

The Un-Brain: A local-first, indelible knowledge multi-graph for private AI memory with deterministic, owner-adjudicated contradiction handling

Author: Nelu Lazar **Affiliation:** Nehloo Interactive LLC · Nehloo Foundation, Inc. **ORCID:** 0009-0008-5548-4109 **Status:** Preprint · contact email: nelu.lazar@nehloo.org **Artifacts:** @nehloo/graphnosis (Apache-2.0, github.com/nehloo/Graphnosis); graphnosis-secure-sync (source-available, github.com/nehloo-interactive/graphnosis-secure-sync). **DOI:** 10.5281/zenodo.20843387

Abstract

Large language models reason fluently but do not remember: the transformer carries no state between sessions, so the burden of memory falls on the human — who, under load, forgets, and worse, holds contradictory beliefs without noticing. Statelessness has a virtue — nothing to leak between sessions — and Graphnosis keeps it: memory stays local and private, so persistence need not cost privacy. We present **Graphnosis**, a local-first knowledge multi-graph for private AI memory that treats persistence not as a transcript store but as an *un-brain*: a structure that keeps the cognitive virtues of human memory (association, reinforcement, recall) while refusing two failure modes — silent decay, and the silent resolution of contradictions.

Its defining behavior is an **adjudication contract**: when a new statement conflicts with one already stored, Graphnosis does not overwrite or auto-resolve it — it records the conflict with provenance, surfaces both sides, and leaves the verdict to the owner. *You* decide what the ground truth is. This is a guarantee of design, computed deterministically with no model in the loop, with a two-stage precision/recall design (a high-precision ingest pass and a recall-oriented reflection pass) gated by a deterministic triage; in the audited deployed systems with documented contradiction handling, automatic resolution or non-total surfacing was the common pattern; this surface-don’t-resolve contract — every detected conflict recorded in an inspectable lane, never silently resolved — is one Graphnosis shares with concurrent work (Tenure reaches it independently), and combines with the dual-graph substrate and deterministic guarantees below (§14).

The adjudication contract rests on a substrate whose guarantees are stated and proved rather than left implicit. Memory is a **dual graph** — directed (ordered) and undirected (associative) edges as co-equal typed classes over the *same* nodes, each independently queryable rather than fused into one ranking, unlike the multi-relational graphs we surveyed (§5.2). Retrieval is a single traversal across both edge classes that scores each node by its strongest evidence path, never a sum; we prove this makes recall **hub-independent** (Theorem 3, §7), surfacing the strongest evidence rather than the most-connected node. The store is **architecturally indelible** (Theorem 1), converges under an **order-independent merge** (Theorem 2), and is constructible with **zero embedding or generation API calls** — running on commodity hardware, fully air-gapped; the cortex is local-first, end-to-end-encrypted, and portable across MCP clients (§14). The highest-accuracy benchmark configuration layers optional cloud components on top (§12.2).

On the public LongMemEval_S benchmark (500 questions, official methodology, GPT-4o judge), Graphnosis scores **78.00%** cloud-paired, **64.60%** with zero embedding-API calls, and **41.60%** fully on-device (local Llama 3.2 3B, no cloud calls of any kind); Zep reports **71.20%** on the same benchmark under its own harness — a same-benchmark juxtaposition, not a controlled comparison (§12.3). The answer model is the dominant accuracy lever; the contribution is the *architecture*, isolated by two ablations that hold the answer model fixed: the dual graph is worth **+13.2** points over flat top-k for GPT-4o and **+5.8** on-device for the 3B model, and the hub-independent scoring rule (Theorem 3) accounts for **+6.0** of the former. We further demonstrate portability: one local, end-to-end-encrypted store served eight AI clients over the Model Context Protocol without re-ingestion. Contradiction detection is deterministic and model-free; we report its operating characteristics as evidence that the contract holds — not as the contribution itself.

1. Introduction

An assistant’s memory is only as trustworthy as its worst silent failure, and the systems we build to give language models long-term memory inherit two from the brain they imitate. The first is **silent decay**: facts are evicted, summarized away, or buried beneath newer ones until they are effectively gone, with no record that anything was lost. The second, subtler and more corrosive, is **silent contradiction resolution**: when a new statement disagrees with a stored one, the system quietly overwrites it, averages the two, or picks a winner — and the owner is never told their memory just changed its mind. A memory that silently rewrites itself is more dangerous than no memory at all, precisely because it is trusted. The transformer offers no help here: it carries no state between conversations, so memory management is delegated to the human user, who is a poor store under load — forgetting facts, and holding conflicting beliefs reconciled *silently* through cognitive bias into a single confident answer that may be wrong.

The adjudication contract is the behavior we ask the reader to hold onto. When a new statement conflicts with one already stored, Graphnosis does not overwrite it, average the two, or pick a winner: it records the conflict as a first-class, provenance-bearing relation, surfaces both sides at recall time, and stops. The resolution belongs to the owner, not the system — *you decide what the ground truth is*. This is a guarantee of **design**, not accuracy: there is no code path by which Graphnosis silently resolves a contradiction. It is the defining move of what we call the **un-brain** — a memory that keeps the brain’s cognitive virtues (association, reinforcement, instant recall) while refusing its two silent failures: it never deletes attested content, and it never decides, on the owner’s behalf, which of two conflicting memories is true. We instantiate it as Graphnosis, a local-first system the user owns end to end.

This is deliberately **not retrieval-augmented generation**. Vanilla passage-retrieval RAG retrieves passages from a document corpus to condition a single generation; by itself it holds no evolving knowledge state, does not notice when two retrieved passages disagree, and carries nothing across calls (stateful RAG and agent-memory hybrids add some of this). The un-brain is the inverse: a *persistent, owner-held knowledge state* — typed, indelible, synchronized across devices, and actively checked for contradictions — that any model may query but none owns. Retrieval is one operation over that state, not its definition; the substrate, its guarantees (indelibility and conflict-free merge, Theorems 1–2; hub-independent retrieval, Theorem 3, §7), and its consistency primitive (§8) are the contribution.

That inversion is also where the project began. The founding question was whether knowledge could be encoded for its *consumer* — an AI traversing structure — rather than for a human reading a document: representations “no human can read, but AI can,” on the premise that prose, transcripts, and PDFs are lossy and noisy for machine consumption. The canonical layer is accordingly a machine-native encrypted binary (.gai, §15), and knowledge is a typed multi-graph the model traverses — directed and undirected edges as co-equal classes over the *same* nodes (§5.2), the founding idea of the project — rather than a document a person scans — yet the owner inspects and governs every memory through the application, never losing access to it. There is empirical motivation for that bet beyond ergonomics: language models read long inputs unevenly, attending most reliably to material at the beginning or end of a context and degrading on what falls in the middle — the U-shaped *lost-in-the-middle* effect [Liu et al. 2024] — so a long document is not only lossy to encode but poorly positioned to be read, whereas a small, dense, typed subgraph (the bounded $\leq 2,000$ -token excerpt of §4) presents the model no long middle in which the relevant fact can sink.

A single principle organizes the system: **determinism first**. Every guarantee that bears on correctness — indelibility, contradiction *surfacing*, conflict-free merge — is computed deterministically, with no model in the loop. The optional local-LLM and neural-network layers are additive, *detachable* overlays — physically separate from the attested store (§5.1) — that enrich the cortex with predicted connections and synthesized understanding, making it a more useful working memory, yet can never alter an attested memory and can be removed at any time, leaving the deterministic core intact. Benchmark accuracy is governed by whichever answer model a user pairs; the contribution is the deterministic, owner-held substrate beneath it.

1.1 Contributions

We separate **research contributions** (empirically testable or formally provable claims) from **system and engineering contributions** (the deployable substrate).

Research contributions.

1. **Deterministic, owner-adjudicated contradiction detection** (§8) — an active, three-path consistency primitive, with no model in the loop, that surfaces conflicts for the owner to adjudicate rather than resolving them. The closest systems either ignore contradictions or auto-resolve them — with an LLM (Mem0 historically, Zep) or by deterministic strategy (Semantica); the nearest on the surfacing axis, Atlas, routes *strategic* conflicts to a human queue but still auto-resolves routine ones. In the audited deployed systems with documented contradiction handling, auto-resolution or non-total surfacing was common — an argued observation over the systems we examined, not a uniqueness claim; concurrent research reaches the same *total* surface-don’t-resolve contract independently (Tenure, §14), so we present it not as ours alone but as one element of the combination below, with conceptual roots in truth-maintenance and belief-revision theory.
2. **A two-stage precision/recall design for contradiction triage** (§8) — high-precision ingest-time detection plus a recall-oriented reflection pass, with a deterministic admission filter, so the active queue stays legible without silently discarding signal.
3. **Node-set indelibility, formalized** (Theorem 1, §5) — the worst case for any memory is reduced retrieval prominence, never erasure.
4. **Last-writer-wins determinism under a strict total order** (Theorem 2, §11) — replicas converge to the same state regardless of operation arrival order.
5. **Hub-independent retrieval, formalized** (Theorem 3, §7) — recall scores each node by its single strongest evidence path, never a sum, so node degree confers no score and a popular-but-irrelevant node cannot crowd out the answer; an ablation attributes **+6.0** points of the graph’s benchmark advantage to this rule specifically (§12.4).
6. **Measured dual-graph coverage** (Observation 1, §5.2) — treating directed (ordered) and undirected (associative) edges as co-equal classes over the *same* nodes covers largely-disjoint retrieval paths: only **8–13%** of directly-connected node pairs are joined by both classes, and the two are functionally complementary under traversal, so a single-relation store misses on the order of half the graph’s direct connections. The property is empirically testable and reproducible with a deterministic harness, and a structure-versus-scoring ablation attributes **+7.2** of the graph’s **+13.2**-point retrieval advantage to the dual structure itself, separately from the **+6.0** earned by the scoring rule of contribution 5 (§12.4).

Contributions 3–5 are guarantees enforced *by construction*, not deep results; we formalize them because a memory substrate should be held to them — and because they must survive concurrency and end-to-end sync — not because the proofs are hard. Contribution 6 is empirical rather than provable: a measured, reproducible property of the deployed graph. We further note the representation as one we did not find kept *co-equal* elsewhere in our survey — directed and undirected edges as two *co-equal, independently-queryable* classes over one node set, where the multi-relational graphs we surveyed pool or fuse their edge classes into a single ranking (HippoRAG’s undirected PageRank graph; the learned and weighted traversals of MAGMA and HAGE; §5.2) — though the contribution’s weight rests on the measurement, which holds whatever the priority.

System / engineering contributions. Built on that dual-graph substrate (research contribution 6): a strict promotion gate between predicted and attested knowledge (Design Invariant 1, §5); a zero-API construction pipeline; **graph-derived query expansion** that mines synonyms from the cortex’s own similarity edges, with no thesaurus or embedding model (§7); a small feed-forward link predictor (the GNN-LP, §10), CPU-trainable with no GPU or Python; a five-tier determinism taxonomy applied to every operation (§4); and a **portable, owner-held, end-to-end-encrypted** memory reachable by any MCP client and synchronized across the owner’s devices (§11, §15).

1.2 Why local-first is essential

A cloud memory the user cannot inspect or hold is, for sensitive work, not an option but a liability. Three settings make the requirement concrete: regulated professionals (legal, clinical, financial) whose data cannot leave controlled infrastructure; air-gapped environments (classified networks, field deployments) where any API call is simply unavailable; and individuals who decline to surrender their personal record to a third party. Graphnosis is built so that the substrate never leaves the owner’s machine, and so that the AI client

receives only a small curated subgraph per query, never the store itself (§4).

2. The Owner-Held Memory Substrate

Although the benchmark and running examples in this paper use the individual case, Graphnosis is not limited to it: the same architecture serves a spectrum of *owners* — an individual’s second memory; a team’s institutional memory; a regulated organization’s auditable knowledge base; a researcher’s evidence corpus; a community’s cultural archive — unified by one non-negotiable constraint: the data stays on the owner’s own infrastructure. We therefore describe Graphnosis as a *local-first knowledge graph for private AI memory*, reserving “personal” for the individual case. The breadth is architectural, not aspirational: this paper presents direct empirical evidence for the individual/benchmark case and articulates the others as inheriting the same design.

A longer-horizon application — an external, auditable record of identity for people with neurodegenerative disease such as Alzheimer’s, built over years by patient and caregivers — motivates the indelibility requirement especially sharply, but it requires clinical validation and regulatory frameworks out of scope here; we note it as a direction, not a result.

3. The Un-Brain: Design Principles

Human memory has virtues worth keeping and constraints worth discarding. The un-brain keeps the virtues and inverts the constraints.

Indelibility — strengthen, never weaken. An attested canonical memory is retained under the write semantics below, and does not decay from disuse. The only events that lower a memory’s confidence are explicit *correctness* events — a contradiction detected, a supersession, a user edit — all audited and reversible. Within the canonical layer, an attested node is not hard-deleted under these write semantics; prominence, by contrast, is earned. What changes over time is not whether an attested memory exists but how prominently it surfaces.

The defining un-brain act — surfacing what the brain hides. The brain’s most consequential failure is not forgetting; it is that, faced with two conflicting beliefs, it resolves the conflict *silently* and hands consciousness one confident answer. The conflict is never seen; a false memory feels exactly like a true one. The un-brain inverts this: when two memories share an entity context yet assert opposing things, it preserves both, flags the pair, and routes it to the user for adjudication — deterministically, with no model guessing in the loop (§8). Crucially, the check fires at the moment of *ingestion*: the instant a new memory is saved, it is tested against what the un-brain already holds, so a conflict surfaces right then — not lying dormant until someone happens to query it. This is what separates a *memory* from a *store*: a store returns what you put in; a memory notices, as you put it in, when it disagrees with itself. We therefore treat contradiction surfacing as a first-class capability, not a background hygiene task.

Associative reinforcement. Connections that are used grow stronger; those unused remain present but quieter. Reinforcement is Hebbian and bounded (Proposition 1, §5.3): edge and node weights converge to a saturation ceiling rather than growing without limit.

These principles are deliberately *brain-inspired, not brain-faithful*: they borrow mechanisms cognitive science has long described — Hebbian co-activation [Hebb 1949], the spreading activation of semantic memory [Collins & Loftus 1975], and the recency-and-frequency activation of rational memory models [Anderson & Schooler 1991] — and *invert* the constraints that make biological memory lossy, above all the exponential forgetting curve [Ebbinghaus 1885]. We cite these as design inspiration, not as claims about neural mechanism. Figure S4 (Additional Material) contrasts that curve with the un-brain’s strengthen-on-recall inversion.

