ACCUMULATION IS MAINTENANCE: ANTI-PREFERENTIAL PLACEMENT AND ONLINE STRUCTURAL CURATION FOR HIERARCHICAL AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory systems for LLM agents have converged on online insertion: each new observation is routed to a location in a hierarchy or graph, and the structure is then left alone; at best, a single top layer is periodically repartitioned. We show that this neglect is not benign. On a multi-topic accumulation workload derived from real usage, undisciplined insertion produces structural pathologies: a single early node absorbs 64% of all records through preferential attachment, topics fragment across near-duplicate siblings, and the hierarchy stays flat. Recall-centric benchmarks such as LoCoMo and LongMemEval miss these pathologies, because recall holds as long as an answer is somewhere in memory. We present Gardener, a hierarchical memory system built on the principle that *accumulation is maintenance, not placement*. Gardener combines anti-preferential placement, which deterministically removes the volume advantage behind rich-get-richer growth; an online curation loop that merges, splits, and relocates nodes under strict guards (original records are never modified, and every write is executed by deterministic code); and a three-way ingestion triage that moves abstention from recall time to accumulation time. We also propose an accumulation-quality evaluation protocol pairing structural-health metrics with writer attribution (the oracle-to-e2e gap). In same-condition comparisons on an English-translated workload against online tree, graph, and extraction baselines (MemTree, A-MEM, Mem0), Gardener matches embedding-based recall while its non-destructive supersede updates eliminate stale-value citations on knowledge-update probes: 0% in every run, against roughly 50% for every baseline. Because the guards are deterministic code, their bound survives a backbone change that model behavior does not: unguarded accumulation ranges from the 64% collapse on the original backbone to moderate drift on a newer one, while guarded accumulation stays within the health gate in both regimes, curation repairing what the newer backbone's intake degrades.

1 INTRODUCTION

LLM agents need long-term memory to accumulate knowledge across sessions. The recent shift is from static construction to online insertion: where RAPTOR (Sarthi et al., 2024) builds its summary tree offline in batch, MemTree (Rezazadeh et al., 2025) descends the tree to place each new observation as it arrives, A-MEM (Xu et al., 2025) links each new note to its neighbors, and Mem0 [preprint] (Chhikara et al., 2025) adds and updates extracted facts at the LLM's discretion. Among the fifteen systems we surveyed, only xMemory [preprint] (Hu et al., 2026b) revisits the structure of already-placed records at all, and its revision stops at periodically repartitioning a single top-level theme layer (§2). The literature has grown sophisticated about "where should this go?"; the question "is what we placed still in the right place?" is only beginning to be asked.

A workload derived from real usage shows why this matters. When 31 multi-topic personal records are ingested sequentially, undisciplined insertion produces three *structural pathologies*: the *magnet* (a broad node created early absorbs 64% of subsequent records), *fragmentation* (the same topic scatters across three sibling nodes differing only in surface form), and *flatness* (no hierarchy forms; every node

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

remains a direct child of the root). The magnet is structural rather than accidental: once candidate generation votes in proportion to node size, the preferential-attachment dynamics of Barabási–Albert (Barabási & Albert, 1999) play out in a memory tree (§3.2).

These pathologies have gone unnoticed because of a blind spot in evaluation: benchmarks in the LoCoMo and LongMemEval family ask whether the system can answer a question, and as long as the answer sits somewhere in memory, scores hold however degraded the structure. Yet degraded structure erodes branch-level context injection, summary coherence, and navigability: costs existing yardsticks do not register.

Our claim fits in one sentence: *accumulation is maintenance, not placement*. Streaming observations accumulate into healthy structure only when a placement discipline and a maintenance loop act together; that health must be checked with yardsticks that measure the structure itself, not recall.

We make four contributions. **(1) Anti-preferential placement**: candidate generation that deterministically removes the volume advantage, so the LLM only ever chooses among verified candidates. **(2) An online structural curation loop** that merges, splits, relocates, and normalizes at arbitrary depth under record-immutability and deterministic-write guards; unlike xMemory (§2), which retroactively reassigns only a single top layer, every operation passes a validation gate, and the system asks the user when its confidence is low. **(3) A three-way ingestion triage**: a human-in-the-loop policy that moves abstention to accumulation time, with trigger rates measured per backbone (§6.5). **(4) An accumulation-quality evaluation protocol** combining structural-health metrics (magnet occupancy, fragmentation, depth distribution) with writer attribution (oracle/e2e); same-condition comparisons against four online tree, graph, and extraction baselines show that structural health is an axis existing systems do not even attempt to measure.

