Hybrid Holographic Memory in a Production Personal Agent: A Seventeen-Day Empirical Characterization of Compounding, Ontology-Gated Recall on Live Operational Data

William White*[1]

[1]Bay West Labs | Sheldon Dynamics

July 17, 2026

Abstract

This report characterizes the hybrid memory substrate of a production personal agent (Hermes v0.18.0, instance "Backbone") running continuously on a 2 GB commodity VPS. The substrate unifies five recall channels in a single 16.3 MB SQLite file: a trust-weighted holographic fact store (1,214 facts, 100 % coverage with 1,024-dimensional HRR phase vectors), an exact-kNN semantic index (1,628 MiniLM vectors via sqlite-vec), a canonicalized knowledge graph (1,464 nodes, 2,544 edges under a closed 35-relation vocabulary), a Hofstadter-inspired analogy slot, and situation playbooks. We combine three instruments, all read-only against the live store: (i) longitudinal telemetry mined from the system's own recall and trust ledgers (1,976 recall events across 235 sessions over 1–17 July 2026); (ii) two replay experiments, a 200-query known-item retrieval probe (best-channel Recall@5 of 54.0 %) and a full replay of all 50 production firings of the analogy slot, which confirms the disagreement filter empirically: mean query-candidate surface cosine 0.102, maximum 0.266, zero violations of the 0.30 design ceiling, against a 0.288 mean for the ordinary semantic channel under the identical measurement; and (iii) latency microbenchmarks placing the full five-block recall pipeline at 37.4 ms median (87.9 ms p95) on two shared vCPUs, with zero API calls on the hot path. Trust dynamics are strictly use-driven (temporal decay disabled by design): 399 ledger events show near-symmetric reinforcement (+0.020) and demotion (−0.019), with demoted facts floored at 0.35 rather than deleted. Abstraction has begun compounding: 9 LLM-summarized chunks spanning 52 member facts, 9 episode nodes, and 138 machine-proposed ontology edges held in an operator review queue that the runtime graph provably excludes. We report design constants, store composition, growth series, failure inventory, and threats to validity in full.

---

1 Introduction

The Sheldon Dynamics memory program targets a single question: *can a personal agent's memory compound, rather than merely accumulate, on hardware a solo operator can afford?* The alternative posture, stateless retrieval-augmented generation over an ever-growing vector

*Correspondence: sheldon@baywestdigital.com. Affiliation: Bay West Labs | Sheldon Dynamics.

store, accumulates text but never re-organizes it: nothing is canonicalized, nothing is abstracted, nothing is forgotten, and nothing is ever surfaced *because it is structurally, rather than topically, relevant.* The position taken here, argued at length in the companion essay,[1] is that analogy is not a luxury feature of cognition but close to its core, following Hofstadter's framing, and that an agent memory should therefore maintain *two* similarity spaces: a surface space for topical recall and a structural space in which distant experiences can rhyme.

The system under test is the live memory substrate of "Backbone," a Hermes-based personal agent that has run continuously in production since May 2026 on a single 2 GB DigitalOcean droplet, handling email triage, job-application tracking, client operations for Bay West Labs, calendar reasoning, and outbound research. The substrate is not a demo: every number in this report is measured against the production store while the agent remained in service, using read-only instruments.

Three architectural commitments distinguish the design:

1. **One substrate, five channels.** Facts, full-text index, semantic vectors, knowledge graph, HRR phase vectors, telemetry, and trust ledger live in one SQLite file. There is no external vector database, no graph server, and no memory-related API call on the per-turn hot path.
2. **Ontology under governance.** Entity names are canonicalized at write time; relations are constrained to a closed 35-predicate vocabulary; machine-proposed ontology changes (alias merges, category promotions) enter a review queue and are excluded from the runtime graph until an operator approves them.
3. **Use-driven forgetting, no temporal decay.** Age is never an input to ranking or deletion. Trust moves only on evidence of engagement or sustained non-engagement, and demotion floors at 0.35 rather than deleting, so a demoted fact can recover.

This report is a *characterization study* of a production system, methodologically sibling to the fourteen-run Little Bear program [1]: where that program tested a decision-making substrate against a frozen holdout, this one instruments a memory substrate in vivo. We state plainly what that buys and what it costs: full ecological validity, no synthetic fixtures, but a single subject, no pre-registration, and observational rather than controlled evidence for most claims. The two replay experiments (§6, §7) are the exception: both are deterministic, seeded, and re-runnable against the checked-in dataset.

2 System under test

Hermes Agent v0.18.0 with the `hybrid` memory provider, a 1,758-line plugin subclassing the upstream holographic provider, plus a pipeline of six asynchronous cron stages. Everything persists in `memory_store.db`: 16.3 MB main file plus 4.3 MB write-ahead log at capture, i.e. roughly 17 kB per fact all-in (vectors, graph, gists, telemetry included).