Human memory	Its failure mode	The un-brain inversion
Exponential decay [Ebbinghaus 1885]	important facts fade if unrehearsed	strengthen-never-weaken; the decay function is disabled (Prop. 1)
Silent conflict resolution	a false memory feels identical to a true one	the conflict is <i>surfaced</i> for the user, never auto-resolved (§8)
Hebbian use-it-or-lose-it [Hebb 1949]	unused traces weaken	Hebbian strengthening only, bounded by saturation (Prop. 1)
Sleep-dependent consolidation	encoding is offline and lossy	deterministic encoding, no offline loss
Source amnesia	one rarely recalls <i>when</i> or <i>whence</i> a fact came	full provenance + op-log + <code>recall_as_of</code> (§5.1, §11)

4. System Overview

Graphnosis is implemented in TypeScript on Node.js and distributed as an open-source library; an encrypted-sync layer is a separate, source-available package. A desktop application wraps the library and exposes the store to AI clients over the open Model Context Protocol — a stdio transport for desktop/IDE clients and an HTTP/SSE transport for others. **For AI clients, the store is reachable only through this protocol boundary, which is where the privacy guarantee for that path is enforced: a client receives only a curated per-query subgraph (capped at $\leq 2,000$ tokens), never the cortex itself. This is the AI-client threat model; it does not by itself speak to OS-level compromise, device backups, or the encrypted-sync path.**

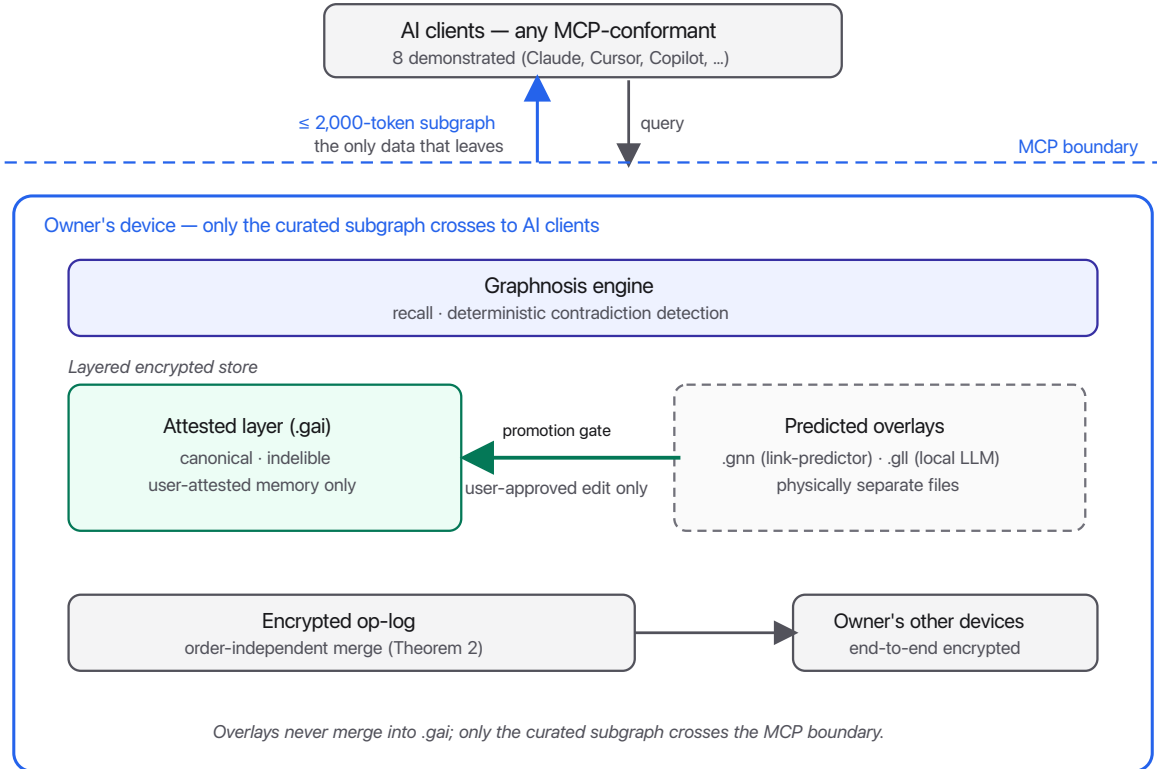


Figure 1. Graphnosis architecture and the privacy boundary. AI clients connect over MCP; only a curated

$\leq 2,000$ -token subgraph crosses the boundary, never the store. On the owner’s device, the engine serves an attested canonical layer (`.gai`) kept physically separate from predicted overlays (`.gnn/.g11`); exactly one user-approved operation crosses the *promotion gate* from predicted to attested (Design Invariant 1, §5.1). An encrypted op-log synchronizes end-to-end to the owner’s other devices under an order-independent merge (Theorem 2, §11).

Key terms. A *cortex* is the user’s full set of encrypted graphs. An *engram* is one graph with its own sensitivity tier. A *subgraph* is the small excerpt delivered to an AI client per query. The graph itself is layered (§5): a canonical attested layer, plus predicted overlays kept physically separate.

Determinism taxonomy. Every operation is labelled with one of five determinism tiers, and the tier is reported to the AI client in each tool’s description.

Tier	Guarantee	Examples
Deterministic	Identical input $\rightarrow$ identical result; no LLM, no randomness; auditable	recall, structured queries, source/engram operations, contradiction detection
Approximate	Similarity scan, no LLM, but results may vary across runs due to LSH randomness	near-duplicate detection (<code>audit_memory</code> , <code>check_duplicate</code>)
Conditional	Deterministic by default; non-deterministic only when an optional local LLM or the neural overlay is enabled, with the path reported	correction/edit, skill training
Mixed	Deterministic core with a clearly separated inferred overlay	recall when neural/LLM overlays intersect the result
Non-deterministic	Requires a local LLM; output varies	strategic synthesis, predictions, distillation

The Approximate tier is small but distinct: it covers exactly the two similarity-scan tools, which are LSH-based and therefore neither LLM-driven nor bit-for-bit reproducible.

5. Knowledge Representation

5.1 The multi-layer graph

A Graphnosis graph is a tuple of three physically separate layers,

$$G = (G_{\text{canonical}}, G_{\text{gnn}}, G_{\text{llm}}),$$

where $G_{\text{canonical}}$ (`.gai`) holds every memory the user attested or that an AI client saved on their behalf; G_{gnn} (`.gnn`) holds edges predicted by a local link predictor; and G_{llm} (`.g11`) holds edges and assertions synthesized by an optional local LLM. The two overlay layers are never merged into the canonical layer.

Within a layer, nodes carry a type (fact, entity, document, section, ...) and edges carry a type and weight, directed (causal, temporal, **supersedes**, depends-on) or undirected (similar-to, shares-entity). Because a single node pair may be joined by both a directed and an undirected edge, each layer is a *multi-graph* (parallel typed edges) in the graph-theoretic sense; the store is thus a multi-graph within each layer, across the three physical layers, and across the owner’s federated engrams (§2).

Theorem 1 (Node-Set Indelibility). *Let N_t be the set of node identifiers present in the canonical layer after the first t write operations. Under the canonical-layer write semantics — every write is either an append (insert a new node) or a soft-delete (a tombstone that lowers a node’s confidence to a floor but retains it) — the node set is monotone non-decreasing: $t \leq t' \implies N_t \subseteq N_{t'}$.*

Proof. By induction on t . The $(t+1)$ -th operation is of one of two kinds. An *append* introduces a fresh identifier, so $N_t \subseteq N_{t+1}$. A *soft-delete* (**forget**, supersession) lowers a node’s confidence and retrieval priority but removes no identifier, so $N_{t+1} = N_t$. No write path deletes from the identifier set, so $N_t \subseteq N_{t+1}$ for every t ; by transitivity $N_t \subseteq N_{t'}$ for all $t \leq t'$. ■

The consequence is the design claim: unlike biological memory, under canonical-layer write semantics an attested node in Graphnosis is never hard-deleted; the worst case for any memory is quieter retrieval, not erasure.

Time-travel recall (`recall_as_of`). A direct consequence of indelibility: because canonical-layer nodes are not hard-deleted under these write semantics and the synchronization op-log (§11) is a complete append-only history, the canonical graph is reconstructible *as of* any past sequence point or timestamp. The store can answer “what did I believe on this date?” without a separate versioning system — indelibility is what makes time-travel recall well-defined — and the capability is exposed to clients as a first-class `recall_as_of` operation.

Design Invariant 1 (The Promotion Gate). Predicted content (`.gnn`, `.gll`) and attested content (`.gai`) are separated not merely as disjoint sets but as a property of the write-path lattice: exactly one user-gated operation (an approved **edit**) crosses the boundary from predicted to attested, and no inference loop can. We present this as an invariant enforced by construction — the file boundary is the on-disk expression of it — rather than as a theorem, because proving set-disjointness would restate the premise. As models grow more capable of proposing facts, this boundary becomes an architectural requirement, not a UX nicety. The layered write-path and this gate are diagrammed in Figure S2 (Additional Material).

5.2 Dual-graph coverage

Observation 1 (Dual-Graph Coverage). A typed graph that carries both directed and undirected edges within a layer covers retrieval paths a single-relation store misses — a supersession chain versus an entity-overlap cluster. On graphs built from the public LongMemEval corpus (15k–45k nodes), the two edge classes are largely disjoint: of all directly-connected node pairs, only **8–13% are linked by both** classes, the rest splitting between directed-only and undirected-only — so a single-relation store misses on the order of half the graph’s direct connections, whichever class it keeps. The classes are also functionally complementary under traversal: at the ≤ 3 -hop depth recall uses, directed edges yield short ordered chains (median reach ≈ 3 nodes) while undirected edges yield broad associative neighborhoods (median reach 53–90), and their union reaches strictly more than either alone (64–107). Coverage strictness is a measured property of the deployed graph, reproducible with a deterministic harness that ships in the source tree. The associative half of that structure is itself a capability: because undirected edges make shared-entity and co-occurrence relationships explicit, traversal surfaces non-obvious connections between memories — associations a human under load might not draw unaided.

In our survey we did not find this representation kept *co-equal and independently queryable* elsewhere — and the claim is worth stating precisely, so it reads as neither weaker nor stronger than it is. It is not that Graphnosis carries two edge types; many systems do. It is that directed and undirected edges are kept as *co-equal, independently-queryable* classes over one node set — both first-class traversal paths, joined in a single recall — rather than one class being collapsed, pooled, or demoted to an auxiliary set. The measured coverage above (Observation 1, research contribution 6 of §1.1) is what the contribution rests on, and what gives the choice its bite: the two classes are largely disjoint and complementary, so keeping both first-class adds reach on our corpus that a single-relation store does not provide without re-encoding both classes itself. Surveyed graphs are either directed entity-relationship stores (Mem0^g, Zep/Graphiti, Semantica), graphs collapsed to undirected for clustering or PageRank (GraphRAG’s Leiden graph; HippoRAG’s PPR graph), or multi-relational graphs that keep several edge classes but fuse them into one learned or weighted traversal rather than leaving each independently queryable (MAGMA and HAGE; §14). HippoRAG is the nearest relative and the overlap is genuine: it places relation and synonymy edges over the same phrase nodes. The difference is one of *use*, not vocabulary — HippoRAG pools both into a single graph and runs Personalized PageRank over their union, with synonymy as an auxiliary edge set feeding one ranking, whereas Graphnosis keeps the two as distinct, independently-queryable edge types — directed for ordered relations (**supersedes**,

precedes, causes, depends-on), undirected for association (**similar-to**, **shares-entity**, **co-occurs**) — and reports their coverage as the largely-disjoint structure measured above. The full taxonomy — **sixteen directed and seven undirected types**, grouped by function — is catalogued in Appendix B; two of the directed types, **contradicts** and **supersedes**, are the contract’s own machinery (§8), so the representation literally contains the behavior the system is named for. A side-by-side of this representation against the fused (MAGMA, HAGE) and directed-only (Cortex, Atlas) encodings is given in Figure S1 (Additional Material).

5.3 Bounded reinforcement

Proposition 1 (Saturation Convergence). *Co-recall reinforcement applies the capped additive update $w_{n+1} = \min(c, w_n + \delta)$ with fixed increment $\delta = 0.03$ and ceiling c ($c = 0.95$ for node confidence; edge weights use the same form with $c = 1.0$). The weight sequence is monotone non-decreasing, bounded above by c , and reaches c in at most $\lceil (c - w_0)/\delta \rceil$ reinforcements, never exceeding it.*

Proof. From the update, $w_n \leq w_{n+1}$ (a reinforcement never lowers a weight) and $w_{n+1} \leq c$ (the cap), so (w_n) is monotone and bounded above, hence convergent. Once $w_n \geq c - \delta$ the cap binds and $w_{n+1} = c$; since every earlier step adds exactly δ , the ceiling is reached after at most $\lceil (c - w_0)/\delta \rceil$ steps. ■

Reinforcement therefore strengthens monotonically and *saturates* — it cannot grow without bound — and, because no opposing decay acts on attested content, a saved memory’s trust only rises or holds. The temporal-decay function in the code ($w \leftarrow \max(\underline{w}, \gamma w)$ with $\gamma < 1$ and floor $\underline{w}$) is dormant by product design, gated on an **ephemeral** marker reserved for a future ambient-capture path; it never touches user-attested memories.

6. Model-free Construction

Ingestion calls no embedding or generation API. Documents, PDFs, notes, and web clips are read locally; entities and relationships are extracted with deterministic analyzers; and the result is indexed into an encrypted binary **.gai** file. The pipeline is source-agnostic: anything reducible to text — documents, PDFs (OCR’d locally), notes, web clips, structured logs, database exports, sensor and IoT JSON, mail archives — is ingested and indexed the same way, and a local webhook endpoint generalizes this to any device or script that can post text; ingestion imposes no schema and never leaves the device. The construction cost in external API calls is zero, which is what makes air-gapped operation possible and what isolates the memory layer from answer-model cost. (Optional dense embeddings — via a cloud adapter, or a local model that runs entirely on-device through ONNX Runtime — can be enabled for higher retrieval quality; they are not part of construction and are reported separately in §12.)

7. Retrieval

Recall transforms a query into a small ranked subgraph: TF-IDF cosine over the lexical index, breadth-first traversal over typed edges, and router-set seed budgets (typically 20–50 nodes within an ~8,000-token answer context) so the AI receives structured, actionable knowledge rather than a document to reparse. Traversal is **time-aware**: a node’s score is modulated by recency of last access, access frequency, and confidence, and a fact whose validity window (**validUntil**) has closed is *down-weighted rather than dropped* — a superseded fact still surfaces, but beneath its replacement, consistent with strengthen-never-weaken (§3). Directed **precedes** edges preserve event order within a session, and a temporal-intent classifier routes time-sensitive questions (“what did I decide *before* X?”) to a traversal strategy that favors them. This time-aware retrieval is the retrieval-side mechanism behind the temporal-reasoning results in §12.2. An optional embedding index adds a hybrid lexical+dense retrieval path; it runs against a cloud adapter or, by default in the desktop application, a **bundled on-device model** (BGE-small-en-v1.5 via ONNX Runtime, ~30 MB), so dense semantic retrieval is available with nothing leaving the device. An optional local LLM can enrich the query at recall time (a server-side rewrite, surfaced to the user, that never writes to the graph). Every recall is logged.

