (3) *Echo defense is causally necessary*: removing it drops origin-MRR to 0.234 — below the kNN baseline. (4) *The episode chain, previously a null result, is a strong origin-finder on graded gold* (0.321 origin-MRR) though weak on evidence depth — it finds the origin thread, not the full evidence set. (5) *The structural AST graph is not the channel that ranks*: the “graph” arm above is the co-edit ledger, and a follow-up that separates the two graph layers finds the AST-expansion arm identical to the base arm on origin-MRR for all 12 mapped-origin queries and indistinguishable from its own degree-preserving sham; the AST subsystem’s measured contribution is provenance integrity, not provenance ranking (next subsection).

5.3 Sham-controlled channel follow-up (E1b)

The E1 grid reports point estimates, and its graph arm conflates the two graph layers the system actually has. A follow-up re-ran the expansion channels alone against the same sealed E2 gold: one process, one deterministic index build over a frozen evaluation unit, exact vector scan, identical seeds, rerank policy, quotas, and budget across arms, with the

expansion channel as the only variable. Five arms: S (no expansion), S+F (session↔file co-edit spread — the channel E1 calls graph spread), S+A (AST symbol-graph spread over resolved `code_edges` only), S+A-sham, and S+F+A. The sham arm is the load-bearing control: it permutes edge destinations under a fixed-seed xorshift64 shuffle that preserves every node’s degree, retaining the AST graph’s density and shape while destroying its structural correspondence. An arm that cannot beat its own sham has not shown that structure, as opposed to edge count, is doing the work. Intervals are 10,000-resample paired bootstraps over per-query deltas (seed 20260731); origin-MRR is defined only on the 12 queries carrying a sealed mapped origin, `nDCG@10` over all 20. A methods note on determinism: after these numbers were sealed, a second-corpus replication exposed hash-map iteration order in the harness’s candidate pools as a per-process tie-break; the harness was made bit-deterministic (total order: score descending, id ascending), and re-measurement showed every origin-MRR contrast above bit-stable, `nDCG@10` drift $\leq +0.0044$, and no verdict change — the sealed numbers are kept as the record of the frozen run. The kit is released at `csr-engine/eval-kit/h1/`; the frozen unit’s sha256 is

a43fada5cd5f2954563707f75676521847005d9461f876b80e7082a8ae66f114

Table 8: Sham-controlled expansion-channel contrasts on the sealed E2 gold, one shared index build, 10,000-resample paired bootstrap (seed 20260731). No contrast excludes zero.

Contrast	Δ origin-MRR (12 mapped)	Δ <code>nDCG@10</code> (20 queries)
S+F — S	+0.0167, CI [0.0000, +0.0500]	+0.0264, CI includes 0
S+A — S	0.0000, identical on 12/12	CI includes 0
S+A — S+A-sham	≈ 0	+0.0002, CI [−0.0132, +0.0139]

Neither graph channel separates from zero. The co-edit channel is directionally positive on both metrics and its origin-MRR interval touches zero exactly at the lower bound — consistent with E1’s ordering, insufficient to certify it. The AST channel is a strict null: it returns the base arm’s origin-MRR on every mapped query, and its `nDCG@10` difference from a graph with identical degrees and scrambled structure is +0.0002. The combined S+F+A arm is reported in the released kit and adds no separable gain over S+F.

Two bounds constrain how far the AST null generalizes, and we state both rather than banking the negative result. First, *sparsity*: the AST channel fired on 8 of 20 queries at all, over 4,092 resolved edges reaching 122 distinct conversation keys; a channel silent on most of the suite cannot be strongly falsified by it. Second, and more consequential, *attribution quality*: the arm maps AST symbols to conversations through the stored `first_conv_id` field, which a same-session audit measured at 50.7% agreement (1135 of 2237) with evidence-bearing symbol-level attribution derived from `code_evolution` change events, against a pre-registered bar of 90% — and 499 of 532 files carry a single uniform id, i.e. the field is a file-level projection rather than per-symbol evidence. Seeds routed through a half-accurate map cannot demonstrate structural signal even where structural signal exists. The honest reading is therefore corpus- and attribution-bounded: on this corpus, with this attribution, AST structure contributes no detectable ranking signal. The rematch — the same sealed

kit with the seed mapping switched to evidence-bearing attribution — is queued, and is stated as a queued experiment rather than as a defense of the null.

Relocating rather than discarding the claim: what the AST layer is measured to deliver on this corpus is *integrity*. Every `calls` / `imports` edge terminates in an evidence-bearing state (witness closure 98.1% against a pre-registered bar of 90%); internal binding is 86.4% against a bar of 70%, on a denominator that excludes only edges with recorded proof of being external, method-dispatched, internal-module, stale, drifted, or local; zero edges are drifted; and consumer surfaces label definition-backed versus name-only matches, carry line anchors, and abstain visibly rather than guess. Those are gate-measured properties, not MRR, and they are what makes a provenance answer trustworthy rather than merely ranked.

5.4 Controlled contamination (E3)

From a session-zero snapshot C0 (17,134 chunks, pre-dating all evaluation dialogue) we built C1 = C0 + only the evaluation-design transcript, C-sham = C0 + an unrelated transcript matched for size (107% of C1’s bytes, zero query-text occurrences), and C5 = C0 + five scripted re-ask cycles (verbatim query re-askings with paraphrased answers, explicitly marked synthetic — a controlled self-referential injection, not a natural-ecology observation). Retrieval used exact brute-force scan, eliminating ANN variance so the independent variable is corpus content alone; 8 queries whose owner-audited origins pre-date C0; three arms.