2.1 Substrate layers

The curated layer is deliberately tiny: `MEMORY.md` is capped at 2,200 characters and `USER.md` at 1,375, frozen at session start and edited only through an explicit tool. Everything else is machine-managed. Both files are mirrored into the vector index on write, so curated notes are also semantically recallable.

[1] sheldondynamics.com/blog/hybrid-memory

Layer	Store	Live rows	Role
Curated	`MEMORY.md` / `USER.md`	9 + 9 sections	Operator-visible working memory, char-capped
Facts	`facts` (+FTS5)	1,214	Distilled statements, trust-scored, HRR-encoded
Gists	`fact_gists`	1,214	One-line LLM paraphrase per fact
Vectors	`vec_index` (vec0)	1,628	MiniLM 384-d, cosine, exact kNN
Graph	`graph_nodes` / `edges`	1,464 / 2,544	Canonical entities, closed 35-relation vocabulary
HRR	`facts.hrr_vector`	1,214 (100 %)	1,024-d phase vectors, 8 kB each
Banks	`memory_banks`	15	Per-category HRR superpositions (emergent categories)
Telemetry	`recall_log` / `trust_log`	1,976 / 399	Every shipped line; every trust change
Abstraction	`chunk_members`	52	Chunk membership (9 chunks, episodes in graph)

Table 1: Substrate inventory at capture (2026-07-17). One SQLite file holds all layers.

2.2 The read path: five-block prefetch

Every turn, the provider assembles up to five blocks in fixed priority order, then hard-caps the total at 2,000 characters (whole lines only; earlier blocks win):

1. **Holographic FTS.** The upstream retriever scores candidates by $0.4 \cdot \text{FTS5} + 0.3 \cdot \text{Jaccard} + 0.3 \cdot \text{HRR}$, then multiplies by trust; minimum trust 0.3; top 5 shipped; operational categories (email, lead) excluded from auto-injection.
2. **Semantic recall.** MiniLM cosine over the vec0 index, query built from a rolling window of the last 3 messages, threshold 0.40, deduplicated against block 1, top 5. Chunk suppression runs here: if a chunk and one of its members both surface, the member is dropped.
3. **Graph neighborhood.** Up to 2 entities matched in the query via the alias map, 1-hop neighbors from an in-memory DiGraph, at most 10 lines.
4. **Analogy candidate.** At most one line (§2.3).
5. **Situation match.** Stored situation playbooks at cosine ≥ 0.42, at most 2.

Every line that actually ships increments the fact's retrieval count and writes a `recall_log` row tagged with its block. This self-instrumentation is what makes §5 possible.

2.3 The analogy slot

Each fact carries a 1,024-dimensional HRR phase vector built by binding role and filler atoms (deterministic SHA-256 phase encoding). The query is encoded the same way and compared against a cached matrix of all fact HRR vectors. Because HRR similarities compress toward 0.5, the gate is relative, not absolute: a candidate must exceed $\max(0.50,\ \mu + 2\sigma)$ of the turn's similarity distribution *and* have MiniLM surface cosine ≤ 0.30 against the query. The slot ships at most one candidate per turn, never repeats the last 20, and requires queries of at least 40 characters. This is the "disagreement filter": structurally resonant, topically distant. Section 7 tests whether production behavior matches this specification.

2.4 The write path: distillation, canonicalization, closed vocabulary

Raw text is never stored. Email and call events spool to JSONL; every 15 minutes an ingest stage sends each document (truncated to 6,000 characters) to `gemini-2.5-flash-lite`, which returns at most 8 facts, 12 nodes, and 12 edges. Every entity name passes through a resolver keyed on domain, then email, then phone, then normalized string: similarity ≥ 0.92

merges silently, 0.85–0.92 files a `POSSIBLE_ALIAS` edge for operator review. Every relation passes through a canonicalizer into a closed vocabulary of roughly 35 predicates (passive forms are direction-flipped; unmappable relations fall back to `RELATED_TO` with the original preserved in `rel_orig`). Facts get first-assertion provenance (`source_ref`). A sync stage then vectorizes new items, mines graph edges from new facts, and distills one-line gists (batch 25 each).

2.5 Governance: trust, review queue, provenance

Nightly, a feedback stage replays the recall log: a fact counts as *engaged* if a later same-session query mentions one of its linked entities or overlaps it at token-Jaccard ≥ 0.3; engaged facts gain $+0.02$ (ceiling 0.85), facts surfaced on ≥ 5 distinct turns in 14 days with zero engagement lose 0.02 (floor 0.35, above the 0.30 retrieval threshold, so demotion never deletes). Weekly, a consolidation stage deduplicates and archives junk (archived rows are appended to a JSONL archive before deletion, never silently dropped), and an abstraction stage clusters old facts (union-find on shared entities, cluster size 3–12, minimum age 7 days) into LLM-summarized *chunks* with an `episode` node each, and proposes `INSTANCE_OF` category promotions (relational-fingerprint cosine ≥ 0.8). All proposals carry `src_tag` values that the runtime graph loader excludes by default: the machine can propose ontology, only the operator can commit it.