Definition 1 (Temporal retrieval score). A node v ’s retrieval score is its base traversal score scaled by a deterministic temporal multiplier,

$$\text{score}(v) = \text{base}(v) \cdot r(v) \cdot f(v) \cdot \text{conf}(v) \cdot e(v),$$

where $\text{base}(v)$ is the breadth-first seed/traversal score (decayed per hop), $\text{conf}(v) \in [0, 1]$ is the node’s confidence, and the recency, frequency, and expiry factors are step functions of stored access metadata:

$$r(v) = \begin{cases} 1.3 & \Delta_{\text{acc}} < 1 \text{ day} \\ 1.1 & \Delta_{\text{acc}} < 7 \text{ days} \\ 1.0 & \text{otherwise} \end{cases} \quad f(v) = \begin{cases} 1.2 & \text{acc} > 10 \\ 1.1 & \text{acc} > 3 \\ 1.0 & \text{otherwise} \end{cases} \quad e(v) = \begin{cases} 0.3 & \text{now} > \text{validUntil} \\ 1.0 & \text{otherwise} \end{cases}$$

with Δ_{acc} the time since last access and acc the access count. Recent, frequently recalled, high-confidence facts surface preferentially; an *expired* fact is down-weighted by 0.3, never dropped — the retrieval-side expression of strengthen-never-weaken (§3). Every factor is a deterministic function of stored metadata; no model is consulted. The recency-and-frequency shape deliberately mirrors the base-level activation of rational memory models [Anderson & Schooler 1991] and the spreading activation of semantic memory [Collins & Loftus 1975] — inspiration for the scoring, not a claim of neural mechanism.

Theorem 3 (Hub-independence of retrieval). *Let $\text{base}(v)$ be the breadth-first traversal score of Definition 1, before the temporal multiplier. The traversal writes a node’s score only through the guarded update $\text{base}(v) \leftarrow \max(\text{base}(v), x)$ for a candidate path score x ; hence*

$$\text{base}(v) = \max_{p \in P(v)} s(\text{seed}_p) \prod_{e \in p} \delta w_e \gamma_e,$$

a maximum over the seed-rooted paths $P(v)$ the traversal explores — per-hop decay $\delta = 0.6$, edge weight w_e , and $\gamma_e = \frac{1}{2}$ for a backward-directed step, 1 otherwise — never a sum. Consequently $\text{base}(v)$ is invariant to the multiplicity of edges incident on v : adding edges into v whose induced path score does not exceed $\text{base}(v)$ leaves it unchanged. Node in-degree confers no score; no node is promoted by edge count alone.

Proof. The only writes to a node’s score are guarded by $x > \text{base}(v)$, so the stored value is the running maximum of every candidate offered for v , never their sum; each candidate equals the dequeued score times $\delta w_e \gamma_e$, which by induction on hop count equals the path product for the extended path. Adding in-edges only offers further candidates, which the maximum discards unless strictly larger. $\square$

This is the structural reason a popular node cannot swamp recall. Additive (spreading-activation / PageRank-style) accumulation would let a high-degree but irrelevant node gather score in proportion to its degree; the max rule cannot, so retrieval surfaces the single strongest evidence chain rather than the best-connected node. The effect is measurable: replacing only this rule with summation — graph, seeds, and answer model held fixed — costs **+6.0 points** end-to-end on LongMemEval (§12.4), nearly half the dual graph’s advantage. Two caveats bound the claim precisely. The final score multiplies $\text{base}(v)$ by the per-node temporal factor of Definition 1, a topology-independent scalar that preserves degree-independence. And because the traversal expands each node once, at its first dequeue, $P(v)$ is the explored path set, so $\text{base}(v)$ is an order-dependent lower bound on the global maximum over all paths; the no-sum and degree-independence properties hold exactly regardless — only global path-optimality is heuristic. Recall is in this sense stateful (it reads wall-clock time and updates each surfaced node’s access metadata), so the determinism this paper claims is of construction (§6) and merge (Theorem 2), not of ranking; the benchmark configurations are stateless — one query per freshly built graph — so no reinforcement compounds.

Graph-derived query expansion. Before retrieval, a query is expanded with synonyms mined not from a thesaurus or an embedding model but from the cortex’s *own* undirected edges: for each **similar-to** or **shares-entity** edge above a weight threshold, the entities of its endpoints become candidate synonyms, accumulated by edge weight, and the strongest few per term are kept. “Acme” expands to “ACME Inc”

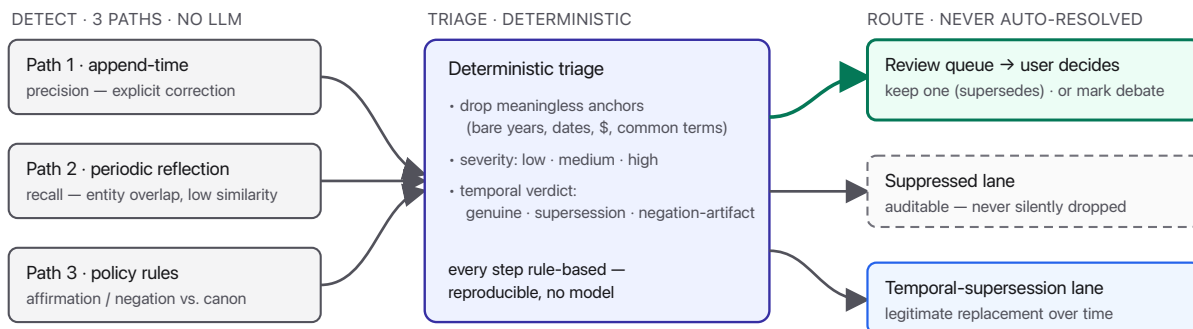
because the two recur on the same edges in *this* corpus, not because a dictionary says so — the expansion is the owner’s own vocabulary rather than a generic one. The technique applies the distributional hypothesis [Harris 1954] to the constructed knowledge graph instead of to raw token co-occurrence. We use this graph-derived query expansion as an implementation mechanism rather than a separately measured contribution; like every component of the retrieval pipeline it is not isolated by a dedicated ablation, and we leave its independent contribution and its relation to prior query-expansion methods to future work.

A full TOP_K / MAX_HOPS sensitivity sweep is left to future work; the reported configuration uses router-adjusted seed budgets (typically 20–50 nodes) within an $\leq 8,000$ -token answer context per question type, onto which session-summary and sibling-turn context are layered.

8. Contradiction Detection

This is the system’s defining capability. A *store* returns similar memories; Graphnosis additionally detects when two memories *disagree*, surfaces the pair, and leaves resolution to the user.

Deterministic contradiction detection — detect → triage → surface (never auto-resolve)



The defining un-brain act: no language model at any step; conflicts are surfaced for the owner, never resolved on their behalf.

Figure 2. The contradiction pipeline. Three detection paths feed a deterministic, model-free triage that filters meaningless anchors and assigns a severity and a temporal verdict, then routes each flagged pair to a human review queue, an auditable suppressed lane, or a temporal-supersession lane. No language model runs at any step, and the system never resolves a conflict on the user’s behalf (§8.1–8.3). Figure S3 (Additional Material) gives the conceptual brain-versus-un-brain contrast behind this pipeline.

8.1 Three detection paths

1. **Append-time (Path 1).** On ingest, new nodes are compared against existing nodes that share entities, indexed for $O(n)$ cost. This path is tuned for precision: it requires explicit correction language and rejects near-duplicate prose.
2. **Periodic reflection (Path 2).** A background scan pairs memories with high entity overlap and low content similarity, then tests for conflict signals. This path is tuned for recall: it admits content-divergence conflicts that carry no overt correction marker.
3. **Policy (Path 3).** Nodes are checked for affirmation/negation against canonical rules and trained procedures, for users who maintain explicit policies (compliance checklists, clinical protocols) alongside free text.

A scope condition holds: Path 1/2 detect *lexically or relationally explicit* disagreement; they do not catch purely numerical drift with no shared linguistic signal, and recall is lower on very small graphs. The system is calibrated for graphs of hundreds to thousands of nodes.

8.2 Deterministic triage — making it usable at scale

Naïvely queueing every flagged pair floods the user — the failure mode that leads most memory systems to abandon active consistency checking. Graphnosis manages the precision/recall tradeoff in two deterministic stages, with no LLM in either.

Detection runs at the two operating points above (precise ingest, recall-oriented reflection). *Admission* to the human review queue is then gated by a deterministic triage that (i) counts only *meaningful* shared anchors — bare years, dates, money amounts, and high-document-frequency corpus terms are filtered out, eliminating the dominant false-positive class in which unrelated notes pair only because both mention a year or an acronym; (ii) admits a high-severity identity conflict with no required anchor (a first-person flip carries none) and a framed- or numeric-value conflict on one, while requiring two anchors for weaker signals; and (iii) routes a temporal *supersession* (a memory legitimately replaced over time) to its own lane rather than the conflict queue. Each pair carries a deterministic **severity** (low/medium/high) and a **temporal verdict** (`genuine_contradiction` / `temporal_supersession` / `negation_artifact`), both computed without any model.

Definition 2 (Triage admission). Detection feeds a deterministic, model-free triage whose admission rule is a *cascade*, not a single threshold. Let $A_{\text{meaningful}}$ be the shared entities surviving the low-signal filter — bare years, dates, money amounts, and high-document-frequency corpus terms removed — let $\sigma \in \{\text{low}, \text{medium}, \text{high}\}$ be the deterministic severity, and let sig be the disjunction of four model-free conflict signals (strong-language, negation-flip, numeric-value, framed-value). The first gate is a **signal-aware anchor floor**

$$\theta(\sigma, \text{sig}) = \begin{cases} 0 & \sigma = \text{high, or a numeric conflict on a shared subject} \\ 1 & \text{a framed- or numeric-value signal} \\ 2 & \text{otherwise.} \end{cases}$$

A pair is admitted to the human review queue **iff it clears this floor and survives every suppression lane**: it is not a temporal supersession or a negation artifact (each routed to its own lane), not low-severity, not a near-duplicate, not a complementary same-subject pair, it carries at least one conflict signal unless severity is high, and — at the stricter ingest operating point only — it also passes an ingest gate (two anchors, or one for identity, with strong-conflict language and sub-duplicate prose). Clearing the floor is therefore **necessary but not sufficient**. Because $\theta = 0$ for high-severity identity conflicts and same-subject numeric drifts, a first-person flip (“*I am a vegetarian*” → “*...not anymore*”) or a subjectless count change (*20* → *25 titles*) queues on **zero shared named entities**. Every pair that fails a gate is written to the auditable suppressed lane tagged with the failing reason, never silently dropped. Because θ , σ , sig , and the lanes are all rule-based, the operating point is itself reproducible and inspectable.

Crucially, a pair the triage holds back is *recorded to an auditable suppression lane, never silently dropped* — preserving the defining guarantee that the system surfaces conflicts rather than resolving them invisibly, while keeping the active queue legible. The triage’s routing — queue, suppression, supersession, negation — is a deterministic classification of *what to surface*, never a resolution of *which fact wins*: no routed pair is overwritten and every lane is recoverable — even the supersession and negation lanes only *defer* a pair from active review, neither picking a winner nor deleting a fact, and the owner can pull any lane back into the queue — so the automatic disposition is of attention, not of truth. An LLM-mediated filter, by contrast, could offer no such reproducible operating point.

8.3 Resolution principle

Contradictions are surfaced as flagged node pairs and **never auto-resolved**. The user keeps one side (creating a **supersedes** edge and soft-deleting the outdated node, lineage preserved) or marks the pair a standing debate (both remain; flagging stops). The losing memory is not erased — it remains at reduced confidence for audit. The same flagged pairs are reachable by a connected AI client, so an assistant can raise or help adjudicate a conflict in conversation; it cannot resolve one on the user’s behalf.

9. Brain Engine — Autonomous Maintenance

Background, cooperative lanes perform reinforcement, periodic reflection (the Path-2 scan), near-duplicate review, and — when enabled — neural and local-LLM inference, always yielding to user-facing recall. All autonomous activity writes only to the overlay layers or proposes changes for approval; it cannot mutate the attested `.gai` layer. A self-healing journal audits any autonomous merge the engine applies.

10. The Neural Link-Predictor (GNN-LP)

The GNN-LP is a small feed-forward network over hand-computed node-pair features — deliberately *not* a message-passing Graph Neural Network. Conventional GNNs (GCN, GAT, GraphSAGE) iteratively propagate node representations across edges and require deep-learning infrastructure (GPU, Python). The GNN-LP instead computes all features deterministically per pair and lets the network learn only which features matter. It is therefore deployable on commodity CPU-only hardware (no GPU, Python, or deep-learning runtime), quick to train, and interpretable (each feature has a clear graph-theoretic meaning). Its output is written to the `.gnn` overlay and never merged into the attested graph (Design Invariant 1). We abbreviate it **GNN-LP**, never “GNN”, to avoid collision with Graph Neural Networks; the `.gnn` file extension is a naming artifact, not the architecture.

11. Security and Synchronization

Threat model and at-rest confidentiality. Each engram’s `.gai` file is encrypted with XChaCha20-Poly1305 (key derived with Argon2id) and integrity-tagged; device identity uses Ed25519 trust-on-first-use. The local MCP interface binds to loopback and requires a bearer token. We claim confidentiality at rest, in transit, and against every party other than the unlocked local user; we do *not* claim protection against an adversary executing code as the unlocked user — that is the job of OS/hardware isolation, not the application — nor against malware, dependency supply-chain compromise, or a coerced passphrase. We state this boundary explicitly rather than overclaim.

Order-independent merge. Synchronization is an append-only operation log replicated across the owner’s devices and reduced to state. The reduction must be independent of the order in which operations arrive.

Theorem 2 (LWW Determinism). *Let O be a finite set of operations, each tagged with a key and a triple (τ, d, s) of timestamp, deviceId and sequence number, and order them by the lexicographic relation $o \prec o' \iff (\tau, d, s) <_{\text{lex}} (\tau', d', s')$. Define the reduced state $S(O) = \text{fold}_{\prec}(O)$, applying operations in $\prec$ -order with each insert setting and each delete tombstoning its key (so the $\prec$ -greatest write per key wins). Then S is permutation-invariant: for every permutation π of O , $S(\pi(O)) = S(O)$.*

Proof. The components τ, d, s are drawn from totally ordered domains and device identifiers are unique, so $\prec$ is a *strict total order* on O — in particular antisymmetric even when timestamps tie, since d then s break the tie. A strict total order on a finite set admits exactly one sorted enumeration; hence sorting any permutation $\pi(O)$ produces the identical sequence, and $\text{fold}_{\prec}$ — applied to identical input — returns the identical state. Tombstones are themselves operations carried in this order, so a delete and a concurrent edit on one key are resolved by the same $\prec$ -comparison as any two writes; deletions thus commute with edits. ■

A timestamp-only comparator is not antisymmetric under equal timestamps and so fails permutation-invariance; the strict total order is what restores it, and a permutation test in the sync layer’s suite checks it empirically.

Encrypted op-log → order-independent merge

Operation-log entry

timestamp	deviceId	seq	op	key	value
-----------	----------	-----	----	-----	-------

(timestamp, deviceId, seq) = strict total order — equal timestamps broken by deviceId then seq
op ∈ { insert · delete-tombstone } — a delete sets a tombstone; the node is never erased

Device A — arrival order

10 · A#1 · insert · loc → "Piatra Neamț"
20 · B#1 · insert · loc → "Brașov"
15 · A#2 · delete · oldAddr ⌘

Device B — different arrival order

15 · A#2 · delete · oldAddr ⌘
20 · B#1 · insert · loc → "Brașov"
10 · A#1 · insert · loc → "Piatra Neamț"

fold under
total order

Converged state — identical on both devices

loc = "Brașov" ts 20 > ts 10 → last writer wins
oldAddr = ⌘ tombstoned, deprioritized
recoverable; never removed from the canonical set

Permutation-invariant (Theorem 2):

any arrival order of the operation set yields the same final state.