Table 9: Contamination dose-response and repair, exact-scan retrieval, mean over 8 eligible queries.

Arm under C5 (dosed re-asking)	echo@10	origin-MRR
kNN	4.9 / 10	0.014
walk — echo-defense	4.9 / 10	0.014
walk with echo-defense	1.4 / 10	0.047

Five scripted re-ask cycles capture half of naive retrieval’s top-10 (on one query the entire top-3 becomes the synthetic cycles); the undefended walk is equally captured, since hop-2 spreads outward from echo seeds. The echo defense is a quantified repair: −72% echo occupancy, 3.4× the origin-MRR under dose, and on one query the origin *enters* the top-10 only in the defended arm — demoting echoes clears ranked space for provenance. The sham control is null (rankings byte-identical to C0), so displacement is content-specific, not a corpus-size artifact; and C1 shows the evaluation-design conversation itself entering a query’s top-3 — storing a single evaluation session measurably perturbs the system under evaluation. Disclosure: under exact scan on the session-zero corpus, origins sit outside the top-10 for 6–7 of 8 queries in *all* conditions (the origin-finding floor from E2 again); the dose-response and repair rows are the load-bearing results, origin-rank trajectories are floor-limited.

5.5 A pre-registered negative result: global ratification weighting

The E2 gold construction suggested a natural extension: if ratification behavior can grade relevance, perhaps it can weight retention. We tested whether a *global* per-conversation

ratification score — DIRECTS / ACCEPTS / REJECTS dialog-acts extracted by an LLM over each conversation, corroborated against the external git ledger, with uncorroborated scores capped at 0.6 — ranks decision origins. The pre-registered gate was Spearman correlation between backfilled scores and the sealed E2 grades (conversation-level, taking the max grade across queries), with an explicit halt condition at $\rho \approx 0$. The gate failed three times, across three extractor generations. Run 1 measured $\rho = 0.060$ but was an extraction artifact: 88% of conversations yielded zero acts (the digest sampled mostly assistant and tool text, and the echo-hardened prompt bailed to empty output on instruction-like digests — which all agent digests are). Run 2, under a rebalanced prompt, extracted acts with genuine variance and still measured $\rho = 0.071$ ($n = 123$); grade-3 sealed origins averaged a score of 0.362 against 0.422 for grade-0 incidental conversations. Run 2 was believed at the time to also use an operator-turn-prioritized digest; post-merge external review exposed that the digest builder never received author labels in production (a storage-layer join omission), so v2 had silently fallen back to head/tail sampling. After fixing the join and re-running the gate with operator-turn prioritization genuinely active, run 3 measured $\rho = 0.036$ ($n = 111$; grade-3 mean 0.447 vs. grade-0 mean 0.476) — the mechanism correction made the correlation flatter, not better.

The mechanism is a construct mismatch, not noise. The ratification score is *global* — this conversation’s work, any work, was directed and accepted — while E2 relevance is *query-conditional* — load-bearing for *this* decision. In a high-ship-rate solo corpus nearly every working session directs, accepts, and ships something, so node-level act-strength cannot separate “origin of this decision” from “productive session about something else.” The behavioral benchmark planned downstream of the gate was not run: benchmarking retrieval on a signal measured as flat would credit policy, not memory. E2’s gold worked precisely because it bound acts to a query’s artifact; the retention-weight version of the thesis dropped that binding and halted at its gate. What survives is an untested *edge-level* formulation: ratification acts bound to (conversation, artifact) pairs, weighting reinstatement graph edges and joined query-conditionally — a redesign, not a tweak, and explicitly open. A post-halt data audit ruled out the zero-cost version of that redesign: the stored extraction rows persist commit identifiers but discarded the file identity that selected them, so strict act-to-artifact binding covers zero sealed-gold pairs on existing data — any future edge-level attempt must persist the binding at extraction time.

5.6 A blinded behavioral benchmark: re-litigation under three memory arms

The gate halted the retention-weighting program but left the behavioral question it was meant to serve — does memory stop an agent from *re-litigating* settled decisions? — decoupled and testable. We ran it as a three-arm benchmark under externally advised binding conditions: a similarity-only control arm was required (a no-memory control alone would only show “context helps”), the majority of tasks had to postdate the retrieval tuning they probe (avoiding train-on-test circularity), and grading had to be human and blinded.

Design. Sixteen tasks were sealed (SHA-256) before any arm ran: realistic teammate requests against the corpus’s own settled history, each with a pre-registered correct action across four classes — *settled-for-cause* (correct action: decline, citing the true reason; 4),

should-change (a recorded decision’s own exit condition is now met; correct action: accept; 4), *irrelevant-past* (superficially similar history that is not applicable precedent; correct action: judge on merits; 2), *ambiguous* (correct action: check with the owner; 2), plus four calibration tasks. Three arms answered every task under an identical prompt and model, differing only in retrieved context: production reinstatement recall (R), similarity-only kNN (K), and no memory (N). Arms ran against a frozen snapshot scrubbed of the design session and its subagent traces (the E3 discipline applied prospectively); retrieval outputs were verified free of scrubbed identifiers; a single shared renderer produced arm-invariant context with no scores or arm-revealing wording. The 16 tasks $\times$ 3 arms produced 48 responses; the 36 responses to the 12 headline tasks were graded on a 0–3 reason-faithful-action rubric, and the four calibration tasks were scored at action level only. Grading was performed by the corpus owner (the sole author) with arm identity hidden behind per-task shuffled codes and the key sealed until all grades were recorded — blind-to-arm, but not independent of the author; we label it *author-graded*, *blinded* rather than independent human evaluation.