2.6 Design constants

Group	Constant	Value	
Embedding	model / dim	MiniLM-L6-v2 / 384	
HRR	dim / bytes per fact	1,024 / 8,192	
Retrieval	FTS:Jaccard:HRR weights	0.4 : 0.3 : 0.3	score × trust
	prefetch char cap	2,000	vector threshold 0.40
	recall tool floor	0.25	rolling window 3 msgs
Analogy	HRR gate	$\max(0.50, \mu{+}2\sigma)$	surface ceiling 0.30
	min query chars / repeat guard	40 / last 20	≤ 8 candidates per turn
Entities	silent merge / review	0.92 / 0.85	closed vocab ≈ 35 rels
Trust	default / floor / ceiling	0.50 / 0.35 / 0.85	deltas ± 0.02
	temporal decay half-life	0 (disabled)	recall log retention 60 d
Abstraction	cluster size / age	3–12 / ≥ 7 d	fingerprint cosine ≥ 0.8
Ingest	per-doc caps	8 facts / 12 nodes / 12 edges	text cap 6,000 chars
Cadence	ingest / feedback / consolidate	15 min / daily 03:45 / weekly	review nudge weekly

Table 2: Design constants of the substrate, verified in source and live configuration.

2.7 Substrate diagram

Figure 1 summarizes the write path, the store, the read path, and the governance loop as they run in production.

2.8 Lineage

The current design is the third generation. A ChromaDB + NetworkX prototype was retired on 2026-06-29 in favor of the single-SQLite store. The "Hofstadter upgrade" (2026-07-01) added HRR dual-encoding, the analogy slot, chunking, and emergent category banks. "Track B" (2026-07-02) added entity hygiene filters, dated-event extraction, and calendar-to-graph

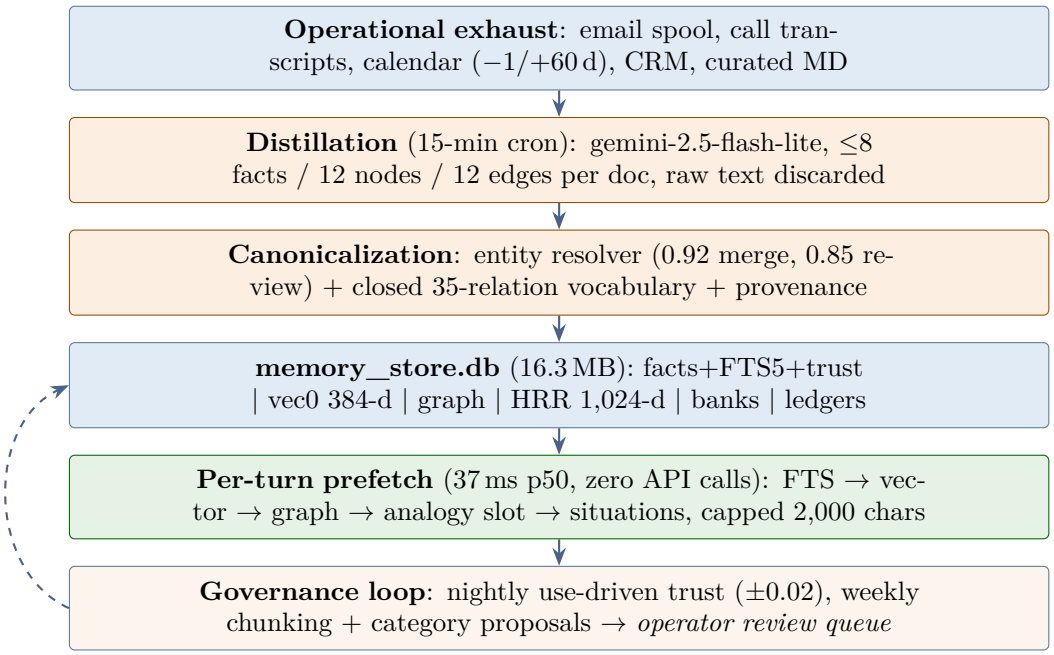

Figure 1: The substrate under test. Write path flows down through distillation and canonicalization; the governance loop writes trust and abstraction back into the store.

episode sync. A third phase (2026-07-06) closed the relation vocabulary (a migration collapsed 344 historical relation strings to the canonical set; 35 are live), added provenance columns, decision traces, and the operator review gate. The telemetry window of this report therefore begins 2026-07-01, the first day the recall ledger existed.

3 Method

Three instruments, all executed against the production store in SQLite read-only mode while the agent remained in service; no experiment writes to the store.

1. **Ledger mining.** The substrate logs every shipped recall line and every trust change as a side effect of normal operation. We treat these ledgers (1,976 recall rows, 399 trust rows, 1,424 ingest-sync runs) as found longitudinal data. The instrumentation predates this study and was not modified for it.
2. **Deterministic replays.** E1 (§6) re-queries the store with 200 seeded known-item probes. E2 (§7) re-embeds the logged query of every production analogy firing and recomputes surface similarity with the same ONNX MiniLM model the system uses.
3. **Microbenchmarks.** Component and end-to-end latencies measured on the production host (2 shared vCPUs, 2 GB RAM plus 2 GB swap) with 60 timed iterations after warmup, reported as p50/p95.