Figure 3. The op-log entry format and the last-writer-wins merge. Each entry carries a strict total-order key (timestamp, deviceId, seq); deletes are tombstones, not removals. Two devices that receive the same operations in different orders fold, under that order, to one identical state — the permutation-invariance of Theorem 2.

12. Evaluation

12.1 Setup

We evaluate on the public **LongMemEval_S** long-term-memory benchmark (500 questions), under the official end-to-end methodology with the verbatim LongMemEval judge prompts scored by **GPT-4o**. We re-ran only Graphnosis under this methodology; comparator figures are leaderboard-reported, not re-run in our harness (§12.3). The headline configurations were produced on the v0.6.1–v0.7.2 line (commits **df7d714–48785ba**), whose TF-IDF retrieval path is identical throughout; the cloud-paired figure was produced at commit **a27c400** (the **v0.7.1** release), and the retrieval path is unchanged through the published **v0.7.2** (**48785ba**, its direct descendant). The additive-scoring ablation arm (56.20%, §12.4) was added at the **v0.7.3** tag on that same unchanged retrieval path; the two naïve-top-k structure-ablation arms (49.00% cloud, 35.80% on-device) were re-run at commit **7476744**, which adds a seed-sharing ablation hook — matching the naïve arm’s seed pool to the graph arm’s while leaving the graph-arm retrieval path unchanged — so every arm shares one retrieval path (Appendix A).

12.2 Configuration decomposition

The headline number is a property of the *whole* pipeline, of which Graphnosis is the memory layer. We therefore report a configuration matrix that isolates which components are cloud-dependent (Figure 4).

Configuration	Embeddings	Query enrichment	Answer model	Accuracy
Fully on-device	none (local)	local LLM	local Llama 3.2 3B	41.60% (208/500)
Zero-embedding-API	none (local)	cloud (gpt-4o-mini)	GPT-4o (cloud)	64.60% (323/500)
Zero-embedding-API, no enrichment	none (local)	none	GPT-4o (cloud)	62.20% (311/500)
Cloud-paired	cloud	cloud	GPT-4o (cloud)	78.00% (390/500)

Configuration decomposition — same memory stack, cloud parts added

LongMemEval_S (500 Q), GPT-4o judge. The answer model is the dominant lever.

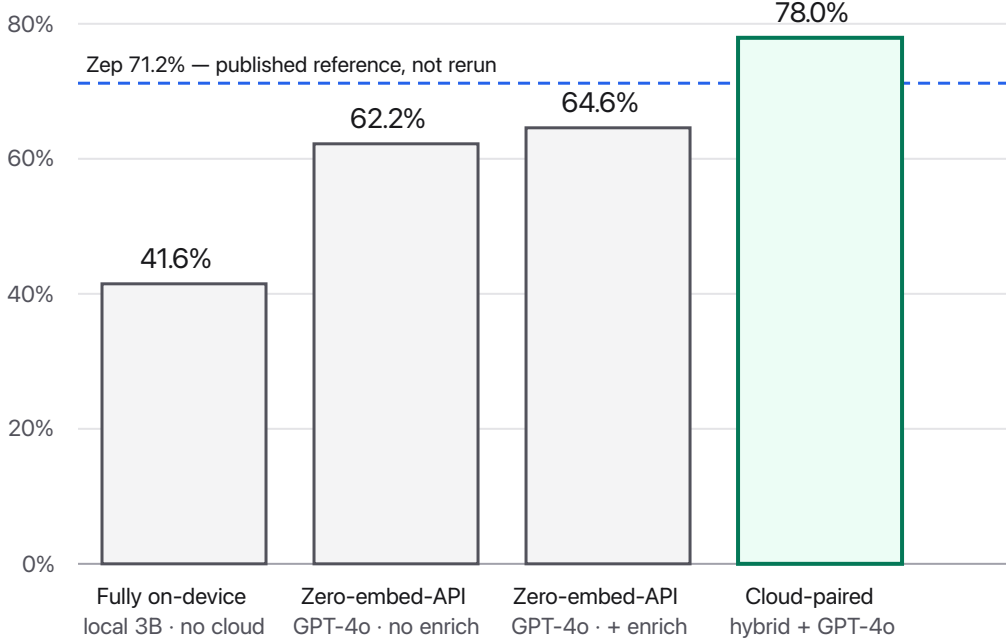


Figure 4. Each configuration shares the same memory stack; cloud components are added left to right. The answer model is the dominant lever (local 3B → GPT-4o), and the cloud-paired bar sits above the published Zep reference line (Zep’s own figure, not rerun here; §12.3).

In every row the retrieval and memory stack is **TF-IDF over a pure-TypeScript lexical index** — no Python, no vector database, no GPU, no fine-tuning; the only external components are the answer LLM and, in the cloud-paired row, optional dense embeddings layered onto that same index. The 64.60% configuration runs gpt-4o-mini enrichment — session-summary nodes at ingest and preference extraction at query time; dropping that enrichment yields 62.20%, leaving the answer model as the only cloud dependency. The cloud-paired (78.00%) configuration adds cloud dense embeddings on top of the same enrichment, so it is *not* zero-API at construction; the zero-API property is a feature of the **substrate** — the graph, the lexical index, and the contradiction triage, which hold in every row — not of the highest-accuracy benchmark recipe. Only the 41.60% configuration is cloud-free end to end. The dense path is not inherently a cloud dependency: the benchmark’s hybrid arm calls a cloud embedding adapter for reproducibility, but the shipping application runs the *identical* hybrid retrieval against the bundled on-device model (BGE-small-en-v1.5 via ONNX, §7) — so a fully on-device configuration *with* dense retrieval exists in the product. We benchmark the TF-IDF on-device arm here and leave the on-device-dense measurement to ongoing work.

Per-category accuracy (cloud-paired, 390/500 = 78.00%).

Question type	Accuracy
Single-session, user	94.29% (66/70)
Single-session, assistant	89.29% (50/56)
Single-session, preference	56.67% (17/30)
Multi-session	65.41% (87/133)
Temporal reasoning	75.94% (101/133)
Knowledge update	88.46% (69/78)
<i>Abstention (subset)</i>	90.00% (27/30)

The weakest categories — multi-session reasoning (65.41%) and user-preference recall (56.67%) — remain the clearest targets for future work (§13); single-session and knowledge-update questions are near ceiling.

Two readings follow. First, the **answer model is the dominant lever**: moving from the local 3B model to GPT-4o, embeddings held local, is the largest single jump. Second, the genuinely owner-controlled number — fully on-device, no cloud calls of any kind — is **41.60%**. We do not present 78.00% as evidence that *Graphnosis* is accurate in isolation; it is evidence that a local-first memory layer *does not break* a strong answer model while adding privacy, indelibility, and contradiction detection. The defensible empirical claim is **viability**: a private, owner-held memory layer reaches competitive accuracy when paired with a strong model and remains functional fully offline.

12.3 The one same-benchmark juxtaposition

The cloud-paired configuration (78.00%, 390/500) exceeds the published Zep figure (71.20%) by **+6.8 points** — **+34 questions** on the 500-item set. Both numbers are end-to-end accuracy on the same LongMemEval S questions under the official methodology, but Zep’s is its *published leaderboard result, produced by its authors in their own harness — not re-run by us* (§12.1); we place the two side by side without re-executing Zep’s pipeline. Even so, +34 questions dwarfs the GPT-4o judge variance we observe (± 1 – 2 questions per category at temperature 0 across our own passes), so the gap is larger than the judge variance we observe. This is the only direct cross-system comparison we make, and we make it once — a same-benchmark, same-methodology juxtaposition, not a controlled in-harness ablation of Zep. What carries the margin is the cloud-paired configuration, not the zero-API substrate: the substrate scores **62.20–64.60%** (§12.2) — *below* Zep — so the lead over Zep depends on the cloud answer model and embeddings. The architecture’s stand-alone contribution is the deterministic contradiction primitive (§12.6), not a raw QA-accuracy win.

We deliberately do **not** present a multi-system leaderboard. Higher LongMemEval figures than ours are published — Mem0, for example, reports **94.8%** under its own harness, an ADD-only store that accumulates every fact and retrieves deeply (top-50–200 memories per question) — but those systems pair different memory designs with different readers, and published figures across the field use heterogeneous metrics (retrieval recall@k versus end-to-end QA), so ranking them in a single column would compare different quantities — precisely the move a careful reviewer would reject. We neither contest nor claim the leaderboard; Zep is the single juxtaposition we draw because it is the nearest architectural neighbor — a temporal knowledge-graph memory reporting the same end-to-end metric — and even there the margin rests on the cloud answer model, not the substrate (§12.2).

12.4 The ablation that isolates the memory layer

To separate the memory layer’s contribution from the answer model, we hold the answer model fixed (GPT-4o, no query enrichment) and vary only the retrieval *structure* over the same TF-IDF seeds: the full Graphnosis graph (breadth-first traversal across typed edges), naïve top-k (the identical seeds with no graph), and full-context (no retrieval — every session chunk concatenated up to the model’s window).

Retrieval (GPT-4o, no enrichment, identical TF-IDF seeds)	Accuracy
Graphnosis graph — traversal + typed edges	62.20% (311/500)
Naïve top-k — same seeds, no graph	49.00% (245/500)
Full-context — no retrieval	22.60% (113/500)

Retrieval ablation — answer model fixed (GPT-4o)

Same TF-IDF seeds; the graph retrieval pipeline varies (LongMemEval, 500 Q)

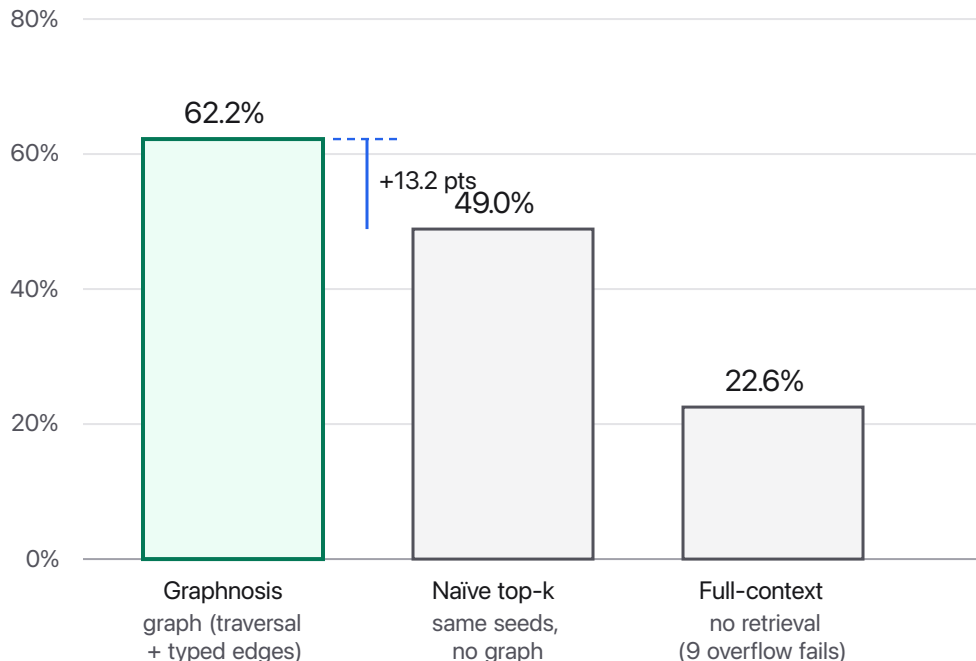


Figure 5. Retrieval ablation on LongMemEval, GPT-4o held fixed and the TF-IDF seeds identical across arms; the graph retrieval pipeline varies. Graph traversal over typed edges is worth +13.2 points over naïve top-k and +39.6 over full-context.

In this harness the pattern is clear (Figure 5): with every other variable held constant, the typed-graph structure is worth **+13.2 points** over naïve top-k retrieval and **+39.6 points** over concatenating the full history. This isolates what the §12.2 decomposition could not — beyond the answer model, the graph retrieval itself does substantial work, and it is the *structure* (typed-edge traversal and the expansion it drives, under the graph pipeline’s standard router budget), not the seed selection, that earns the gain: all three arms draw from the **identical seed pool** (the naïve arm takes those seeds as its evidence; the graph arm traverses and structurally expands them), so the variable isolated is the graph retrieval pipeline. Full-context is not merely weakest: on the longest histories it fails outright (nine questions overflowed the context window), the exact failure mode a bounded, curated subgraph is built to prevent. We read this as the controlled evidence for the founding hypothesis that a typed graph beats flat chunks for a fixed reader (the format-versus-flat-text claim, §13).

The same structure helps a local model. Repeating the graph-versus-flat contrast with the answer model swapped to the on-device Llama 3.2 3B, the dual graph scores **41.60%** (208/500) against **35.80%** (179/500) for naïve top-k — **+5.8 points** for a 3B model running with no cloud calls of any kind, the on-device mirror of the +13.2 above (conservative, since the graph arm counts eight Ollama generation errors as incorrect). The architecture’s benefit is not an artifact of a frontier reader.

Decomposing the gain: structure versus scoring rule. The +13.2 has two sources — having the graph at all, and scoring it by the hub-independent max rule (Theorem 3, §7). Holding the graph, seeds, and answer model fixed and replacing only the max rule with additive accumulation scores **56.20%** (281/500): the dual graph with naïve additive scoring already recovers **+7.2** of the gain (the structure), and the max rule earns the remaining **+6.0** (Theorem 3). Both matter, and nearly half the dual graph’s advantage is the scoring rule that makes recall hub-independent.

12.5 Efficiency

Across the 500 LongMemEval questions (cloud-paired run), the delivered subgraph is compact: a **median of 35 nodes** per query (mean 36, p95 45), rising to ~120 on the densest questions once session-summary and sibling-turn context are layered onto the 20–50-node seed set — a ranked excerpt, never the full store. This is the benchmark’s answer-context subgraph; the tighter $\leq 2,000$ -token cap of §4 governs separately what crosses the MCP boundary to external clients. Answer-model latency (GPT-4o) was a median of **1.6 s** (p95 3.1 s). The per-question graph build, which reconstructs each question’s session history from scratch, ran a median of 12.9 s; this is a benchmark artifact, not a steady-state cost — in normal use the cortex is built once and updated incrementally.

We further isolate retrieval cost and on-disk size with a separate deterministic micro-benchmark — no answer model, no per-question rebuild — over the same corpus on an Apple M3 Max (14 cores, 38.7 GB; Node 20). Warm recall (query $\rightarrow$ ranked subgraph) runs a median of **0.28 s on a 15k-node cortex and 1.4 s at 45k nodes**, scaling with cortex size and the bounded breadth-first traversal, with no API latency at any point. The query-independent synonym index is built once per graph and cached, so recall pays no per-query graph scan. On format size, the `.gai` container serializes the identical graph to **~88% of the equivalent JSON** at every size measured: MessagePack removes structural overhead, but node text is the irreducible floor, so the format’s value is machine-native parsing, an integrity tag, and at-rest encryption rather than dramatic compaction. The token efficiency that actually governs what the AI client pays is enforced separately, at the $\leq 2,000$ -token subgraph boundary (§4).