Results. At action level the memory arms each took the pre-registered correct action on 12/16 tasks against 8/16 for no memory — and R and K produced *identical* verdicts on all sixteen. On blinded reason quality (12 headline tasks, max 36): R 27, K 27, N 24. Per-task paired grades:

Table 10: Per-task blinded rubric grades (0–3), unblinded post hoc. R = reinstatement, K = kNN, N = no memory. Paired differences are sparse: R and K differ on two tasks only (T01, T11, net zero); N trails on three settled-for-cause tasks (T01–T03) and nowhere else.

Task	Class	R	K	N
T01	settled-for-cause	3	2	1
T02	settled-for-cause	1	1	1
T03	settled-for-cause	3	3	1
T04	settled-for-cause	3	3	3
T05	should-change	1	1	1
T06	should-change	1	1	1
T07	should-change	3	3	3
T08	should-change	1	1	1
T09	irrelevant-past	3	3	3
T10	irrelevant-past	3	3	3
T11	ambiguous	2	3	3
T12	ambiguous	3	3	3

The aggregate conceals the structure: on *settled-for-cause* tasks the memory arms score 10/12 and 9/12 against 6/12 for no memory — and the no-memory failures are the dangerous kind (on a calibration task it accepts re-introducing a shipped performance regression that both memory arms decline, citing the original incident). On *should-change* tasks all three arms score an identical 6/12: in 10 of 12 responses the arms deflected justified changes to “needs more information” even when the retrieved context contained

the original decision’s own exit condition — a uniform, arm-independent conservatism. Memory shifts the failure mode from *re-breaking settled decisions* toward *over-deferring on justified change*; task mixes without a should-change class would miss this entirely. On *irrelevant-past* tasks every arm scored perfectly — including on a task proposing the edge-level follow-up immediately after the node-level halt: no arm over-generalized the negative result into a blocker.

Reading. The benchmark gives the paper’s central claim its behavioral cash value — *conversation memory* (not reinstatement specifically) roughly halves reason-quality failures on the settled-for-cause class, exactly where Naur’s theory-loss predicts, and nowhere else — while delivering a null that scopes the invention: reinstatement earned no separation from one-shot kNN, at either level, on these single-hop tasks (“does history bear on this request?”). The +53% spike advantage was measured on multi-hop provenance chains; the tie localizes reinstatement’s value to that regime, and a multi-hop behavioral benchmark is the experiment that would test it.

5.7 Staleness benchmark: dreaming vs. deterministic baselines (T4/H1)

The dreaming mechanism’s staleness rule was evaluated on labels the repository gives away for free: symbol time-travel across release tags. Between v8.0.0 and v9.5.0, 13 of 17 tags are sampled deterministically (integer linspace, endpoints included) and every (file, symbol) witnessed at an intermediate tag is a “belief held then.” The prediction rule is the online one — superseded if any later sampled tag holds a different stamp, obsolete if absent at the final tag — and ground truth is defined directly off the final tag. One structural property is stated up front: the rule’s later-tag scan includes the final tag, so predicted-stale is a superset of ground-truth-stale and recall 1.0 holds *by construction*; precision is the one empirically meaningful number. Five arms score the same 13,626 beliefs through one confusion-matrix implementation, with no arm-specific filtering and no LLM anywhere.

Table 11: T4 Tier A staleness classification, 13,626 beliefs from 11 intermediate tags, run at 5d2bd81. Recall for the dreaming arm is guaranteed by construction; precision is the comparison column.

Arm	TP	FP	FN	TN	Precision	Recall	F1
dreaming (witness rule)	1,265	0	0	12,361	1.000	1.000	1.000
grep, symmetric normalisation	20	0	1,245	12,361	1.000	0.016	0.031
recency 30 d	920	5,677	345	6,684	0.139	0.727	0.234
recency 90 d	556	3,020	709	9,341	0.155	0.440	0.230
recency 180 d	0	0	1,265	12,361	0.000	0.000	0.000

Reading. Grep is precise but blind: all 20 of its stale predictions are correct, but it finds 20 of 1,265 stale beliefs. Recency baselines confirm what the TAD-v2 design assumes — wall-clock age is a weak staleness signal at every threshold. Dream precision 1.000 must be read with its limits: the corpus contains zero revert commits, so the one mechanism that can produce dream false positives (change-and-revert between sampled tags) never

occurs; the claim is *unfalsified on this corpus, not proven*. Determinism held: back-to-back reruns produce byte-identical metrics (provenance-stripped SHA-256 equal), and each result carries the benched repository’s HEAD, the stamper binary’s SHA, and per-tag stamping statistics — fields that make an empty or collision-blind path visible rather than treating a completed run as evidence of a valid label set.