What is and is not pre-registered. Nothing in this study was pre-registered; it is a characterization, not a hypothesis test. However, the design thresholds it checks (the 0.30 analogy surface ceiling, the 0.40 vector threshold, the trust floor at 0.35) were committed to configuration before the telemetry window opened, so §7's central result (production behavior conforms to a pre-committed specification) is not post-hoc curve fitting. Where an instrument is optimistic (the fused channel in E1) or approximate (query reconstruction in E2), the text says so at the point of use.

4 S1: Store composition and growth

Category	Facts	Mean trust	Retrievals
email	750	0.500	96
lead	127	0.500	6
general	109	0.495	243
project	97	0.520	1,033
claude-memory	48	0.497	102
user_pref	46	0.475	242
job application	16	0.500	17
chunk	9	0.483	8
other (7 cats)	12	0.517	177
Total	**1,214**		**1,924**

Edge provenance	Edges
ingest	1,897
curated	445
proposed (gated)	78
alias-candidate (gated)	60
chunk	47
calendar	17
Total	**2,544**
runtime-visible	2,211

Table 3: Left: fact composition. The operational exhaust (email, lead) dominates row count but not retrieval; the project category alone accounts for 54 % of all retrievals. Right: edge provenance. The 138 gated rows (proposed + alias-candidate) are excluded from the runtime graph until operator review.

Two facts about Table 3 carry the composition story. First, retrieval is sharply skewed away from bulk: email is 62 % of facts but 5 % of retrievals, while project facts (8 % of the store) draw 54 % of retrievals. The exclusion of operational categories from auto-injection is doing its job: bulk exhaust is reachable through explicit tools but does not crowd the per-turn context. Second, the operator gate is live, not aspirational: the runtime graph loader excludes exactly the 138 gated edges (verified by loading the graph with production exclusion tags: 2,211 edges visible versus 2,544 raw).

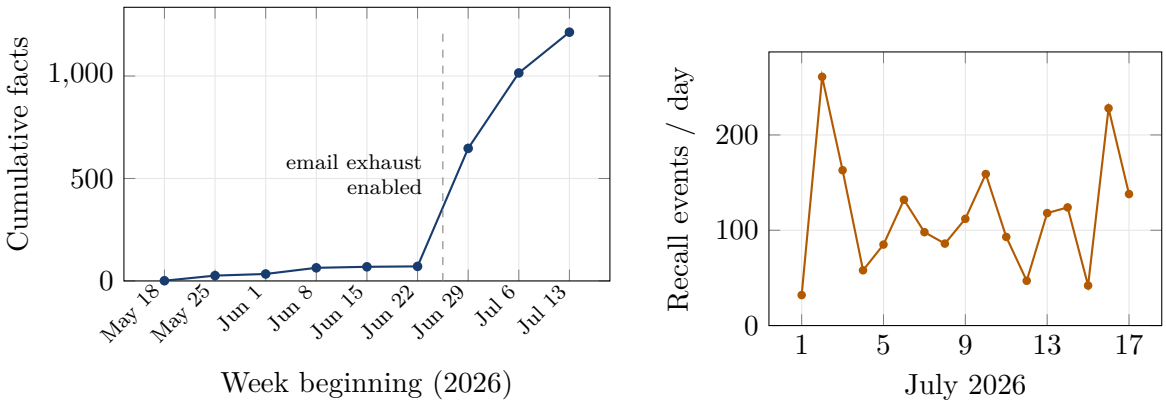

Figure 2: Left: cumulative fact growth. The inflection at the week of June 29 is the single-SQLite migration plus the email-ingest turn-on: 576 facts in one week, mostly operational exhaust. Right: daily shipped recall lines over the telemetry window (mean 116/day, 235 distinct sessions).

The entity graph is heavy-tailed, as knowledge graphs over one person's operations should be: 552 of 1,473 runtime nodes have degree 1, while the top hub (the operator node) reaches degree 220, followed by the home city (Tampa, 76) and active projects. Fitting no distribution, we simply note the shape in Figure 3 and that hub suppression exists in the abstraction stage (entities linked to more than 15 facts are ignored for cluster seeding, preventing chunks like "everything about Tampa").