12.6 The contradiction primitive, measured directly

LongMemEval measures end-to-end QA, which §12.2 showed is governed by the answer model; it does not isolate the contribution this paper foregrounds — the deterministic contradiction triage of §8. We evaluate that primitive on its own. The triage is a pure, model-free function (Definition 2): given two memories and their shared entities, it routes the pair to one of four lanes — the human review *queue*, a *temporal-supersession* lane, a *negation-artifact* lane, or *suppression* (an auditable lane, never a silent drop). We score its routing against a labeled benchmark.

The benchmark holds **71 pairs** of atomic saved memories. Of the thirty-five genuine-conflict positives, **28 are grounded in LongMemEval_S knowledge-update values** drawn from the dataset (a charity-5K best of 27:12 later 25:50; a mortgage pre-approval of \$350k later \$400k; a Gold-tier threshold of 300 stars later 120), atomized into the saved-memory form the triage actually sees; the other **7 are curated — 5 entity-anchored conflicts** (a person’s location, an organization’s headquarters, a project’s lead, a database’s engine) and **2 first-person identity flips** carrying no shared named entity (a dietary identity; a home city) — categories the knowledge-update set under-represents. The rest are curated: temporal supersessions, negation artifacts, near-duplicates, weak-shared-entity pairs, complementary same-subject facts, and six same-subject / *different-attribute* numeric pairs. Shared entities are computed by the production entity extractor, not supplied by hand. The harness is deterministic, uses no language model, and ships in the source tree.

Building the benchmark drove the primitive forward in two rounds. As first written, the triage queued only conflicts exposing two or more shared named entities; it missed first-person identity flips, bare numeric drifts, and — revealingly — the paper’s own canonical example, because the identity rubric matched the first-person *live in* but not the third-person *lives in*. Closing these required predicate-frame and numeric-value conflict signals and an entity floor that admits identity, framed, and numeric conflicts on fewer anchors. That lifted conflict recall to complete, but a deliberately adversarial probe then exposed a new failure class:

X has 50 engineers versus *X has 200 customers* — same subject, *different attribute* — read as a conflict (two of six probe pairs). A second fix requires the differing value to quantify the same attribute and removes the over-broad numeric predicate frames. Over the full 71-pair benchmark, including the six adversarial pairs (Figure 6), every pair routes to its correct lane — 35 conflicts queued, 7 supersessions and 4 negation artifacts to their lanes, 25 non-conflicts suppressed — with no off-diagonal error, no false positive, and no loss of conflict recall.

Contradiction-triage routing — labeled benchmark

71 pairs across four lanes. Rows = human-correct lane, columns = lane the deterministic triage chose.

	actual lane (triage)			
	queue	supersession	negation	suppress
	queue	35	0	0
	supersession	0	7	0
	negation	0	0	4
expected lane	suppress	0	0	25

100% on the 71-pair author-constructed regression set · 0 off-diagonal · 0 false positives

Regression coverage, not a generalization estimate; held-out + model-authored sets scored separately (§12.6).

Deterministic, no language model. The suppress column is an audit lane — pairs recorded, never dropped.

Figure 6. Routing of the deterministic triage over the 71-pair labeled benchmark (rows: the human-correct lane; columns: the lane the triage chose). All weight is on the diagonal — 35 genuine conflicts queued, 7 temporal supersessions and 4 negation artifacts routed to their lanes, 25 non-conflicts suppressed — with no off-diagonal error and no false positive, including six adversarial same-subject/different-attribute pairs. Deterministic, no language model; the suppress column is an auditable lane, not a silent drop.

The 71-pair benchmark is author-constructed and *co-evolved with the logic it tests*, so its perfect score is evidence of **correctness across the lane taxonomy and of regression coverage**, not an estimate of generalization. To probe generalization directly, we ran the same frozen triage over a **held-out set of 24 instances** whose subjects and values are disjoint from the 71-pair set and which were authored *after* the rules were fixed: it routed **23/24 (95.8%)** to the correct lane with **zero false positives**, the lone miss a knowledge-update item whose frequency change (*every week* → *every other week*) the near-duplicate guard absorbed because the two phrasings differ by a single token. This is held-out-*instance* validation under frozen rules, not third-party-blind construction — a set whose *instances and labels* both come from outside the rule-author is stronger still, which we report next — but it directly answers the concern that the primitive and its evaluation co-evolved: on instances the rules were never tuned against, the lane taxonomy still holds. Because the triage is deterministic, both scores characterize the rule set’s behavior on these categories exactly, not a sample from a noisy model. Within that scope the primitive does what §8 claims: it surfaces entity-anchored contradictions, routes supersession and negation to their own auditable lanes, suppresses non-conflicts including the adversarial different-attribute class, and resolves nothing on its own. Building the benchmark also surfaced and corrected two real defects, one of them in the paper’s own showcase example.

We then went further, to a **model-authored held-out set**. A frontier model (GPT-5.5), given only the

four lane *definitions* and never the triage rules, authored and labeled 45 memory pairs spanning value types the tuning sets under-represent — contact details, hosting providers, policies, dated changes. Run against the frozen triage it agreed on **21/45 (46.7%)**, and the disagreements clustered: the rules reliably caught conflicts exposing a shared named entity and a modeled signal but missed conflicts whose subject is a common noun (“*my dentist is X*” vs “*... Y*” — no shared named entity) or whose value type trips no signal (a month, a country, a hosting provider). We treated the set as held-out — frozen by hash, with fixes tuned only to the diagnosed *categories*, never the individual items — and two principled extensions, a frame-general value-conflict signal and narrative-temporal supersession, lifted agreement to **35/45 (77.8%)** (conflict recall $5/18 \rightarrow 15/18$) with **no regression** on the 71-pair or held-out sets and unchanged non-conflict precision (15/15 suppressed; 0/6 adversarial false positives). The residual gaps are specific: value conflicts carried by lowercase tokens (an email local-part, a password), synonymous-negation restatements that surface as low-value queue items, and one dated-numeric pair the model and the rules route to different lanes with equal justification. Its instances *and* labels come from outside the rule-author — a model that never saw the rules, though not an independent human annotator, and the 77.8% follows post-hoc category-level (not item-level) tuning. It characterizes a **recall envelope** (entity-anchored, numeric, framed-relation, identity, explicit-temporal, and narrative-supersession conflicts) rather than asserting completeness.

13. Discussion and Limitations

What the un-brain design buys, and costs. Indelibility guarantees auditability and resistance to silent hard deletion, at the cost of monotonically growing storage and the need for prominence (not deletion) to manage attention. Determinism-first design guarantees reproducibility and removes a class of model-introduced errors, at the cost of recall ceilings that a learned retriever might exceed.

Optional intelligence, cleanly detachable. That ceiling is not a wall. The same architecture admits two optional, non-deterministic overlays — a local neural link-predictor (`.gnn`, §10) and a local-LLM layer (`.gll`) — that propose the connections and synthesized understanding the deterministic passes never recorded: the associative leaps that turn an exact record into a working memory. They are *additive and detachable*: separate files, gated from the attested layer by the promotion invariant (§5.1), removable at any time with no effect on the canonical dual-graph. We claim no determinism for them and no benchmark result rests on them; their role is to understand the owner’s data more richly and connect its nodes, never to change what the owner attested. Determinism is the floor on which correctness rests; learned enrichment is layered on top, never baked in.

Independent support for the symbolic substrate. A recent impossibility result bears directly on this tradeoff. Barman et al. [2026a] prove that any *semantically continuous, kernel-threshold* memory — the embedding-similarity design under most production memory systems — cannot simultaneously eliminate interference-driven forgetting and associative false recall, and that the architectures escaping the result are exactly those adding “*an external symbolic verifier or exact episodic record*” (their No-Escape theorem). Graphnosis’s indelible attested layer over a deterministic lexical index is such a substrate; in their own experiments a keyword/filesystem store records zero forgetting and zero false recall. The same result quantifies the cost — a pure keyword store agrees with embedding retrieval on only 15.5% of queries — which is why Graphnosis does not stop at the symbolic base, but layers typed-graph traversal (§12.4) and optional embeddings — cloud or fully on-device (§7) — over it, taking the determinism escape without forfeiting recall. The result is a 2026 preprint, not yet peer-reviewed.

Mimicry versus inversion. The contrast with that line of work runs deeper than deployment. Its companion result [Barman et al. 2026b] argues that the brain’s memory failures — forgetting, interference, false recall — are “*not bugs of biological implementation but features of any system that organizes information by meaning*,” and shows embedding memory reproduces them with no phenomenon-specific engineering; the corresponding product embraces the analogy as a *company brain*. The un-brain takes the opposite stance: these are not features to reproduce but failures to repair. Graphnosis declines to organize memory by meaning alone — it keeps an indelible, deterministic record precisely so it does *not* inherit the brain’s forgetting and its silent reconciliation of conflicts. Two programs reach opposite architectures from the same evidence: one

mimics the brain, the other inverts it. That inversion is what *un-brain* names.

Scoping the formal results. Of the three structural properties, two carry non-trivial proofs (indelibility; LWW determinism) and one is a design invariant true by construction (the promotion gate). We do not claim three co-equal theorems.

Scoping the contributions. Both research-flagged claims are now backed by measurement rather than assertion. Dual-graph coverage strictness is measured directly (§5.2). The format-versus-flat-text reliability advantage — the founding hypothesis that a typed graph outperforms flat chunks for a fixed answer model — is borne out by the controlled ablation of §12.4: with GPT-4o held fixed, graph retrieval scores 62.20% against 49.00% for naïve top-k and 22.60% for full-context, a +13.2-point margin attributable to the graph retrieval pipeline (typed-edge traversal under its standard router budget), with the seed pool and answer model held fixed across arms.

The contradiction evaluation is author-shaped, and we say so plainly. The 71-pair set co-evolved with the rules; the 24-instance held-out set and the 45-pair set are author- and model-authored — frozen, but not independently labeled. Together they establish *regression coverage* and *bounded generalization* — the lane taxonomy holds on instances and labels the rules were not tuned against — not an unbiased estimate of general contradiction-detection accuracy. The evidence the centerpiece still lacks, and that we name as the clear next step, is a **pre-registered, third-party (human-authored) labeled set with published labels, run against the frozen triage and against Zep/Graphiti, Mem0, Semantica, and Pith under one harness**. Until then we scope the claim to the adjudication *contract* — deterministic detection surfaced for the owner, never auto-resolved — which the comparison establishes by construction, rather than to a generalization-accuracy number.

Open frontiers. The largest measured gap is multi-session reasoning; hierarchical session summaries are a natural next step. Contradiction detection generalizes beyond personal memory to any consistency-critical corpus (regulatory, clinical, legal), which we view as the most promising research direction the primitive opens. The substrate also carries a procedural layer — skills with explicit goals that invoke sub-skills and bounded (max-iteration) loops — for running the owner’s recurring tasks over the job memory with fewer tokens and tighter control; we develop it in a companion paper and only flag it here.

14. Related Work

Prior art we learned exists for every ingredient — local-first software [Kleppmann et al. 2019], typed knowledge graphs, LSH deduplication [Charikar 2002], LWW/CRDT determinism [Shapiro et al. 2011], MCP, and personal knowledge graphs as a research agenda [Balog & Kenter 2019]. The contradiction primitive in particular has deep conceptual roots: truth-maintenance systems [Doyle 1979] and the theory of belief revision [Alchourrón, Gärdenfors & Makinson 1985] formalize how a knowledge base tracks dependencies and revises under conflict.

Graphnosis’s stance — surface the conflict and decline to revise without the owner’s assent — is a deliberate position in that space, carried from the symbolic-reasoning setting where it originated into a deployed, local-first AI-memory substrate. Our contribution is therefore not any single ingredient but their *combination* — and, within it, a small set of properties that — across a source audit of documented 2026 memory systems (below) — we did not find a documented system exposing together.

The systems closest in intent — Mem0 [Chhikara et al. 2025], Zep [Rasmussen et al. 2025], MemGPT [Packer et al. 2023], the GraphRAG family [Edge et al. 2024], HippoRAG [Gutiérrez et al. 2024], Sentra (sentra.app), LangChain/LangGraph memory, and platform memory features — are predominantly *append-and-retrieve*: they store memories and return the most similar ones. Several of them address contradictions, and those that resolve do so automatically rather than surfacing the conflict for the owner — the distinction from our approach is the contribution’s sharpest edge. Mem0’s update phase uses a language model to choose ADD/UPDATE/DELETE when a new fact conflicts with an old one (more recent versions instead keep both and defer to retrieval); Zep’s bi-temporal graph uses a language model to detect a contradicting edge and *invalidate* the superseded one by closing its validity window. Where they resolve, both do so on

the two axes Graphnosis rejects: (i) an LLM judges the conflict, and (ii) the resolution is automatic — silently selecting a winner or invalidating a fact (Zep today; Mem0’s earlier update phase, before its current keep-both default, which neither resolves nor surfaces). Graphnosis differs on both axes: it detects the conflict *deterministically*, with no model in the loop, and *never resolves it*, surfacing the pair for the owner to adjudicate. Auto-resolution is precisely the failure mode the un-brain is built to avoid (§3): a system that silently invalidates the losing fact reproduces the brain’s silent reconciliation, only in software. This LLM-mediated auto-resolution dates to 2024 — Graphiti’s temporal edge-invalidation, which Zep retains, and Mem0’s update phase, since superseded by its keep-both default. Determinism alone does not close the gap: Semantica [Hawksight AI 2026] performs deterministic, model-free conflict detection over a typed provenance graph, yet still resolves automatically — by credibility-weighted, temporal, or majority-voting strategies. What we fill is therefore sharper than determinism: detection that *surfaces the conflict for the owner to adjudicate and never auto-resolves it*. The adjudication contract, not determinism, is the distinguishing commitment.

Sentra (sentra.app), the most recent entrant, is closest in ambition — a bi-temporal semantic memory graph for teams and agents (per its product materials; its research line, including the impossibility theorem below, is Barman et al. 2026a, 2026b). It is deployment-sovereign (cloud, isolated VPC, or air-gapped on-prem, with data kept in the customer’s environment), so the contrast is not cloud-versus-local but one of *scope* and *substrate*: Sentra is a **company/team brain** — multi-user, captured by an agent that joins meetings — whereas Graphnosis is an **individual’s** owner-held store on their own device; and Sentra builds on embedding-semantic retrieval and a bi-temporal graph — whose primary materials do not document its contradiction handling, so we do not characterize it — where Graphnosis is deterministic, surfaces conflicts rather than resolving them, and openly auditable — the SDK is Apache-2.0 open source, and the application and encrypted-sync layer are source-available under the Functional Source License (§15) — so its determinism claims can be inspected rather than taken on trust. Sentra’s own impossibility result, however, is among the strongest external motivations for our design (§13).