Two rigged-eval catches are part of this result, not footnotes to it. Audits of this benchmark were explicitly directed to search for rigging in the proposed method’s favour, and found it twice. First, the grep baseline normalised Rust-style `::` qualification but not dot qualification, making qualified JavaScript symbols invisible to grep while dreaming saw the ledger identity; the repair (symmetric `::` and `.` handling) removed all 66 originally reported grep false positives, and the once-claimed dream-over-grep *precision* gap is withdrawn. Second, a deeper audit found the supposed ground truth itself was collision-bound: historical extraction collected definitions by (repo, file, kind, name) before witness qualification, collapsing coexisting same-named definitions — ten source-intact beliefs (six `CodeContext::is_empty`, four `format_text` methods on different stats types) were labelled stale by both prediction *and* ground truth, hiding operational false positives inside an apparent 1.000. That withdrew the pre-correction precision claim entirely (demonstrated ceiling 0.992 for that run). The honest response was not to hand-patch ten rows but to fix extraction before qualification — occurrence keys of (node, span, AST ordinal) — and rerun every arm from the same corrected labels; the table above is that rerun (belief population 13,575 $\rightarrow$ 13,626, stale 1,275 $\rightarrow$ 1,265). The second label tier (episode ancestry over 140 release tags: 558 reachable commits, 540 shipped, 0 reverted, 60 session-linked) contributes coverage only — its positive class is empty on this corpus, so reporting classifier metrics for it would be fabrication.

6 Discussion

6.1 What the results support

On this corpus and task, reinstatement recall materially outperforms one-shot kNN for provenance coverage at equal budget. The +53% spike lift cleared a pre-registered +25% gate with zero per-query losses, equal diversity (M2), and qualitative origin-rescue on the canonical failure case (M3). Mechanism attribution requires care, because it moved as the gold improved. Raw channel counting against file-touch coverage placed most GT hits in the TCM-style blend channel; the controlled ablation against decision-graded gold reverses this — graph-only beats the full walk on every graded metric — and we treat the graded result as authoritative. This reversal also changes the shape of the circularity defense: blend-dominance was the original argument that the coverage win was not an artifact of the graph channel sharing plumbing with the file-touch GT. That argument no longer carries the weight; what replaces it is that the graded gold itself is plumbing-independent — grades derive from dialog-acts and ship-event ledgers, not from `code_evolution` — so graph-channel dominance measured against it is not self-confirming in the way graph-vs-file-touch coverage would be. A corollary we own explicitly: the shipped `csr_why` default is the fused walk, and the ablation shows a blend-free graph-heavy configuration outscoring it on graded gold; the deployed configuration is provisional pending the fusion re-weighting

identified in E1. Production DoD confirms the walk can ship as a sub-100 ms, LLM-free MCP tool with honest offline eval.

This is a machine-assisted answer to Naur’s theory-loss problem for agentic coding: when theory lives in the intent trace of a saga, retrieval that only matches surface similarity loses the theory; retrieval that reinhabits seed context and spreads through artifact links can recover it (Naur, 1985; Howard & Kahana, 2002).

6.2 Observer effect / corpus self-contamination

We name the phenomenon *self-indexing evaluation contamination* (retaining “Heisenberg-like observer effect” as the informal gloss). A methodological observation at least as consequential as the coverage table: **measuring a self-indexing memory system changes the system**. CSR’s hooks import the evaluation session while the evaluation is designed and discussed. Near-verbatim restatements of queries and spike dumps become high-similarity competitors to origin conversations — exactly the failure mode one-shot retrieval already suffers, now amplified by the experimenters themselves.

Two consequences follow:

1. **Method.** Any eval of self-recording agent memory must run against a **frozen corpus snapshot** taken *before* the evaluation session can be captured and re-imported. Same-day “freeze after discussion” is insufficient. Prefer a session-zero backup. Report both clean and contaminated numbers when contamination is unavoidable; use the gap as a diagnostic, not as a reason to discard the clean run. A second, smaller comparability requirement surfaced during regression testing: ANN index construction is not deterministic across rebuilds (HNSW insertion effects), and per-query coverage counts near score boundaries can shift by ± 1 between two indexes built from the identical frozen corpus (observed: B=18 vs B=17, with two binaries producing per-query-identical results on the *same* index build). A/B comparisons must therefore share one index build, not just one corpus.
2. **Design.** Phase 1.5 applied provenance-aware reranking to the reinstatement pool inside `csr_why` and measured it on the frozen corpus (see Evaluation). The hypothesis — that echo demotion specifically counters contamination — was partially confirmed: query-echo defenses restored origin-surfacing on the canonical displaced query (Q3, 0 $\rightarrow$ 1) while holding aggregate coverage, but chunk-level demotion does not rescue decisions whose re-askings dominate whole conversations. Notably, two rerank heuristics that work in general-purpose search (`reflect_on_past`) were *rejected* by evidence in the provenance task: tool-mechanic chunks are evidence there, not noise, and flat user-authority boosts promote weak user chatter over strong evidence. Rerank policies are task-relative.

We hypothesize self-indexing evaluation contamination is a class-level issue rather than a CSR-only bug — the mechanism requires only that an agent index its own tool use while researchers discuss eval items, which would inject high-similarity confounds in any such system — but only one implementation has been tested; replication on other self-recording memory stacks is open.

A third instance surfaced during the v10 work, and it is the purest form of the class: the system was indexing *its own retrieval output*. CSR’s importer embedded its own MCP tool

calls and results, and its hook-injected context blocks, as ordinary conversation content. Measured on the live corpus, 747 conversations carried CSR-injected material — memory-manifest headers, past-context blocks, episode indexes — as searchable, high-similarity text. The loop is closed contamination: what the memory system says to the agent becomes what the memory system later retrieves. The fix follows the paper’s own prescription rather than fuzzy filtering: exact identity-based suppression of the system’s own tool payloads (with id-correlated results; unresolvable identities are kept, since false suppression of user content is worse than residual contamination) and exact-header scrubbing of hook-injected wrappers, applied through one shared sanitizer across every reflection-producing pipeline. The audit trail here repeated the silently-inert lesson above in inverted form: adversarial review found the sanitizer initially covered five of the pipelines while a sixth (episode extraction in the stop hook) still read raw transcripts — a partial fix that unit tests passed and only pipeline enumeration caught.