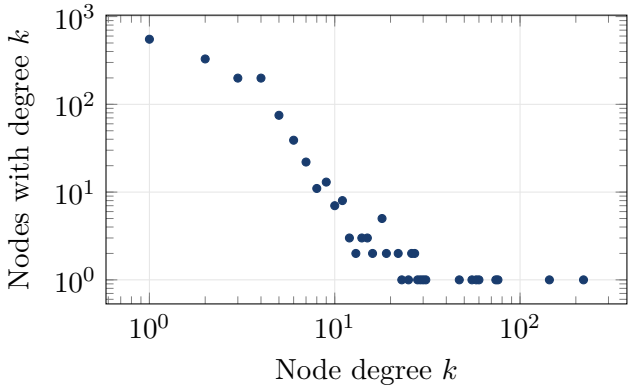

Figure 3: Degree distribution of the knowledge graph (log-log). Heavy tail with the operator node as the top hub (degree 220); hub-aware clustering prevents degenerate abstraction.

Ingest throughput over the window: 1,424 incremental sync runs added 830 vectors and pruned 18 orphans; the vector index grew from 589 to 1,628 items. The ingest spool currently holds 25 failed document batches retained for inspection (LLM extraction errors); these emails are absent from memory, and we count them as a known coverage gap rather than silently ignoring them.

5 S2: Retrieval telemetry

Channel	Events	Share	Median score	IQR	Max
fts (holographic)	1,154	58.4 %	0.220	0.202–0.247	0.566
fact_store (legacy JSON)	497	25.2 %	0.269	0.209–0.302	0.512
vector (semantic)	259	13.1 %	0.410	0.406–0.526	0.908
analogy	50	2.5 %	0.532	0.527–0.539	0.590
recall tool (explicit)	15	0.8 %	0.480	0.471–0.626	0.769
situation	1	0.1 %	0.449		
Total	**1,976**				

Table 4: Shipped recall lines by channel, 1–17 July 2026. Scores are channel-native (BM25-derived for fts, cosine for vector, HRR similarity for analogy) and are not comparable across rows; the analogy channel's tight band (0.518–0.590) reflects the compression of HRR similarity around 0.5.

Turn-level structure: 274 turns shipped at least one recall line. The modal combination is fts+vector (117 turns), followed by fts alone (66). The analogy slot fired on 49 turns (17.9 % of recall-bearing turns), shipping exactly one candidate on 48 of them and two once (a same-turn re-entry); across its 50 events it surfaced 39 distinct facts, so the slot is exploring the store rather than replaying one association: the most-repeated analogy candidate appeared three times, and the repeat guard (last 20) bounds recurrence by construction.

The channels are complementary by design and the telemetry bears this out: the fts channel ships the most lines at modest scores (trust-weighted lexical-plus-HRR blend), the vector channel ships fewer, higher-similarity lines (its 0.40 threshold truncates the left tail, visible in Table 4), and the analogy channel is rare and tightly banded, exactly what a $\mu + 2\sigma$ relative gate produces.

6 E1: Known-item retrieval

Design. For each fact, the ingest pipeline stores a one-line gist: an LLM paraphrase written by `gemini-2.5-flash-lite` at ingest time. We sample 200 (fact, gist) pairs (fixed seed 14, gists under 30 characters excluded), issue the gist as a query, and ask whether each channel retrieves the source fact: FTS5 with BM25 ranking (OR-query over the first 12 content tokens), and exact kNN over the vector index with the query embedded by the production MiniLM model. The fact's own gist vector is excluded from the vector channel (it would be a trivial self-match). We also report a best-of-either rank, which is an *oracle channel selector* and therefore an upper bound on any parameter-free fusion, not an implemented system.

Channel	R@1	R@5	R@20	n
FTS5 (BM25)	33.0 %	47.5 %	59.0 %	200
Vector (MiniLM cosine)	17.5 %	31.5 %	49.0 %	200
Best-of-either (oracle)	38.0 %	54.0 %	68.5 %	200

Table 5: Known-item retrieval with the fact's own gist as the query.

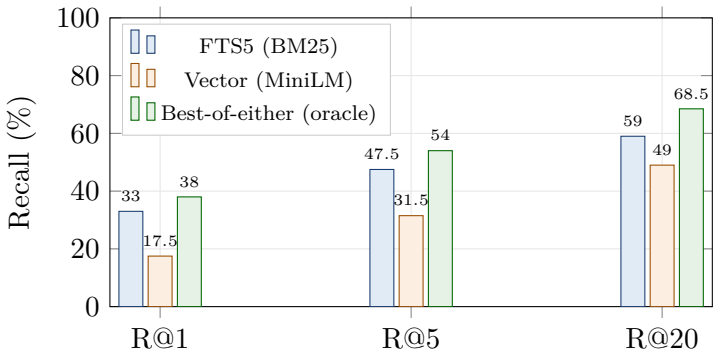

Figure 4: E1 results. The channels err on different queries: the oracle gains +6.5 points of R@5 over the better single channel, which is the empirical case for running both.

Reading. Three observations. First, the absolute numbers are moderate, and honestly so: a gist is a maximally compressed paraphrase, frequently dropping every distinctive token of its source, and 62 % of the store is near-duplicate-rich operational email against which a one-liner underdetermines its source. Second, the lexical channel beats the semantic one on this probe; gists share salient tokens with their sources by construction (same author model, same source text), which advantages BM25 and is a circularity we flag rather than hide. Third, the oracle gap (+6.5 R@5 over FTS alone, +19.5 over kNN at R@20) shows the channels fail on *different* queries, which is the design rationale for a hybrid: no single channel dominates.