Four further systems share parts of the design, and the overlap is real enough to name. **Atlas** [Scheffren 2026] is the nearest on the axis we claim: it routes *strategic* and core-protected conflicts to a working human adjudication queue (an Obsidian Markdown review file) rather than resolving them silently. Two source-clear differences remain. Its contract is *selective*, not total: per its own description, routine reassessments auto-apply through its Alchourrón–Gärdenfors–Makinson revision operators without prompting, and only strategic conflicts reach the owner — whereas Graphnosis surfaces *every* detected conflict and auto-resolves none. And its store is a *directed-only* property graph with no undirected edge class, so it is not a co-equal dual graph in our sense. We do not characterize Atlas’s detector beyond what its materials state. **Pith** [Estey-Ang 2026] is likewise deterministic, model-free, and local-first over the same MCP surface; its materials document contradiction detection, a contested state, and type-ranked resolution in a background reflection cycle, but do not make clear which conflicts remain owner-visible versus automatically resolved — so we mark it *resolution documented, total owner adjudication not established*. **Bicameral** [yhl999 2026], a production delta layer over Graphiti, is closer than most: it shares an append-only hash-chained ledger, owner-gated fact *promotion*, and *deterministic, model-free* detection (a pure graph pass, no LLM). It differs on resolution — its contradiction closure *auto-invalidates* the superseded fact rather than surfacing every conflict for owner adjudication. **OIDA** [Bottino et al. 2026] represents contradictions as persistent signed graph edges for *organizational* retrieval, but specifies no detector and down-weights contradicted nodes automatically. The pattern is instructive: determinism, local-first storage, append-only ledgers, and even human-routed *strategic* conflicts each appear elsewhere — no one of them is novel here. We distinguish owner-gated *promotion* (which new facts may enter, as in Bicameral) and *selective* human routing (strategic conflicts only, as in Atlas) from the **total adjudication contract** we adopt: *every* detected conflict recorded in an inspectable lane; admitted conflicts surfaced for the owner, none ever silently resolved. That contract is not ours alone — concurrent work reaches it independently.

Concurrent and prior work (2026). Owner-held AI memory became an active area in 2026; a source audit across arXiv, GitHub, and Zenodo finds independent work touching each ingredient — some published before this system’s first public commits (April 2026), some alongside them. We do not claim priority over any of it; what distinguishes Graphnosis is a set of *mechanism* choices, set out system by system below, that

stand independently of publication date. *Tenure* [Flynt 2026], a local-first belief store, adopts essentially the same adjudication contract — its merger “does not silently overwrite: it queues the conflict for user review, preserving both versions until the user resolves it” — but over a *typed belief store* rather than a co-equal dual graph; it does not present the substrate or retrieval theorems claimed here, and it evaluates on its own retrieval-precision benchmark. *MAGMA* [Jiang et al. 2026a] independently traverses selected relation views and then *fuses* the resulting subgraphs; *HAGE* [Jiang et al. 2026b] keeps relation-specific views over shared nodes but combines them through learned, query-conditioned routing — neither exposes the relation classes as fixed, independently-queryable edge sets the way the dual graph does. *Engram*, *Heirloom*, and *Cortex* are local-first, end-to-end-encrypted, MCP-reachable memories — Heirloom even uses the same XChaCha20-Poly1305 and Argon2id primitives — yet are flat encrypted stores with search, without a typed graph or a contradiction contract; Cortex, the closest in feature breadth, *auto*-supersedes conflicting facts (its documentation lists “Contradiction detection — Automatic with confidence scores”), the inverse of the contract here, and syncs by last-writer-wins over a hybrid logical clock without a proof. *SuperLocalMemory V3* [Bhardwaj 2026] detects contradictions deterministically with no LLM, but auto-creates the supersedes edge rather than surfacing it. The convergence is informative rather than discouraging: several groups independently arrived at deterministic, owner-respecting memory, which strengthens the design’s motivation. What none of them exposes is the *conjunction* of properties we state next.

Capability matrix. Each cell marks whether the system, as documented, provides the capability. *Local-first* credits a default of owner-held local storage — a self-hostable library (GraphRAG, LangGraph) qualifies; a hosted service whose self-host is a non-default path does not (§). *Typed multi-layer graph* credits multiple edge classes or layers; the **Yes** is its strongest form — co-equal directed and undirected edge classes over the *same* nodes (§5.2) — which the *partial* systems (one edge class, or a derived community hierarchy) do not reach. *Model-free construction* means the substrate is built with no embedding or generation model, local or remote — strictly stronger than avoiding only a cloud API. The contradiction column distinguishes *surfacing* from *auto-resolution* (see legend). Cells reflect public documentation as of June 2026; the Mem0, Atlas, Tenure, and Cortex rows are additionally verified against current source. The two *concurrent* entries (Tenure, a 2026 research preprint; Cortex, a 2026 open-source project) are included because the related-work narrative above compares against them directly; the broader 2026 wave is sourced in Appendix C. Per-row primary sources, verbatim quotes, and version pins are catalogued in Appendix C.

System	Local-first	Typed multi-layer graph	Surfaces contradictions†	E2E-encrypted sync	Model-free construction
Mem0	partial‡	partial	keeps both	No	No
Zep	partial‡	partial	auto	No	No
Sentra	partial	partial	—	No	No
Semantica	partial	partial	auto	No	Yes
Pith	Yes	partial	—	No	Yes
Atlas	Yes	partial	partial	No	partial
MemGPT / Letta	partial‡	No	No	No	No
GraphRAG family	Yes	partial	No	No	No
HippoRAG	Yes	No	No	No	No
LangChain / LangGraph	Yes	No	No	No	Yes
Platform memory features	No	No	No	No	No

System	Local-first	Typed multi-layer graph	Surfaces contradictions†	E2E-encrypted sync	Model-free construction
Tenure (<i>concurrent</i>)	Yes	partial	Yes	not assessed	No
Cortex (<i>concurrent</i>)	Yes	partial	auto	Yes	No
Graphnosis	Yes	Yes	Yes	Yes	Yes

† **Surfaces contradictions:** Yes = detects deterministically (no LLM) and surfaces the conflicting pair for the owner to adjudicate, never auto-resolving; *auto* = detects but resolves automatically — whether LLM-mediated (Zep closes the superseded edge’s validity window) or by deterministic strategy (Semantica resolves by credibility, recency, or majority vote); *keeps both* = accumulates conflicting facts without surfacing or resolving them — Mem0’s current ADD-only default (a single LLM extraction call, no UPDATE/DELETE; earlier versions auto-resolved via an LLM update phase selecting ADD/UPDATE/DELETE); *partial* = surfaces some conflicts but not all — Atlas routes *strategic* and core-protected conflicts to a human adjudication queue but auto-resolves routine ones without prompting; — = contradiction handling not documented in the system’s primary materials (Sentra), or documented but ambiguous as to which conflicts remain owner-visible versus auto-resolved (Pith — see §14); No = contradictions not addressed.

‡ Mem0, Zep (via Graphiti), and Letta are Apache-2.0 and self-hostable with a local model; the *partial* mark reflects their default hosted/cloud posture, not an inability to run locally.

The conjunction we claim. Reduced to its substantive core, the contribution is not a list of separable firsts but one *bundle* we did not find assembled in any system we examined — and that each concurrent system realizes only in part. Three properties define it:

- **A local-first, owner-held co-equal dual graph** — directed (ordered) and undirected (associative) edges exposed *to recall* as two fixed, independently-traversable classes over the same nodes (§5.2). MAGMA independently traverses selected relation views and then fuses the resulting subgraphs; HAGE uses relation-specific views with learned, query-conditioned routing; Cortex stores directed triples only. Graphnosis differs in *exposure*: a fixed directed/undirected dual class the caller queries directly, rather than relation views combined by query-adaptive routing or final-stage fusion. We do not claim this is uniquely unavailable to other graph encodings — only that we did not find it kept co-equal and independently queryable elsewhere.
- **A total, deterministic adjudication contract** — every detected conflict recorded in an inspectable lane; admitted conflicts surfaced for the owner to adjudicate, none silently resolved, with no model in the loop (§8). Tenure independently reaches the same contract over a typed belief store; the other detectors we examined auto-resolve (Cortex, SuperLocalMemory V3, Zep, Semantica) or document resolution without establishing total owner adjudication (Atlas surfaces a subset; Pith documents detection, a contested state, and type-ranked resolution). We claim the contract only as one element of the bundle, not as ours alone.
- **An inspectable, deterministic substrate** — the graph, the lexical index, and the contradiction triage are built with no model, and the substrate’s behavior is pinned by stated guarantees: node-set indelibility (Theorem 1), strict-total-order merge determinism (Theorem 2), and hub-independent traversal (Theorem 3, to which an ablation attributes +6.0 points). These are guarantees enforced largely *by construction* — audit discipline more than deep results, and we present them as such; order-independent merge in particular is standard last-writer-wins practice, stated here as a guarantee rather than offered as a contribution.

Two things support this bundle without being part of it. The **formal theorems** above are evidence the substrate behaves as stated, not a separate novelty axis. The **LongMemEval** results (§12) — 78.0% cloud-paired, 41.6% fully on-device — are evidence the assembled system is viable end to end; MAGMA and Zep both report LongMemEval results of their own, so we treat the benchmark as shared evidence, not a

differentiator.

Individually, none of the bundle’s ingredients is ours to claim as a first: local-first storage, end-to-end encryption, MCP reach, an entity graph, supersession, and hybrid-logical-clock sync are, by 2026, common (Engram, Heirloom, and Cortex all have them), and each is credited to its originators in the discussion and the references (§17). The capability matrix above compares *deployed* systems on five documented capabilities; it is not itself the proof of this bundle, which is argued system-by-system here and sourced in Appendix C. The contribution is that the co-equal dual graph, the total adjudication contract, and the inspectable deterministic substrate hold together in one owner-held system.

15. Implementation and Availability

@nehloo/graphnosis is released as open source (Apache-2.0, github.com/nehloo/Graphnosis); the encrypted-sync layer is a separate package, source-available under the Functional Source License (github.com/nehloo-interactive/graphnosis-secure-sync); and the desktop application that packages both is likewise source-available under the FSL (github.com/nehloo-interactive/graphnosis-app). Graphnosis has been in active, publicly-documented development since April 2026; the commit histories of these repositories — and the project’s founding design discussion ([original-discussion.md](#)) — are the primary record of its evolution. This work is archived on Zenodo: doi.org/10.5281/zenodo.20843387. The privacy boundary defined in §4 is enforced here: a client receives only the curated per-query subgraph, never the cortex.

The store is reachable by AI clients over MCP, with each tool carrying the determinism label of §4. On-disk formats are encrypted containers: `.gai` (canonical graph, XChaCha20-Poly1305 over MessagePack), `.gnn/.gll` (prediction overlays, stored separately and never merged into the canonical graph), and signed, air-gapped snapshot/skill packs (AES-256-GCM with Ed25519) requiring no network infrastructure.

Time-bounded memory (obligations). Alongside static facts, the store records *obligations* — memories with an explicit validity horizon, such as a deadline, a renewal, or a review-by date. While an obligation’s **expiresAt** lies in the future, it pins its source against retention purge, so a scheduled cleanup cannot quietly drop a fact the user still depends on; once it lapses, ordinary retention resumes. This is the temporal complement to indelibility — indelibility governs what is *kept*; obligations govern what must stay *active until a date* — and it maps directly onto the regulated settings of §1.2 (matter deadlines, license renewals, review cycles).

A deployed interoperability result. Over roughly three months of ordinary use, a single local encrypted store served **eight distinct MCP clients** — three Claude surfaces (Desktop, Code, Cowork), plus Cursor, GitHub Copilot, an editor plugin, a self-hosted open-source MCP client, and a local-LLM agent bundled with the application — each connecting through MCP and querying the same graph without re-ingestion or manual copying. One ran entirely on local inference, confirming the interface is independent of the answering model — any MCP-capable client implementing the Graphnosis tool contract can query the cortex, cloud or fully on-device. We report this as an existence proof that one open protocol can mediate a private, owner-held memory across a heterogeneous client set.

16. Conclusion

The un-brain is not a metaphor for how Graphnosis works; it is the design specification. Every architectural decision traces to it: indelibility (attested knowledge is preserved as canonical nodes rather than hard-erased — only superseded or made quieter in retrieval); contradiction surfacing (the brain’s silent resolution produces false memories, so software must surface conflicts rather than resolve them invisibly); and bounded reinforcement (Hebbian strengthening as a virtue, Hebbian decay as a constraint, not a goal).

This paper’s contribution is a *synthesis* rather than any single first: a co-equal dual graph, a total deterministic adjudication contract, and an inspectable deterministic substrate, held together in one owner-held system (§14). At its center is the consistency primitive — deterministic, no-LLM, three-path contradiction

detection with a reproducible precision/recall triage, under a contract in which every detected conflict is recorded in an inspectable lane and never silently resolved, which concurrent research also reaches (§14). Supporting that bundle rather than standing as separate contributions: three by-construction guarantees on the substrate and its retrieval — node-set indelibility (Theorem 1), last-writer-wins determinism (Theorem 2), and hub-independent retrieval (Theorem 3, to which an ablation attributes +6.0 points of the graph’s benchmark gain), plus a promotion gate (Design Invariant 1) — and the end-to-end demonstration that the substrate is buildable with zero construction-time API calls, runs fully offline, and is viable at competitive accuracy (64.60% zero-embedding, 41.60% on-device, 78.00% cloud-paired; §12.2). The benchmark validates the assembled system; the contribution is the architecture, and its sharpest expression is the **adjudication contract** — a guarantee of design, not of accuracy: no code path silently resolves a contradiction on the owner’s behalf. That guarantee holds by construction, independent of any benchmark, which is why no answer-model confound can explain it away.

A memory whose attested nodes persist under its stated write semantics and that never silently reconciles its own contradictions — while connecting its nodes deterministically through ordered and associative edges — is a different object from a transcript store. Building it correctly requires that indelibility be the first property — and this paper establishes it.

17. 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.
- Balog, K., & Kenter, T. (2019). *Personal Knowledge Graphs: A Research Agenda*. *Proc. ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR)*.
- Barman, S. R., Starenky, A., Bodnar, S., Narasimhan, N., & Gopinath, A. (2026a). *The Price of Meaning: Why Every Semantic Memory System Forgets*. arXiv:2603.27116. Preprint.
- Barman, S. R., Starenky, A., Bodnar, S., Narasimhan, N., & Gopinath, A. (2026b). *The Geometry of Forgetting*. arXiv:2604.06222. Preprint.
- Bhardwaj, V. P. (2026). *SuperLocalMemory V3: Information-Geometric Foundations for Zero-LLM Enterprise Agent Memory*. arXiv:2603.14588. Preprint.
- Bottino, Ferrero, Dosio, & Beneventano (2026). *Retrieval Is Not Enough: Why Organizational AI Needs Epistemic Infrastructure*. arXiv:2604.11759. Preprint.
- Charikar, M. (2002). *Similarity Estimation Techniques from Rounding Algorithms*. *Proc. ACM Symposium on Theory of Computing (STOC)*.
- Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). *Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory*. arXiv:2504.19413.
- Collins, A. M., & Loftus, E. F. (1975). *A Spreading-Activation Theory of Semantic Processing*. *Psychological Review*, 82(6), 407–428.
- Doyle, J. (1979). *A Truth Maintenance System*. *Artificial Intelligence*, 12(3), 231–272.
- Ebbinghaus, H. (1885). *Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie*. Duncker & Humblot. (Eng. trans. *Memory: A Contribution to Experimental Psychology*, 1913.)
- Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2024). *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. arXiv:2404.16130.
- Estey-Ang (2026). *Pith: A Governed Cognitive Architecture for Persistent AI Memory*. Technical Disclosure Commons, defensive publication #9660.
- Flynt, J. (2026). *Structured Belief State and the First Precision-Aware Benchmark for LLM Memory Retrieval*. arXiv:2605.11325. Preprint.
- Gutiérrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., & Su, Y. (2024). *HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models*. *Proc. NeurIPS*. arXiv:2405.14831.
- Harris, Z. S. (1954). *Distributional Structure*. *Word*, 10(2–3), 146–162.