6.3 Silently-inert mechanisms: a second class-level failure

Preparing this paper surfaced the same failure shape twice, in unrelated mechanisms: a shipped component that does nothing, invisibly, while every unit test passes. First, the v2 ratification extractor’s operator-turn digest never fired in production — a storage-layer reconstruction hardcoded a default author label, so the digest builder’s operator filter matched nothing and extraction silently fell back to head/tail sampling; the run was reported (and believed) as testing the prioritized digest. Second, the resolution ledger’s write contract keys on chunk identifiers, but the search renderer originally never emitted them — verdicts written exactly per the tool’s own documentation would have silently no-opped. In both cases unit tests were green throughout: each layer honored its local contract while the composition was dead. Both were caught only by end-to-end replay of the real workflow (an external reviewer tracing the storage join; a live replay of the dogfooding queries through the production MCP client). We take this as a methodological finding on par with echo contamination: in agent-memory systems, where a mechanism’s output feeds a model that produces *plausible output regardless*, inert mechanisms do not crash — they degrade invisibly. End-to-end replay of the deployed path, not unit coverage, is the detection instrument; and when a mechanism is corrected, its experiment must be re-run (here, re-running the gate made the negative result stronger, which is the honest outcome working as intended).

6.4 One ambition, two mechanisms, one measured winner

The ratification programme and dreaming pursue the same ambition: prevent a memory system from presenting historically relevant but no-longer-current material as if it described present code. They differ in where they locate the verdict. Generative consolidation placed it in an LLM extraction of conversational acts and tried to turn a global per-conversation ratification score into a retention weight; it failed its pre-registered gate ($\rho = 0.036$ against sealed relevance grades, after the digest defect was corrected and the intended path genuinely ran), and the programme was halted with the negative result published. Its failure was a construct mismatch: “productive or accepted conversation” is global, whereas “still load-bearing for this artifact” is conditional on a particular claim and artifact.

Dreaming keeps the ambition and changes the mechanism. It does not ask a generator whether a belief *sounds* superseded; it joins the claim to an append-only span witness, compares hashes at commits, and abstains when closure or identity is not established. Under adversarial replay the repository-evidence mechanism survives exactly the cases that break conversational inference: exact reverts reinstate, non-identical $A \rightarrow B \rightarrow A'$ histories abstain, squash and cherry-pick identity is not reduced to wall-clock order. The contrast validates the contamination thesis in a stricter form: a generator reading a self-recorded corpus is exposed to the same echoes, scaffolds, and silently-inert joins as retrieval, while repository receipts are an independent evidential channel. One ambition; two mechanisms measured; the inference-based one failed its registered gate and the evidence-grounded one passed its substrate gates and the T4 benchmark above — with its end-task retrieval effect still unmeasured, and stated as such.

6.5 From measurement to mechanism: asserted state, not inferred state

The ratification negative result did productive work in the deployed system. A dogfooding incident posed the design problem: an agent queried memory for “items queued for a future binary” and four of five recalled items had already shipped in later sessions — retrieval was semantically correct, but the system indexes *prose*, *not state*, so a “queued” statement stays retrievable forever with no marker that a later session shipped it. The obvious fix — infer closure or supersession from conversation prose — is precisely the construct our pre-registered gate had just measured as unreliable ($\rho \approx 0.04\text{--}0.07$ across three extractors). The negative result forbade the inference path, and the shipped design is its constructive consequence: an *asserted-state resolution ledger*. The verdict about whether a remembered item is still true already exists, in-session, at verification time — the agent just checked the repository; the ledger gives that verdict a durable home instead of letting it evaporate as prose. A resolution tool records status (resolved / still-open / regressed) with mandatory cited evidence into an append-only ledger; at search time every returned chunk is annotated with its latest verdict and resolved chunks are demoted *within* the returned page (nothing is dropped or deleted); a regression appends a re-opening row that restores full rank. The epistemic division of labor is the point: retrieval stays similarity-based and complete, while *state* is only ever asserted by an agent that just performed a verification, with evidence attached — the system never claims to know an item is closed, it replays a cited claim that it was. This is a truth-maintenance move (Doyle, 1979) rather than a belief-revision one (Alchourrón et al., 1985): justifications are recorded and replayed, never inferred, and demote-don’t-delete preserves the full history that regression detection and provenance need. Together with the re-litigation benchmark the ledger brackets the staleness problem — the benchmark shows memory preserves *why things were settled*; the ledger makes it knowable *when they stopped being true*. Honest limits: a single dogfooding incident motivates the design; there is no controlled study of ledger benefit yet; and its value depends on verifying agents actually writing verdicts, a workflow property rather than a mechanism property.

Three live sightings during the preparation of this paper. The effect was not confined to the benchmark. (1) The attempted VCS-trailer GT construction was contaminated by its own extraction step (see Method): running `git log` inside an indexed session

imported the trailer IDs being extracted. (2) A file-history lookup made while drafting this paper returned, as its top hit, the session drafting this paper. (3) Most instructively, CSR’s own context-injection hook exhibited the failure in its selection layer: asked whether session-start continuity had captured the current work, the hook’s semantic intent classifier matched the prompt to a six-day-old episode consisting of *asking about that same hook* — at similarity 0.55, exactly its acceptance threshold — and attached its resume-context block to the stale episode instead of the live thread. The injection path had not received the Phase 1.5 echo defenses; the provenance tool had. The general lesson: in a self-indexing system, *every* retrieval surface — benchmark, analysis tooling, and context injection alike — needs echo defense and margin gating, because each one is a place where the system’s record of being asked can outrank the thing that was asked about.