7 E2: The disagreement filter, replayed in full

Design. The analogy slot's specification commits to two inequalities per shipped candidate: HRR similarity above a relative gate, and MiniLM surface cosine at most 0.30 against the query. The recall ledger stores, for each of the 50 production firings, the query text and the HRR-side score, but not the surface cosine. We therefore re-embed every logged query with the production embedding model, fetch the shipped fact's stored surface vector from the

index, and recompute the surface cosine. For comparison, we apply the identical procedure to 150 events from the ordinary semantic channel and 150 from the fts channel.

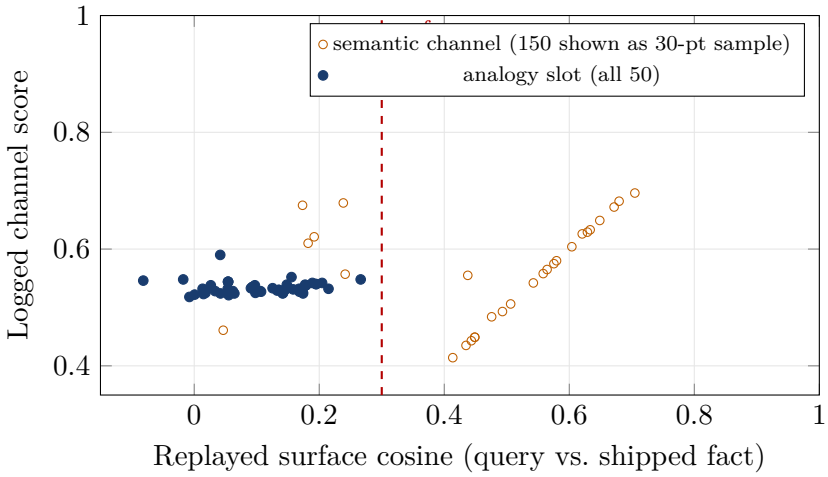

Figure 5: E2: every production analogy firing (blue), replayed. All 50 sit left of the 0.30 surface ceiling in a tight HRR band (0.518–0.590); the ordinary semantic channel (orange) lives to the right. The empty upper-left of the orange cloud and the empty right of the blue cloud are the two filters doing complementary work.

Result. All 50 analogy firings conform to the specification: replayed surface cosine mean 0.102 (s.d. 0.074), maximum 0.266, zero violations of the 0.30 ceiling; logged HRR similarities span 0.518–0.590, all above the 0.50 absolute floor. Under the identical measurement, the semantic channel's mean surface cosine is 0.288 (s.d. 0.172) and the fts channel's is 0.230. The slot is thus doing precisely what it is specified to do in production, not merely in unit tests: shipping candidates that are structurally resonant while being *less* surface-similar than either conventional channel's typical pick.

Instrument validity. The replay reconstructs the query from the ledger's stored query text, while the live vector channel embeds a rolling window of the last three messages; for 20 % of semantic-channel events the replayed cosine matches the logged score within 0.01 (single-message turns, an exact reproduction of the production computation), and the remainder deviate in the expected direction. For the analogy channel the stored query is the gate's actual input, so the replay is exact up to embedding determinism. We flag the rolling-window mismatch as the reason the semantic comparison is conservative: window context can only raise topical overlap.

8 E3: Latency and footprint

The end-to-end row composes query embedding, FTS, kNN, and graph lookup sequentially, mirroring the prefetch structure. Two design consequences are visible. First, *exact* search is the right call at this scale: brute-force cosine over 1,628 vectors costs 3.3 ms, so approximate indexing would buy nothing and cost recall. Second, the whole recall path is cheaper than a single network round trip; the substrate adds well under a tenth of a second to turn latency on hardware costing under $15/month, with the 2 GB swap file absorbing the embedder's resident set. Storage compounds equally gently: 17 kB per fact all-in at the current store size, dominated by the 8 kB HRR blob.

Operation (production host, live store)	p50	p95	unit
Graph 1-hop (in-memory DiGraph, 5 seeds)	0.026	0.120	ms
FTS5 BM25 top-8 (per query)	0.55	2.0	ms
Exact kNN top-8 over 1,628 vectors (per query)	3.3	7.6	ms
Query embedding (MiniLM, CPU ONNX)	30.6	53.1	ms
End-to-end five-block pipeline shape	**37.4**	**87.9**	ms
One-time per session: graph load	126		ms
One-time per process: embedder load	1.23		s

Table 6: Latency on 2 shared vCPUs with the agent in service. The hot path budget is dominated by the local embedder; no network call occurs anywhere in it.