2 Related Work

Hierarchical memory: online, but only up to insertion. RAPTOR (Sarthi et al., 2024) builds its summary tree by offline batch clustering, so integrating new information means rebuilding the whole tree. MemTree (Rezazadeh et al., 2025) inserts online by $O(\log N)$ descent under cosine similarity and a depth-adaptive threshold $\theta(d) = \theta_0 e^{\lambda d}$; we adopt the same hierarchical inductive bias for our base structure. H-MEM (Sun et al., 2026) places records into a fixed four-layer hierarchy with a dedicated extraction model. Common to all three, **structure is never revisited after insertion**: MemTree never rebalances the tree it grew, and H-MEM explicitly defers lifecycle management to future work. The exception is xMemory [preprint] (Hu et al., 2026b): restricted to the top (theme) layer, it runs split and merge periodically, retroactively reassigns existing nodes (44.91% reassigned), and shows that freezing the structure hurts performance. But its reorganization is confined to that single top layer, with no validation gate, no immutability guarantee for sources, no human involvement under uncertainty, and no measurement of the structural dynamics of accumulation. Meanwhile, an observational study [preprint] (Zhou et al., 2026) finds that free-form reorganization of memory files without preservation discipline sharply degrades accuracy. The gap, then, is not reorganization per se but **verified reorganization**: precisely the point our curation loop (merge, split, and relocation at arbitrary depth, with guards and ask-back) fills.

Graph memory: updates exist, but destructive. In the graph line, HippoRAG (Gutiérrez et al., 2024) and HippoRAG 2 (Gutiérrez et al., 2025) only add incrementally to a knowledge graph. The lines that do handle updates are mostly **destructive**: the memory evolution of A-MEM (Xu et al., 2025) rewrites the context and tags of neighboring notes, AriGraph (Anokhin et al., 2025) deletes stale edges on contradiction detection, and Mem0 [preprint] (Chhikara et al., 2025) leaves the four operations ADD/UPDATE/DELETE/NOOP to the LLM's discretion. The only non-destructive precedent is the bi-temporal edge invalidation of Zep/Graphiti [preprint] (Rasmussen et al., 2025) (marking invalid instead of deleting), which our §4.5 carries into a tree context and validates with knowledge-update probes (we cite the architecture only). The contrast runs along two axes: even where update operations exist, structural reorganization (merge, split) is absent, and destructive updates cannot in principle answer "what was the original value" temporal queries; our record-immutability principle and marking-style invalidation depart on both.

Consolidation and idle-time processing: derivatives, not correction. The reflection of Generative Agents (Park et al., 2023) **generates** higher-level insights with evidence citations but does not re-place existing records; our roll-up evidence citations (§4.4) inherit this lineage. MemoryBank (Zhong et al., 2024) adds Ebbinghaus-decay forgetting over a fixed three-layer chronology. Sleep-time compute [preprint] (Lin et al., 2025) shares the **timing** of idle-time recomputation with our nightly curation, but its object is context precomputation. Derivative generation, decay, and precomputation are all distinct from structural correction: re-placing what has already been placed.

Evaluation: yardsticks that measure only recall. LoCoMo (Maharana et al., 2024, on average 600 turns and 16K tokens, up to 32 sessions) and LongMemEval (Wu et al., 2025, five abilities, including knowledge update and abstention) measure recall accuracy: as long as the answer survives anywhere in the store, the structural pathologies escape these yardsticks. We inherit RoboMME's (Dai et al., 2026) oracle/e2e separation and combine it with structural metrics (the inheritance is stated). HaluMem [preprint] (Chen et al., 2025) is the closest precedent on the accumulation side, but it measures hallucination in extraction and update **operations**; we measure the health of the **structure**. StructMemEval [preprint] (Shutova et al., 2026) evaluates organizing memory into a given structure as a task, but its measurement is still final-answer accuracy, with no quantitative yardstick for the structure itself. MemoryAgentBench (Hu et al., 2026a) shares our incremental-ingestion methodology. Our position in one phrase: a benchmark formalization of structural-health metrics, plus writer attribution.