6.6 Limitations

(a) Scale and design. Evaluation uses 12 hand-written queries on a single system and single corpus — no external dataset, no multi-organization coding history. M3 has been upgraded from single-annotator to a blind three-judge cross-vendor panel with moderate agreement (Fleiss $\kappa = 0.51$), but the judges are LLMs rather than humans, the judge instructions are hypothesis-aligned (they direct attention to re-askings and harness echoes; a neutral-prompt replication is queued), and 12 queries remains a small suite. The graded E2/E1 metrics are point estimates without interval estimates — with 7 of 12 queries at zero for both arms, a one-to-two-query swing moves the relative lifts materially. Paired-bootstrap intervals have since been produced for the expansion channels only, on a reduced five-arm re-run against a frozen evaluation unit (E1b): every contrast includes zero, so E1’s arm ordering stands as a within-grid ranking rather than a separated effect, and the AST arm’s null is additionally bounded by channel sparsity (fired on 8 of 20 queries) and by 50.7% attribution accuracy. The full seven-arm grid remains unbracketed. The behavioral benchmark’s rubric grades are author-graded blind-to-arm, not independent human evaluation.

(b) GT and graph plumbing. GT is built from `code_evolution`, which Arm B also uses in step 3. Circularity is mitigated by seed dependence and by blend-channel dominance in attribution, but not eliminated. An attempted independent GT from VCS commit trailers failed on sparse trailer coverage and on self-contamination of the ID bridge (see Method); the mitigation path is specified but not yet realized.

(c) Observer effect partially mitigated, not solved. Frozen snapshots and Phase 1.5 echo defenses repair the measured displacement cases but do not rescue origin conversations for decisions whose re-askings dominate entire conversations; conversation-level exclusion and supersedes chains remain open work.

(d) Episode-chain value revised, not settled. The episode hop contributed zero GT hits in the coverage-proxy era (sparse/young `prev_episode_id` chains); on graded gold the episode-only ablation arm is a strong origin-finder (0.321 origin-MRR) but weak on evidence depth. Its value is now positive but characterized on 12 mapped queries only.

(e) Generalizability. The primary suite is maximally dogfooded (CSR indexing its own development). Within-operator cross-project replication on a second, unrelated corpus

(TypeScript product/marketing history, +47% vs. the primary +53%) removes the self-description confound but not the single-operator one: all indexed history is one person’s work with one agent stack. Multi-operator and multi-organization validation remain open.

(f) GT layers and their residual weaknesses. The coverage results use a file-touch proxy (all `code_evolution` sessions touching the target file count equally), which counts design-originating, implementing, and merely-discussing sessions as equally correct and shares plumbing with Arm B’s graph channel; those results should be read as coverage of *decision-relevant activity*. The graded gold (E2) removes the proxy but has its own limits: grades derive from LLM-extracted dialog-acts (dual-vendor consensus, directs $\kappa = 0.41$ — moderate; splits resolved conservatively and routed to owner audit, which the owner completed only for the five contested origin maps), the gold is same-corpus (mitigated by feature disjointness: grades come from dialog-acts and ship ledgers, retrieval from embeddings and graph structure), and origin-MRR rests on $n=12$ single-operator sealed recollections.

(g) Concurrent-work race. ACT-Up, MRMS, and E-mem have now been close-read in full (Related Work reflects their actual mechanisms and evaluations), but the space is moving quickly; other concurrent preprints may exist that we have not surveyed, and the composition could be assembled by others within months.

(h) Incomplete three-source modeling. CMR motivates source-factored reinstatement of intent vs. deliberation vs. artifact (Polyn et al., 2009). Phase 1 labels sidechains and stores sequence but still uses a single fused context blend and does not yet weight by `is_sidechain` or `seq`.

6.7 Deferred work and theoretical next steps

The three experiments queued in earlier drafts — the factored ablation grid, graded provenance gold, and the controlled contamination experiment — have been executed and are reported above (Phase 2 experiments). What they leave open sets the queue. From E1: fusion re-weighting (a larger graph share, per-channel quotas) and a blend-free graph+episode arm. From E2: the origin-finding floor — 5 of 12 owner-verified origins missed by every arm, concentrated in the corpus’s earliest era — motivating age-compensating retrieval; and a cross-persona replication with marketing-operations queries against the machine’s purpose-built release-train ledger. From E3: conversation-level echo exclusion (chunk-level demotion repairs occupancy but cannot rescue origins whose re-askings dominate whole conversations) and *supersedes* population. From the ratification negative result: node-level act-strength weighting is closed by its pre-registered gate, not deferred; the formulation that remains open is edge-level ratification — acts bound to (conversation, artifact) pairs, weighting reinstatement graph edges and joined query-conditionally — which is untested.