9 S3: Trust dynamics

Ledger reason	Events	Mean Δ
engaged	196	+0.0200
surfaced-unengaged	203	−0.0191
Total	**399**	

Trust value (binned)	Facts
0.35 (floor)	20
0.40–0.45	11
0.50 (default)	1,157
0.55–0.75	25
0.90	1

Table 7: Left: the complete trust ledger over the window; feedback pressure is nearly symmetric. Right: the resulting trust distribution; 95 % of facts still sit at the default, and the floor cluster (20 facts at 0.35) is demoted, not deleted.

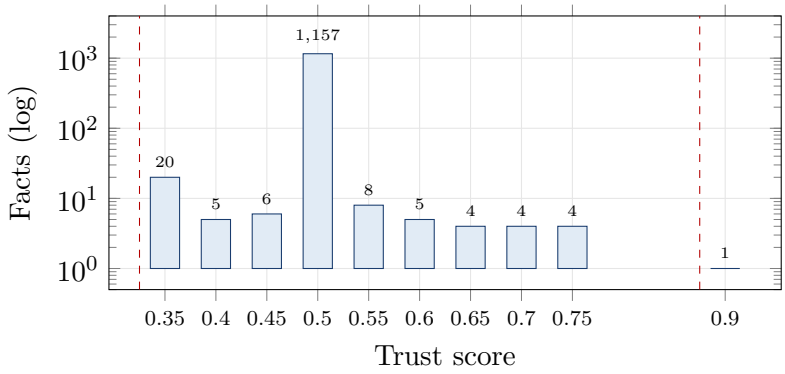

Figure 6: Trust distribution (log scale). The mass at the 0.50 default is facts that have not yet earned evidence either way; the dashed lines mark the demotion floor (0.35) and the region of the single ceiling-approaching fact (0.89). Movement away from the default requires repeated nightly evidence at ±0.02 per step.

The design forbids temporal decay, so every departure from 0.50 in Table 7 is earned by evidence. The cleanest illustration is a natural pair from the job-search workload: the live metrics fact (retrieved 156 times, repeatedly engaged) has climbed to trust 0.89 and is the store's most-trusted item, while a stale tracker fact (retrieved 105 times but never engaged since) has been demoted to the 0.35 floor. Both remain retrievable; the ranking multiplier now separates them by a factor of 2.5. This is the intended failure mode of the design: irrelevance costs rank, not existence, and a demoted fact that becomes relevant again can climb back at +0.02 per engaged night.

Aggregate feedback throughput over the window: 11 nightly runs applied 132 boosts and 157 demotions; decision-trace boosts, a governance feature that credits facts cited in accepted recommendations, have fired on 0 facts because both logged decisions aged out unresolved. We report this as a feature that exists but is empirically unexercised (§12).

10 S4: Abstraction and the operator gate

Compounding, as distinct from accumulation, requires the store to grow *upward* (summaries, categories, episodes) and not only outward. The abstraction stage has run twice in the window and the results are measurable, small, and real:

- **Chunks:** 9 chunk facts summarize 52 member facts (cluster sizes 3–11). Members are never deleted; the read path's chunk suppression ships the summary instead of the members when both surface, and an explicit tool expands a chunk on demand.
- **Episodes:** each chunk creates an `episode` node in the graph (9 live), linking the summary into entity neighborhoods; 74 dated `event` nodes from calendar sync give the graph a temporal spine (windows of $-1/+60$ days, stale futures pruned).
- **Emergent categories:** 15 per-category HRR banks exist, of which the long tail (single-fact banks like finance, tool, data) reflects category proposals awaiting consolidation rather than design intent.
- **The gate holds:** 78 proposed edges (mostly `INSTANCE_OF` promotions) and 60 alias-candidate edges sit in the review queue; none are visible to the runtime graph. The weekly review nudge is the mechanism by which they either enter the ontology or die.

The honest summary is that abstraction is the youngest layer: two production runs, single-digit chunk counts, and a review queue that is accumulating faster than the operator clears it. The mechanism demonstrably works end to end; its compounding value at scale is a forecast, not yet a measurement.

11 Discussion

The substrate thesis, memory edition. The companion Little Bear program concluded that for operational decisioning, the model is the interface and the substrate is the product. This study lands the same thesis one layer down: recall quality here is governed by store structure (canonical entities, closed relations, trust weights, gated ontology), not by model choice. The only LLM in the memory system is a commodity distiller on the asynchronous write path; swap it and the read path does not change. Every per-turn operation is local, deterministic, and instrumented.

What the analogy result does and does not show. E2 establishes conformance: the production slot obeys its specification on all 50 firings, and its picks are measurably unlike ordinary semantic recall. It does not establish *usefulness*: whether the shipped analogies changed downstream behavior is a question the current ledgers cannot answer (the engagement signal does not distinguish which block earned the engagement). Closing that loop, per-block engagement attribution, is the single highest-leverage instrumentation gap this study surfaced.