3 The Accumulation Problem

3.1 Setting

Streaming multi-topic personal records (work logs, expenses, schedules, contacts) arrive in time order and must be processed immediately: an online setting with no batch reconstruction. The store is a **tree with a soft-link overlay**: each record belongs to exactly one tree node (unique membership, the base axis for navigation and summarization), and cross-branch relations are soft links (an $n{:}m$ graph overlay) that do not alter the tree. On top sits the **record-immutability principle**: originals are never modified or deleted; reorganization changes only placement, derivatives (summaries), and the overlay. This principle underwrites auditability and the "original value" temporal queries of §4.5.

3.2 Reproducing preferential attachment

If candidate generation votes in proportion to a node's record volume—the natural default, since larger nodes accumulate larger sums of retrieval hits—the placement probability becomes $P(\text{place} \to v) \propto \text{size}(v)$: exactly the growth dynamics of the Barabási–Albert model, in which large nodes grow larger and a winner-take-all hub forms. Measured growth curves confirm this: in the pre-improvement system, once a broad node had formed, no new node was created for 23 consecutive messages (a **magnet absorption phase**), and final maximum node occupancy was 64%. The tree-memory literature does not engage this dynamic as a dynamic: MemTree's $\theta(d)$ responds only to depth and is insensitive to node size (confirmed by measurement in §6.1), and the nearest neighbor, xMemory's sparsity objective, penalizes imbalance as a static objective without the growth-dynamics framing or time-series measurement (§2).

3.3 Structural-health metrics

On a tree $T = (V, E)$, with $\text{entries}(v)$ the number of records placed directly on node v, $P = \sum_v \text{entries}(v)$, and U the number of unclassified records, we define four indicators, all computed deterministically (no LLM involved), so comparison across systems and over time is reproducible.

- **Magnet occupancy** $M = \max_v \text{entries}(v)/P$. The outcome indicator of preferential attachment (BA-model hub formation in a tree context); gate threshold $M \leq 0.30$.
- **Fragmentation index** $F = |\{(u, w) : u, w \text{ siblings}, \ J(\text{tok}(u), \text{tok}(w)) \geq 0.5\}|$, the number of sibling pairs whose title-token Jaccard similarity is at least 0.5: a proxy for scattered placement of the same topic. The threshold is $F = 0$. Heterographic fragments

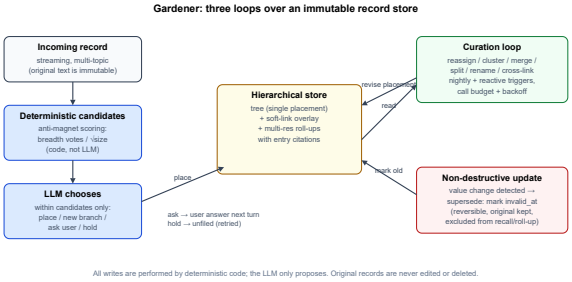


Figure 1: Gardener system overview. Around the immutable store (center) run the immediate placement loop (left), the curation loop (upper right), and non-destructive updating (lower right). All writes are performed by deterministic code.

(e.g., "IR materials" vs. "fundraising") are missed; an LLM-judged indicator would break reproducibility, a deliberate trade-off of a deterministic proxy.

- **Depth distribution** $D(d) = |\{v : \text{depth}(v) = d\}|$ and mean branching factor $\bar{b}$. Distinguishes flat degeneration (all $d = 0$) from hierarchy formation; the causal observation variable in the ablations (§6.4).
- **Unclassified rate** $U/(P + U)$. Measures the cost of the no-forced-placement policy; the threshold is ≤ 0.10.

The three thresholds derive from measurement (regression cut lines separating the pathological state $M = 0.64$, flat, from the improved state $M \approx 0.20$, hierarchical); they are calibration targets per system and domain, not universal constants.

4 Method: Gardener

Gardener is three loops running over an immutable record store (Figure 1): the **immediate loop** places each new record by deterministic candidate generation and LLM choice within the candidates (§4.1–§4.3); the **maintenance loop** (curation) corrects already-placed structure on nightly and reactive triggers (§4.4); and for records whose values change, **non-destructive updating** marks the old record invalid (§4.5). In all three loops, writes are performed only by deterministic code; the LLM proposes (§4.7).