- Hawksight AI (2026). *Semantica: knowledge graphs, context graphs, and deterministic reasoning for production AI*. Open-source software (MIT), v0.5.0. GitHub — semantica-agi/semantica.
- Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. Wiley.
- Jiang, D., Li, Y., Li, G., & Li, B. (2026a). *MAGMA: A Multi-Graph based Agentic Memory Architecture for AI Agents*. arXiv:2601.03236. Preprint.
- Jiang, D., Li, Y., Li, Q., Li, G., & Li, B. (2026b). *HAGE: Harnessing Agentic Memory via RL-Driven Weighted Graph Evolution*. arXiv:2605.09942. Preprint.
- Kleppmann, M., Wiggins, A., van Hardenberg, P., & McGranaghan, M. (2019). *Local-First Software: You Own Your Data, in spite of the Cloud*. Proc. ACM SIGPLAN Onward!.
- LangChain (software). *Memory modules / LangGraph persistence*. **ConversationBufferMemory** and the related conversation-memory classes were deprecated in LangChain v0.3.1 (scheduled removal in v1.0.0) and superseded by LangGraph checkpoint-based persistence (LangChain v0.3.1 migration guide).
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2024). *Lost in the Middle: How Language Models Use Long Contexts*. Transactions of the Association for Computational Linguistics, 12, 157–173. arXiv:2307.03172 (2023).
- Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., & Gonzalez, J. E. (2023). *MemGPT: Towards LLMs as Operating Systems*. arXiv:2310.08560.
- Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (2025). *Zep: A Temporal Knowledge Graph Architecture for Agent Memory*. arXiv:2501.13956.
- Schefren, R. (2026). *Atlas: a local-first cognitive memory with Alchourrón–Gärdenfors–Makinson belief revision*. Open-source software (Apache-2.0), GitHub — RichSchefren/atlas. Builds on Park, Y. B. (2026). *Graph-Native Cognitive Memory for AI Agents*. arXiv:2603.17244.
- Shapiro, M., Preguiça, N., Baquero, C., & Zawirski, M. (2011). *Conflict-Free Replicated Data Types*. Proc. International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS).
- Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., & Yu, D. (2025). *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory*. Proc. ICLR. arXiv:2410.10813.
- yhl999 (2026). *Bicameral: a dual-brain memory runtime for agents*. Open-source software, GitHub - yhl999/bicameral.

Acknowledgments

The research question, the evaluation methodology, the key tradeoffs, and the system architecture throughout — from its AI-native, owner-held representation and layered multigraph to the indelibility model, the determinism taxonomy, the surface-not-resolve contradiction primitive, and the federated-sync and procedural-skills layers around them — were defined and directed by the author. This direction runs end to end — across the representation and graph model, retrieval, memory integrity, encrypted sync and federation, the privacy model, and the procedural-skills layer. Implementation was AI-assisted under the author’s design specifications. During the preparation of this work the author used Claude (Anthropic) for coding assistance and drafting support, and then GPT-5.5 and Codex 5.3 (OpenAI) for independent hostile review of the claims over a few iterations; the author reviewed and edited all content, independently verified every reference and empirical result, and takes full responsibility for the scientific claims and the final text. No AI system is an author.

Appendix A — Reproducibility and Evidence

All headline numbers were produced with the open-source SDK @nehloo/graphnosis (Apache-2.0, **v0.6.1–v0.7.3**) against the public **LongMemEval_S** dataset (xiaowu0162/LongMemEval, 500 questions). The scorer in every configuration is the official LongMemEval judge prompts run on **gpt-4o**; the answer model and the retrieval structure are the only variables. Sanitized per-question scoring records — **question_id** plus our prediction and the judge verdict, keyed to the public dataset, with the dataset’s own question/answer text

removed under its license — are committed at `benchmarks/evidence/` in the SDK repository on the same TF-IDF retrieval path: the headline, full-context, and additive-scoring (56.20%) arms at commits `a421940–621555f` (the additive-scoring arm under the `v0.7.3` tag), and the two naïve-top-k structure-ablation arms (49.00% cloud, 35.80% on-device) re-run at commit `7476744`, which adds a seed-sharing ablation hook that matches the naïve seed pool to the graph arm’s while leaving the graph-arm retrieval path unchanged; each run’s command and SHA-256 are recorded in `manifest.json`. Each file reconciles to the count below.

§	Configuration	Score	Evidence file
12.2 / 12.3	cloud-paired: hybrid (cloud embeddings) + GPT-4o + gpt-4o-mini summaries/prefs	390/500 (78.00%)	<code>cloud-paired-78.0.jsonl</code>
12.2	zero-embedding-API: TF-IDF + GPT-4o + gpt-4o-mini enrichment	323/500 (64.60%)	<code>zero-embed-enrich-64.6.jsonl</code>
12.2 / 12.4	TF-IDF graph + GPT-4o, no enrichment	311/500 (62.20%)	<code>zero-embed-noenrich-62.2.jsonl</code>
12.2	fully on-device: TF-IDF + Llama 3.2 3B (Ollama), zero cloud	208/500 (41.60%)	<code>on-device-41.6.jsonl</code>
12.4	ablation — naïve top-k, GPT-4o, no graph	245/500 (49.00%)	<code>ablation-naive-topk-49.0.jsonl</code>
12.4	ablation — full-context, GPT-4o, no retrieval	113/500 (22.60%)	<code>ablation-full-context-22.6.jsonl</code>
12.4	ablation — on-device naïve top-k, Llama 3.2 3B, no graph	179/500 (35.80%)	<code>on-device-naive-topk-35.8.jsonl</code>
12.4	ablation — graph + additive scoring (GNOSIS_SCORE_RULE=additive), GPT-4o	281/500 (56.20%)	<code>ablation-additive-scoring-56.2.jsonl</code>

Per-category reconciliation (78.00%). LongMemEval distributes abstention items (`_abs`) across four categories rather than as a standalone class; the §12.2 per-category table folds them into the Combined column. The split below sums to non-abstention **363/470** plus abstention **27/30 = 390/500**.

Category	Non-abstention	Abstention	Combined
single-session-user	60/64	6/6	66/70
single-session-assistant	50/56	0/0	50/56
knowledge-update	64/72	5/6	69/78
temporal-reasoning	95/127	6/6	101/133
multi-session	77/121	10/12	87/133
single-session-preference	17/30	0/0	17/30
Total	363/470	27/30	390/500 (78.00%)

Versioning. The headline runs span the `v0.6.1–v0.7.2` SDK line, whose TF-IDF retrieval path is unchanged throughout; the headline figure was produced at commit `a27c400` (`v0.7.1`), and the published **v0.7.2** (`48785ba`) is its direct descendant on that same path. The additive-scoring ablation arm (56.20%)

was added at the **v0.7.3** tag on the identical retrieval path; the two naïve-top-k structure-ablation arms (49.00%, 35.80%) were re-run at commit 7476744, which adds a seed-sharing ablation hook that leaves the graph-arm retrieval path unchanged. The headline 78.00% and its reconciliation are published in the SDK benchmark history (`benchmarks/benchmarks.md`, Run 30) and the README badge.

Contradiction triage (§12.6). Fully deterministic — no language model, no API. The labeled sets and harness ship in the desktop application at `apps/desktop-sidecar/tests/contradiction-eval/` (committed at 1798057): `pnpm --filter @graphnosis-app/desktop-sidecar test:contradiction` reproduces the 71/71 routing result on the tuning set, `node apps/desktop-sidecar/tests/contradiction-eval/run-heldout.mjs` the held-out **23/24 (95.8%)** on the 24 instances disjoint from it (`held-out-set.json`), and `node apps/desktop-sidecar/tests/contradiction-eval/probe-precision.mjs` the 0/6 adversarial-precision result. Because the triage is rule-based, these are exact, not sampled.

Appendix B — Edge-Type Catalog

The dual graph (§5.2) carries two co-equal classes of typed edge over one node set. The complete taxonomy as constructed by the deployed system (`src/core/types.ts`, v0.7.2) is **sixteen directed** and **seven undirected** types, grouped below by function. Directed edges are read $A \rightarrow B$; undirected edges are symmetric. Every edge is inferred deterministically at construction (§6) or written by an owner-approved correction; none requires a model.

Directed edges (16).

Type	Reading ($A \rightarrow B$)	Group
<code>contains</code>	A contains B (section $\rightarrow$ paragraph; also located-in)	structural
<code>precedes</code>	A comes before B in time or sequence	structural
<code>summarizes</code>	A (a summary) distills B (its source turns)	structural
<code>discussed-in</code>	A (knowledge) was discussed in B (a conversation)	provenance
<code>cites</code>	A references B (also wrote / authored)	referential
<code>defines</code>	A defines the concept B	referential
<code>causes</code>	A produces or leads to B	logical
<code>depends-on</code>	A requires B	logical
<code>supports</code>	A corroborates B	logical
<code>contradicts</code>	A conflicts with B — the contract’s detector (§8)	logical
<code>supersedes</code>	A replaces B (new over old) — the contract’s resolver edge (§8)	logical
<code>knows</code>	A knows B (person $\rightarrow$ person)	identity / social
<code>works-with</code>	A works with B (professional)	identity / social
<code>reports-to</code>	A reports to B (hierarchy)	identity / social
<code>collaborated-on</code>	A (person) collaborated on B (document / concept)	identity / social

<code>prefers</code>	A (the owner) favors B (an approach, tool, option)	preference
----------------------	--	------------

Undirected edges (7).

Type	Reading ($A \leftrightarrow B$)	Group
<code>similar-to</code>	A and B share vocabulary (TF-IDF cosine)	semantic
<code>shares-topic</code>	A and B belong to the same topic cluster	semantic
<code>co-occurs</code>	A and B appear in the same section or passage	structural
<code>same-source</code>	A and B were ingested from the same source file	structural
<code>shares-entity</code>	A and B mention the same person, place, or concept	entity
<code>same-person</code>	A and B are two mentions of the same person across sources	entity
<code>related-to</code>	general association (fallback for all other cases)	general

The `contradicts` and `supersedes` directed types are where the representation meets the adjudication contract: detection records a `contradicts` edge with provenance and surfaces the pair (§8); an owner’s resolution — never the system’s — then writes a `supersedes` edge that lowers the retired claim’s prominence without deleting it (Theorem 1).

Appendix C — Related-Work Source Audit

Selected capability-matrix cells in §14 — contradiction handling, end-to-end-encrypted sync, and model-free construction — are grounded in a primary source below; the *local-first* and *typed multi-layer graph* cells are argued in the text (§1.2, §5) rather than re-sourced here. Quotes are verbatim from the cited material; *no claim found* means no primary document asserts the capability — we do not treat absence as proof. **Access date: 2026-06-25.** Systems with a tagged release or resolvable commit are pinned; hosted products carry the access date only. Atlas and Mem0 were additionally read at source level.

System	Version / pin	Contradiction handling — primary-source evidence	E2E-encrypted sync	Model-free construction
Mem0	rel. <code>ts-v3.0.10</code> (2026-06-24); <code>main</code> <code>0fbbb2f5</code>	<i>Keeps both.</i> README: “Single-pass ADD-only extraction — one LLM call, no UPDATE/DELETE. Memories accumulate; nothing is overwritten.”	no claim found	No — “one LLM call” (LLM-driven extraction)

System	Version / pin	Contradiction handling — primary-source evidence	E2E-encrypted sync	Model-free construction
Zep / Graphiti	Graphiti v0.29.2 (2026-06-08); main 413b9b2e	<i>Auto-resolves.</i> README: “Automatic fact invalidation with temporal history preserved”; blog: “We mark conflicting edges as ‘expired’ by setting the <code>expired_at</code> field.”	no E2E claim (at-rest BYOK only)	No — “Graphiti defaults to using OpenAI for LLM inference and embedding”
Sentra	site (no version / repo); arXiv:2603.27116	<i>Not assessed.</i> Primary materials describe a “bi-temporal context graph” but do not document contradiction handling.	no claim found (site cites SOC 2 / ISO 27001, not E2E)	not assessed
Semantica	v0.5.0 (2026-05-11); main e87f0832	<i>Auto-resolves.</i> README: “Resolve using multiple strategies” (<code>credibility_weighted</code> , <code>temporal</code> , <code>voting</code>); “Conflicts are detected, not silently overwritten.”	no claim found	Yes — “The reasoning engines, KG construction, and provenance layer are fully deterministic; no LLM is required to use them.”
Pith	site (no version / repo)	<i>Resolution documented;</i> <i>total owner</i> <i>adjudication not</i> <i>established.</i> Site: “Contradictions resolve with auditable evidence trails” and a “contested” state — the materials do not specify which conflicts remain owner-visible versus automatically resolved (§14).	no claim found (local; no sync layer described)	scoring is model-free (“Every confidence score is computed from evidence strength — not decided by a model”); extraction step not stated
Atlas	master, pushed 2026-06-13 (no tagged release)	<i>Partial.</i> paper: “Strategic + core_protected route to a human-readable Obsidian markdown queue... routine items auto-resolve via the AGM operator without prompting.”	no E2E claim; SHA-256 ledger is “for tamper detection” (integrity, not confidentiality)	partial — deterministic extractors by default, but its edges (relations, contradictions) are LLM-extracted via Graphiti; “one LLM call per downstream node when the LLMReassessor is enabled”
MemGPT / Letta	v0.16.8 (2026-05-14); main 6d8cb7fd	<i>Not addressed</i> (no contradiction mechanism documented)	no claim found (self-host: password + HTTPS only)	No — requires OPENAI_API_KEY
GraphRAG	v3.1.0 (2026-05-28); main 6d02c235	<i>Not addressed</i>	no claim found	No — “extract entities and relationships out of the raw text using the LLM”

System	Version / pin	Contradiction handling — primary-source evidence	E2E-encrypted sync	Model-free construction
HippoRAG	v1.0.0 (2025-02-27)	<i>Not addressed</i>	no claim found	No — “extract a set of noun phrase nodes N and relation edges E... via OpenIE” (LLM)
LangChain / LangGraph	langgraph==1.2.6 (2026-06-18)	<i>Not addressed</i> (store upserts/deletes JSON by key)	no claim found (deployer’s responsibility)	Yes for the raw store (namespaced JSON, no model); semantic search is opt-in and uses an embedding model
Platform memory (ChatGPT / Claude / Gemini)	hosted, accessed 2026-06-25	<i>Not addressed</i> (no conflict surfacing; closest is silent consolidation)	no claim found (“encryption at rest” ≠ E2E)	not assessed (opaque)

Notes. (1) *At-rest* ≠ *E2E*: the *deployed/commercial* systems in this matrix document encryption at rest (Zep BYOK, platform memory) but not end-to-end-encrypted device sync. We checked each row’s primary docs, README, and security/product pages as of the access date and found no such claim. End-to-end-encrypted local memory is *not* unique to Graphnosis, however: the concurrent open-source systems Engram, Heirloom, and Cortex provide it (Heirloom with the same XChaCha20-Poly1305/Argon2id primitives; §14). We therefore present E2E sync only as one element of the §14 combination, not as a standalone first. (2) *Not assessed* marks a cell with no primary-source documentation; we assert nothing either way (Sentra’s contradiction handling; platform-memory structure). (3) *Graph “partial”* credits one edge class or a derived hierarchy — HippoRAG’s triple set unioned with a synonymy augmentation (E + E’), GraphRAG’s Leiden community hierarchy — distinct from the co-equal directed/undirected classes of §5.2.