Why the moderate E1 numbers are the credible ones. A characterization that reported 95 % recall on self-generated queries against a store that is 62 % near-duplicate email would deserve suspicion. The 47.5 % single-channel R@5, the channel complementarity, and the identified circularity (gists share an author with their sources) are consistent with what a

compressed-paraphrase probe against a production store should yield, and they motivate the concrete fixes in §13 rather than a victory lap.

12 Threats to validity

First, this is a single-subject, single-host observational study; nothing here estimates variance across users, workloads, or hardware. *Second*, nothing was pre-registered; the defensible core is conformance of production behavior to configuration thresholds committed before the window, plus deterministic seeded replays. *Third*, E1's queries are LLM gists that share an author model with the facts they target, which plausibly inflates the lexical channel; the best-of-either row is an oracle upper bound, not a system. *Fourth*, E2's semantic-channel comparison reconstructs queries from stored text rather than the live rolling window (exact for the analogy channel, conservative for the comparison). *Fifth*, the telemetry is self-instrumented by the system under study; a bug in the ledger writer would corrupt both behavior and evidence, though the 20 % exact-reproduction subset in E2 cross-checks logger against replay. *Sixth*, the engagement heuristic (entity mention or token-Jaccard ≥ 0.3) is a proxy with unknown precision against true usefulness, and decision-trace boosts, the designed stronger signal, have zero production activations. *Seventh*, coverage is known-incomplete: 25 failed ingest batches are absent from the store. *Eighth*, no comparison against an external baseline (hosted memory API or plain RAG over raw text) has been run; the internal channel comparisons do not substitute for one.

13 Recommended next steps

In order of leverage: (1) per-block engagement attribution in the feedback stage, so the analogy slot's downstream value becomes measurable rather than arguable; (2) a rank-fusion pass (RRF) in prefetch to convert E1's oracle gap (+6.5 R@5) into realized recall, at zero added latency budget; (3) an ingest-failure retry lane to close the 25-document coverage gap; (4) operator review-queue burn-down (138 gated edges) followed by a re-run of the S4 census to measure ontology acceptance rates, mirroring the decision-acceptance instrument the Little Bear program used; (5) an external-baseline experiment (plain RAG over the same corpus) using the E1 harness unchanged.

14 Conclusion

A five-channel hybrid memory substrate, holographic facts, semantic vectors, a governed knowledge graph, an analogy slot, and situation playbooks, runs in production inside one 16.3 MB SQLite file on a 2 GB commodity host, adds 37 ms median to turn latency with zero hot-path API calls, and holds to its written specification under full replay: 50 of 50 production analogy firings inside the disagreement filter's committed thresholds. Forgetting is use-driven and reversible; ontology growth is machine-proposed and operator-committed; provenance and telemetry are first-class. The store compounds: facts become chunks, chunks become episodes, episodes join the graph. What remains open is measured too: moderate known-item recall with a quantified fusion headroom, an unexercised decision-trace loop, a young abstraction layer, and a review queue awaiting its operator.

Practitioner one-liner

> *The Backbone memory substrate is a working demonstration that compounding agent memory does not require a vector-database bill: 1,214 trust-scored facts*

with 100 % holographic coverage, five recall channels in one SQLite file, 37 ms median recall on two shared vCPUs, a production-verified analogy mechanism (50/50 firings within specification), and an ontology that only grows through an operator's hands: all on a $12/month droplet, with every claim in this report replayable from the checked-in dataset.

Reproducibility

All instruments are read-only against the production store and are checked into the `hermes-memory-study/` directory of this repository alongside the extracted dataset (`dataset.json`) and this document's source: `bench_memory.py` (composition, telemetry, microbenchmarks), `bench_fix.py` (graph benchmarks, E2 replay), `bench_knownitem.py` (E1, fixed seed 14). The public representation of the memory system (patches, plugin source, architecture documentation) is maintained at github.com/sheldon904/hermes-hybrid-memory; the public snapshot trails the live install by one governance phase (decision traces and the ontology review gate are not yet extracted), a divergence we disclose here rather than paper over. Live-store row counts, log line counts, and configuration constants were captured on 2026-07-17 and are reproducible from the dataset without store access.

Acknowledgments

The author thanks the NousResearch Hermes project for the upstream agent and holographic memory base, the sqlite-vec project for the embedded vector index, and the Bay West Labs operations workload for supplying two months of unforgiving production traffic.

References

[1] W. White. *Augmented Operational Decisioning with Ontology-Grounded Local Agents: A Fourteen-Run Empirical Study Against Real Construction-Operations Data.* Bay West Labs, Little Bear Foundry Research, May 2026.

[2] D. Hofstadter. *Analogy as the Core of Cognition.* In *The Analogical Mind*, MIT Press, 2001.

[3] T. Plate. *Holographic Reduced Representations.* IEEE Transactions on Neural Networks, 6(3), 1995.

[4] NousResearch. *Hermes Agent* (v0.18.0). `https://github.com/NousResearch/Hermes-Agent`, 2026.

[5] A. Garcia. *sqlite-vec* (v0.1.9). `https://github.com/asg017/sqlite-vec`, 2024.