Honest attribution, and the AST rematch it gates. The follow-up’s second bound (E1b) is a work item, not only a caveat. Symbol→conversation attribution is today a file-level projection: 50.7% agreement with evidence-bearing change events against a 90% bar, with 499 of 532 files carrying one uniform id. Two independent channels can replace it without inferring anything. Transcript change events cover 33.1% of indexed symbols (2237 of 6767), with a structural ceiling of 44.5% once the kinds that never appear in change events

(constants, modules) are excluded — sparse, but exact and fine-grained. Git introducers, recovered as the first line of `git log -L` over each symbol’s stored span, corroborate the transcript channel on 82.2% of the overlap within 48 hours (1103 of 1342, median gap 1.3 hours) on the receipt run’s clean-span sample — 74.1% (1611 of 2174) on the full live backfilled table, whose denominator includes every dual-channel symbol rather than the sampled clean-span subset; both are reported rather than reconciled — and reach 95.1% of the symbols that have no transcript event at all — dense, but coarse: single squash-merge commits introduce between 1 and 393 symbols. The asymmetry is the design, not a defect: git covers deleted and pre-ledger history, transcripts resolve granularity inside a squash, disagreements are surfaced as two labeled channels rather than silently merged, and symbols with neither channel are rendered `unattributed` rather than inheriting their file’s first toucher. Shipping that, and only then re-running the AST arm on it, is the path by which the structural null becomes falsifiable.

Learned decay (Anderson & Schooler, 1991) is theoretically motivated and now *unblocked* by MCP retrieval logging, but blocked on log accumulation. Chunk-level temporal reinstatement using `seq`, sidechain-aware scoring, and `SessionStart` saga narration remain non-goals of this phase. External and multi-operator validation is the immediate scientific next step.

7 Conclusion

We introduced *three-trace sagas* — joint episodic structure over intent, deliberation, and artifact in agentic software construction — and *reinstatement recall*, a five-step, LLM-free retrieval algorithm that seeds with kNN, blends query and seed context (TCM analogue), spreads over a session–file code graph, hops episode chains, and fuses results under a provenance-aware rerank. On CSR’s own development corpus, a pre-registered spike showed +53% summed GT-session coverage at $k = 10$ versus one-shot kNN, with mechanism attribution locating most gains in semantic context blend rather than graph walk. Production reimplementations as `csr_why` met latency and integration DoD under partial same-day self-contamination (15 $\rightarrow$ 18/19), and a frozen-corpus rerank pass repaired the contamination-displaced acceptance query (0 $\rightarrow$ 1) while holding aggregate coverage.

Supported on these corpora, at the retrieval-metric level: reinstatement beats flat kNN for multi-hop provenance questions, at sub-100 ms warm cost, without an LLM in the loop — supported by GT coverage on two unrelated within-operator corpora (+53% Rust/systems, +47% TypeScript/product; pooled sign test $p \approx 0.001$), by blind cross-vendor judge panels on both (8/12 preferred, $\kappa = 0.51$; 5/8 preferred with zero baseline votes; instructions hypothesis-aligned as disclosed in Evaluation), and by ratification-derived graded gold (+37% origin-MRR, +29% `nDCG@10`, +25% graded recall — point estimates on $n=12$ owner-sealed origins, without interval estimates on those lifts; intervals exist for the expansion-channel contrasts only, from the E1b follow-up’s 10,000-resample paired bootstrap at seed 20260731, and the full seven-arm grid remains unbracketed). On graded gold the channel ablation attributes the gain to code-graph spread — the session $\leftrightarrow$ file co-edit trace, reversing the coverage-era blend attribution, with the graded gold plumbing-independent of the graph channel — as a within-grid ranking on decision-graded gold rather than a causal attribution: the sham-controlled follow-up measures that channel’s marginal

contribution as directionally positive but not separable from zero (+0.017 origin-MRR, 95% CI lower bound exactly 0.000), and finds the AST-structural arm a strict null against both the base arm and its own degree-preserving sham. The ablation also shows the echo defense is causally necessary (removing it falls below the kNN baseline); controlled contamination dosing shows five re-ask cycles capture half of naive retrieval’s top-10 and that the same defense repairs 72% of echo occupancy at $3.4\times$ the origin-MRR, with a null sham control. A blinded three-arm behavioral benchmark adds the deployment-level claim: conversation memory — an effect plain kNN delivers equally — roughly halves reason-quality failures on settled-for-cause decisions (10 and 9 vs. 6 of 12 under sealed tasks and author-graded, blind-to-arm rubric grading), with the no-memory arm accepting the re-introduction of a shipped regression that both memory arms decline for the recorded reason.

Measured as a tie: on single-hop re-litigation tasks, reinstatement and one-shot kNN produced identical actions on all sixteen tasks and identical blinded reason-quality totals (27–27). Reinstatement’s advantage is real only where its multi-hop mechanism has work to do; the single-hop tie is reported with the same prominence as the multi-hop win, and a multi-hop *behavioral* benchmark remains the open experiment. The same benchmark surfaced an arm-independent conservatism bias — all arms deflect justified changes whose own exit conditions are met — which memory does not cause but also does not cure.

Not yet proven: learned usage-weighted decay; sidechain-weighted scoring; chunk-sequence temporal reinstatement; generalization beyond a single operator’s history; benefit of the asserted-state resolution ledger (motivated by one dogfooding incident, no controlled study); and origin recovery itself — the verified origin conversation stays outside every arm’s top-10 on over half the graded queries, and re-askings that dominate whole conversations remain unrescued by chunk-level defenses.

Added since the first version of this record: evidence-grounded forgetting (“dreaming”) — an append-only span-witness ledger with deterministic, abstention-first staleness verdicts and demote-plus-annotate consumption. On the T4 replay benchmark (13,626 beliefs, five arms, one scorer) the witness rule classifies staleness at precision 1.000 where fair grep achieves recall 0.016 and recency baselines peak at F1 0.234 — stated with its limits (recall by construction; precision unfalsified on a zero-revert corpus; two rigged-eval defects caught by directed adversarial audit and corrected before these numbers were admitted). Its end-task retrieval effect is the open experiment.