Graphnosis’s own row. Graphnosis’s matrix cells are substantiated in the text rather than by external citation: local-first, owner-held storage (§1.2, §4); deterministic, owner-adjudicated contradiction surfacing (§8); the co-equal dual graph (§5.2); end-to-end-encrypted sync (§11); and model-free construction (§6) — with the benchmark evidence and per-run provenance in Appendix A.

Concurrent work — sources (2026). Systems that appeared alongside Graphnosis and are referenced in §5.2 and §14. Each was audited at the source level where a preprint or repository exists, and each is compared on the single axis it touches — not scored across the full matrix, because most address only one element of the §14 bundle.

System	Source (pin)	Verified characterization — primary-source evidence
Tenure	arXiv:2605.11325 (Flynt, May 2026)	Local-first belief store; §5.1: merger “does not silently overwrite: it queues the conflict for user review, preserving both versions until the user resolves it” — the same adjudication contract. Built on a typed 14-field belief schema (Def. 1) with a relation type and supersession pointers — a typed/versioned belief store, not a co-equal dual graph. No theorems of the kind in §5/§7/§11; evaluates on its own PrecisionMemBench, not LongMemEval.

System	Source (pin)	Verified characterization — primary-source evidence
MAGMA	arXiv:2601.03236 (Jan 2026)	Multi-graph agentic memory; abstract/§3.3: <i>“selects relevant relational views, traverses them independently, and fuses the resulting subgraphs”</i> via RRF + a policy-guided beam — not relation classes kept independently queryable at output. Reports LongMemEval (Table 2) and LoCoMo; no formal theorem.
HAGE	arXiv:2605.09942 (May 2026)	RL-driven graph memory; <i>“relation-specific graph views over shared memory nodes”</i> combined by a learned QueryRouter (softmax over query-conditioned edge scores). Evaluates on LoCoMo + HotpotQA; no convergence/formal theorem.
SuperLocalMemory V3	arXiv:2603.14588 (Bhardwaj, Mar 2026)	Zero-LLM contradiction detection via sheaf cohomology; §5.4: on conflict it <i>“creates a supersedes edge from the newer memory to the older contradicted memory”</i> — auto-resolves, not owner-surfaced. Proves its own information-geometry theorems (Fisher–Rao, convergence) — not indelibility/merge/hub of the kind here.
Engram	github.com/EvolvingLMs-Lab/engram — main 70b89aa (2026-06-12)	<i>“end-to-end encrypted memory layer for AI assistants”</i> ; AES-256-GCM, <i>“keys never leave your device”</i> , MCP-compatible; vector similarity search. A flat encrypted/search store — no typed graph, no contradiction contract.
Heirloom	github.com/MayonaiseLover/heirloom — v1.0.4, main 5819608 (2026-05-14)	<i>“A local-first, MCP-native personal memory layer for AI.”</i> <i>“XChaCha20-Poly1305 authenticated encryption with an Argon2id-derived key ($m=64$ MiB, $t=3$, $p=1$)”</i> — same primitives as §11. Encrypted SQLite + BM25/TF-IDF; flat store, no typed graph or contradiction contract.
Cortex	github.com/gambletan/cortex — v2.2.0, main fa882ed (2026-06-16)	Local-first; AES-256-GCM + Argon2id (not XChaCha20). Stores directed subject-predicate-object triples; no undirected edge class documented. <i>“Contradiction detection — Automatic with confidence scores”</i> — auto-supersedes (salience $\times 0.3$, kept as history). <i>“Merge uses Last-Writer-Wins with Hybrid Logical Clocks”</i> ; no convergence proof.

System	Source (pin)	Verified characterization — primary-source evidence
Bicameral	github.com/yhl999/bicameral — main 2a26296 (2026-03-23); fork of getzep/graphiti, no tags of its own	<i>“production delta layer on top of upstream Graphiti.”</i> Append-only hash-chained ledger; owner-gated promotion (<i>“facts stay quarantined”</i> for non-owner content). Detection is <i>“Pure graph pass — no LLM calls”</i> (deterministic, model-free); contradiction closure <i>auto-invalidates</i> the superseded fact rather than surfacing it.
OIDA	arXiv:2604.11759 (Bottino et al., Apr 2026)	<i>“contradictions should be represented as persistent signed relations, not rediscovered from raw text at query time”</i> ; specifies no detector; a contradicted node is <i>“down-weighted or paired with its contradiction, not deleted”</i> (automatic penalty).

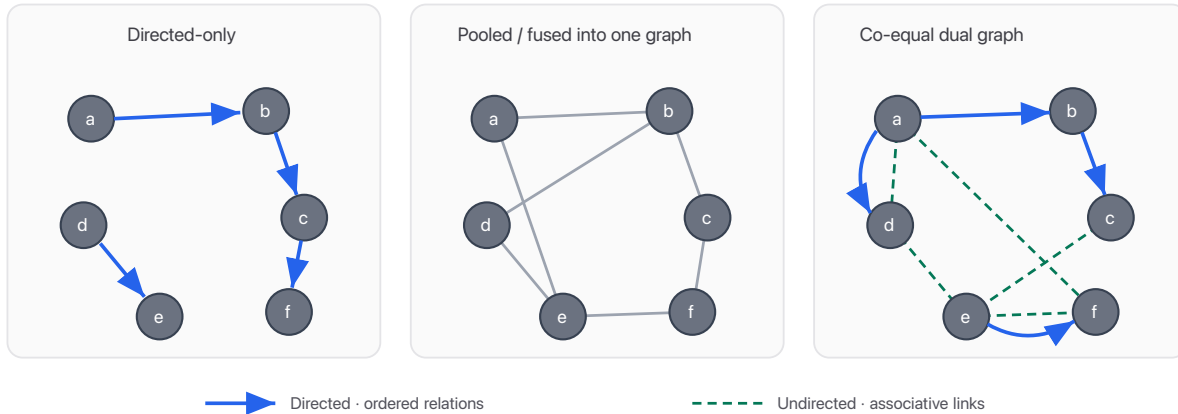
These rows substantiate the concurrent-work discussion of §14: each system touches one element of the bundle; none assembles all three. Sources accessed June 2026; arXiv identifiers and repositories are current as of the access date.

Appendix D — Additional Material (Supplementary Figures)

Supplementary figures expand on points in the main text; they are not required to follow the argument.

One node set, three relation encodings

Why keeping directed and undirected edges co-equal reaches connections the other encodings collapse.



Panels L→R: directed-only (Mem0, Zep, Cortex, Atlas) · pooled or fused to one ranking (GraphRAG, HippoRAG; MAGMA, HAGE) · co-equal dual (Graphnosis).

Same six nodes throughout; curves at a–d and e–f mark pairs carrying both classes — in the corpus only 8–13% do.

Reach: directed → short ordered chains (≈3) · undirected → broad neighborhoods (53–90) · union 64–107 (Observation 1, §5.2).

Figure S1. The same six nodes under three relation encodings (§5.2). *Directed-only* stores (Mem0, Zep, Cortex, Atlas) keep one ordered class; *pooled or fused* graphs (GraphRAG, HippoRAG; MAGMA, HAGE) collapse all relations into a single undifferentiated ranking; the *co-equal dual graph* (Graphnosis) keeps directed and undirected edges as two independently-queryable classes, with the curves at a–d and e–f marking the pairs that carry both. Only **8–13%** of connected node pairs carry both classes, and the two reach largely-disjoint neighborhoods — directed ≈3-node ordered chains, undirected 53–90-node neighborhoods,

union 64–107 — so co-equal exposure reaches connections a single class or a fused ranking does not surface as a separable class (Observation 1, §5.2).

The dual graph with detachable GNN/GLL overlays

One node set · two co-equal attested edge classes · two predicted, removable overlays (§5.1).

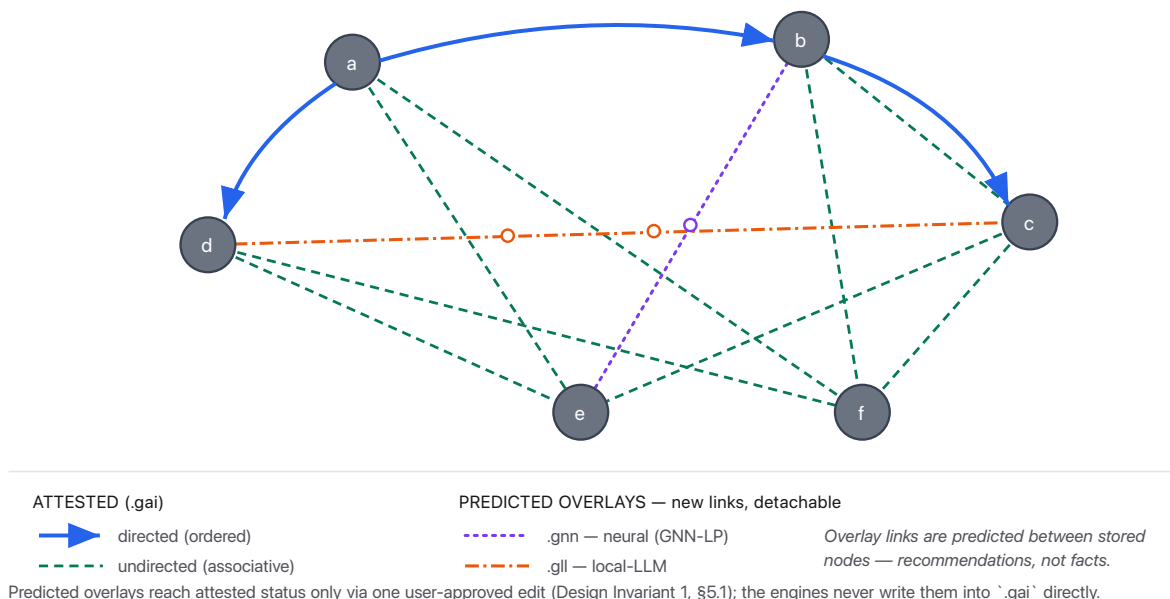
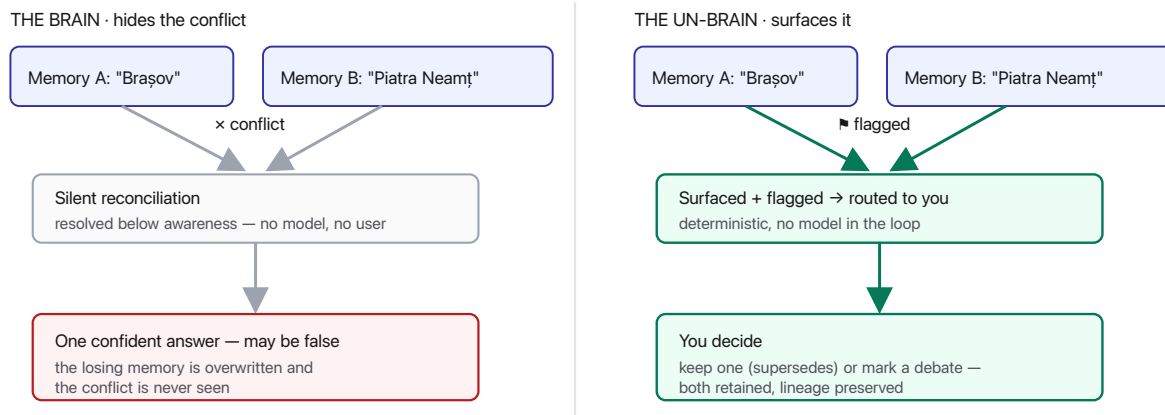


Figure S2. The layered graph (§5.1) over one node set. Attested edges form the co-equal dual graph — directed (ordered, blue arrows) and undirected (associative, green dashed). Two *detachable* predicted overlays add new links between stored nodes: the `.gnn` neural overlay (purple, from the GNN-LP) and the `.gll` local-LLM overlay (orange). Overlay links are recommendations, not facts — user-discardable, and reaching attested status only through one user-approved edit (Design Invariant 1); the overlay engines never write into `.gai` directly. Recall surfaces overlay rows only in a separate inferred layer, tagged `[gll·]`/`[gnn·]` by confidence.

Silent resolution vs. surfaced conflict



The brain's most consequential failure isn't forgetting — it's resolving conflict invisibly. The un-brain inverts exactly that.

Figure S3. The adjudication contract as a contrast — the conceptual companion to the pipeline of Figure 2. Faced with two conflicting memories, the brain reconciles them below awareness and returns one confident answer that may be false; the un-brain flags the conflict, routes it to the owner, and retains both with lineage until the owner decides. Deterministic, with no model in the loop (§8).

The forgetting curve, inverted

The brain forgets unless rehearsed; the un-brain retains, and strengthens on recall.

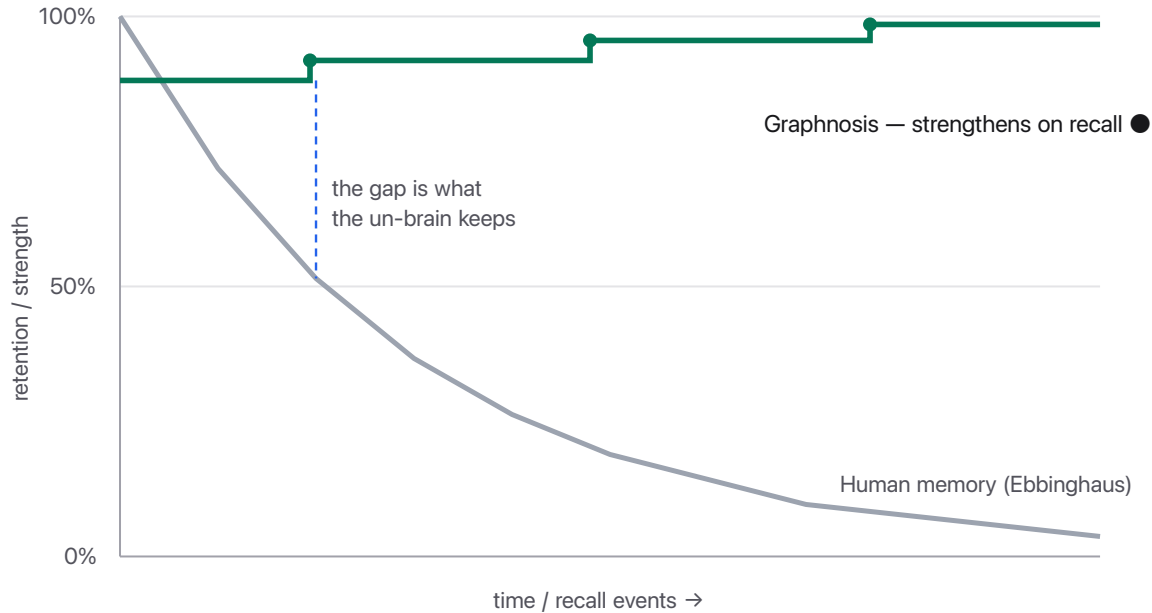


Figure S4. The Ebbinghaus forgetting curve beside the un-brain inversion (§3). Human retention decays exponentially unless rehearsed; an attested Graphnosis memory is retained at full strength and *reinforced*, not weakened, on recall — bounded by saturation (Proposition 1). The decay function is disabled by design.