Ruled out by pre-registration: global per-conversation ratification weighting as a retention signal — Spearman against the sealed graded gold measured $\rho = 0.060 \rightarrow 0.071 \rightarrow 0.036$ across three extractor versions ($n = 111\text{--}123$), halting that program at its pre-registered gate; the result survived a post-hoc mechanism correction (a digest bug found in external review), and only the edge-level, query-conditional formulation remains open.

Immediate next steps: conversation-level echo exclusion and supersedes population; enforce session-zero frozen snapshots for all future evals; accumulate MCP retrieval logs for Phase 2 learned decay; a multi-hop behavioral benchmark (tasks whose justification chain spans multiple conversations — the regime where reinstatement can still separate from kNN); a controlled staleness study of the resolution ledger (replay historical sessions,

measure stale-recall rate with and without verdicts); the AST rematch — re-running the sealed E1b kit with symbol seeds mapped through evidence-bearing two-channel attribution (`code_node_attribution`: transcript and git) instead of projected `first_conv_id` — together with method-edge resolution for the largest unresolved edge class and `repo_defs` densification toward a repo-complete structural corpus; a neutral-prompt judge-panel replication; a small multi-annotator regrade of the behavioral rubric; validate on external agentic coding corpora.

For agentic construction, Naur’s theory is no longer only in human heads — it is stranded across sagas. Systems that index only similar text will keep losing it. Systems that reinstate the encoding context of the work that produced the code have a chance to keep it.

Data availability and privacy

The evaluation corpus is one operator’s private development history — session transcripts, subagent traces, and code evolution across personal and commercial projects — and cannot be released as-is. Transcript excerpts quoted in this paper were reviewed by the corpus owner before inclusion. The evaluation *harness* (frozen-snapshot protocol, MCP probe scripts, blinded judging packets, agreement computation) contains no corpus content and is releasable; external replication therefore requires running the harness on one’s own agent history, which is exactly the deployment condition the system targets. The Phase-2 harness is released: the E2/E1/E3 extraction prompts, grading and scoring scripts, and protocol specs are available at `csr-engine/eval-kit/` in the repository (sanitized; sealed gold labels and private transcripts excluded).

The T4 staleness benchmark is fully releasable and needs no private corpus at all: `codewitness` labels and `codewitness` bench derive both label tiers and all five arms from any git repository’s own history. Runs are deterministic (byte-identical metrics across reruns), and every result carries a provenance block — repository HEAD at run time, stamper-binary SHA-256, and per-tag stamping statistics. The harness, this paper’s result files, and the per-tag detail are at `csr-engine/eval-kit/t4/`; the retired Python label/replay scripts produce no paper numbers.

References

- Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2), 510–530.
- Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2(6), 396–408.
- Deng, Y. (2026). Entity-collision: A stratified protocol for attributing retrieval lift in agent memory. *arXiv preprint* arXiv:2605.29630.
- Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence*, 12(3), 231–272.

- Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025). From RAG to memory: Non-parametric continual learning for large language models. *Proceedings of ICML 2025*. arXiv:2502.14802.
- Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3), 269–299.
- Ko, A. J., DeLine, R., & Venolia, G. (2007). Information needs in collocated software development teams. *Proceedings of ICSE 2007*, 344–353.
- Kunz, W., & Rittel, H. W. J. (1970). *Issues as elements of information systems* (Working Paper 131). Institute of Urban and Regional Development, University of California, Berkeley.
- LaToza, T. D., & Myers, B. A. (2010). Hard-to-answer questions about code. *Proceedings of PLATEAU 2010*. ACM.
- Li, J., & Shi-Nash, A. (2026). MRMS: A multi-resolution memory substrate for long-lived AI agents. *arXiv preprint* arXiv:2607.04617.
- MAGMA. (2026). MAGMA: A multi-graph based agentic memory architecture for AI agents. *arXiv preprint* arXiv:2601.03236.
- Moran, T. P., & Carroll, J. M. (Eds.). (1996). *Design rationale: Concepts, techniques, and use*. Lawrence Erlbaum Associates.
- Naur, P. (1985). Programming as theory building. *Microprocessing and Microprogramming*, 15(5), 253–261.
- Nygard, M. (2011). Documenting architecture decisions. Cognitect blog, November 15, 2011.
- Polyn, S. M., Norman, K. A., & Kahana, M. J. (2009). A context maintenance and retrieval model of organizational processes in free recall. *Psychological Review*, 116(1), 129–156.
- Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (2025). Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint* arXiv:2501.13956.
- Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313–323). Prentice-Hall.
- Thomson, R., & Lebiere, C. (2026). Rapid prototyping of event-driven contextual memory in the ACT-Up cognitive architecture. *arXiv preprint* arXiv:2606.28045.
- Wang, et al. (2026). E-mem: Multi-agent based episodic context reconstruction for LLM agent memory. *Proceedings of ICML 2026*, PMLR 306. arXiv:2601.21714.
- Wu, R., Zhang, H., Kim, S., & Cheung, S.-C. (2011). ReLink: Recovering links between bugs and changes. *Proceedings of ESEC/FSE 2011*, 15–25.
- Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., & Zhang, Y. (2025). A-MEM: Agentic memory for LLM agents. *Proceedings of NeurIPS 2025*. arXiv:2502.12110.