4.1 Anti-preferential placement (immediate loop)

When a new record arrives, **deterministic code** generates the placement candidates and the LLM chooses only within that candidate set: a third way between the fully deterministic placement of the MemTree line and the full LLM discretion of the A-MEM and Mem0 line. The candidate score is

$$\text{score}(v) \;=\; \frac{\sum_{k \in \text{kw}} \max(\text{lexical hit}_k) \;+\; 4 \cdot [\text{title match}]}{\sqrt{|v| + 1}} \;+\; \text{recency bonus}, \tag{1}$$

and two devices block preferential attachment: **per-keyword breadth voting** (each keyword votes for a node at most once, removing the volume-sum advantage of a large node containing the same keyword many times) and **square-root size normalization** (the denominator $\sqrt{|v| + 1}$ makes any residual size advantage diminishing); what size can still earn is only "diversity of related keywords," a legitimate relevance signal. Every LLM output (candidate id choice, new-branch creation, hold, ask-back) passes deterministic validation: hallucinated ids demote to unclassified; branch creation is subject to a depth cap and idempotent reuse of same-parent, same-title branches.

4.2 Hierarchy formation during ingestion

Two prompt disciplines create hierarchy at insertion time. First, the **short-category-title discipline**: branch titles must be short, reusable, fixed category names (noun phrases), and sentence-like summary

titles are forbidden (not reusable; they seed broad magnet nodes). Second, the **wide-branch nudge**: a branch whose direct records exceed a threshold (8) is marked "wide," and for records that are details of it, creating a sub-branch is encouraged over direct placement. This combination forms hierarchy during ingestion (§6.4 confirms causality: nudge off $\Rightarrow D(1) = 0$).

4.3 THE THREE-WAY INGESTION TRIAGE (CLARIFICATION-IN-THE-LOOP)

The placement decision splits into three paths by confidence. (1) **Confident placement**, the majority path. (2) **Ask-back on ambiguity**: when several candidates are plausible or only an unrelated broad branch exists, the system neither forces a placement nor abandons the record; it presents two or three candidates and asks. The record is registered as pending (with a TTL) and placed definitively by the user's answer. (3) **Hold**: the record stays unclassified and curation retries it. The triage moves recall-time abstention ("I don't know") to **accumulation time** ("please confirm where this goes"), cutting off misplacement before it contaminates downstream summaries and recall; the trigger rate is a measured, backbone-dependent quantity bounded by the pending queue and TTL (§6.5).

4.4 ONLINE STRUCTURAL CURATION (MAINTENANCE LOOP)

Curation consists of seven steps: relocation of unclassified records, cluster creation (2+ records), sibling merge with link rewiring, procedural marking (non-destructive), bloat split, title normalization, and related-link proposal. Triggers are two-path (nightly and reactive); a no-progress backoff and an LLM-call budget cap control the cost.

Each line of a multi-resolution summary (parent = summary, children = originals) carries a suffix citing its evidence records ($\langle e\colon \ldots \rangle$), inherited from the reflection citations of Generative Agents. The LLM is instructed to preserve citations in promoted summaries, and **deterministic validation removes hallucinated citations**. Citations live only in storage and browsing representations and are deterministically stripped from LLM-facing prompt surfaces (Appendix H reports the recall regression that motivated this).

4.5 NON-DESTRUCTIVE UPDATES (BI-TEMPORAL SUPERSEDE)

When a value update is detected (a postponed schedule, a changed amount), the LLM outputs only the replacement-target judgment (`supersedes`) and deterministic code **marks** the old record with `invalid_at`. Nothing is deleted and the marking is fully reversible (unsupersede). Stale values drop out of recall at three points (retrieval, roll-up synthesis, summary-promotion input), while a history option still answers "when was it originally?" temporal queries—queries that destructive updating (rewriting, deletion) cannot in principle answer.

Measured results: at introduction time (Korean workload, original backbone; Appendix G), stale-value citation on the knowledge-update fixture was eliminated entirely (**oracle 28.6%→0%, e2e 42.9%→0%**) with no suite regression. The result carries over to the primary English evaluation: across all 18 forbidden questions, the writer's invalidation produces **zero stale-value citations in every run**, including under the noisy-history, remote-update conditions of e8.

4.6 THE RECALL LAYER: HYBRID RETRIEVAL, CHOOSING AND PRESENTING APART

Recall is a hybrid: three channels collect candidates inside the reader agent's search tool and Reciprocal Rank Fusion (Cormack et al., 2009) merges them—a **lexical channel** (relevance ranking from full-text search); a **cue channel** matching entities and tags extracted at accumulation time against query tokens by bidirectional prefix match (zero extra LLM calls; accumulation artifacts promoted into recall assets); and an **embedding channel** (multilingual cosine similarity for cross-lingual and synonym queries; absent a backend, it degrades to the other two).

Two disciplines keep this layer consistent with §4.5. First, **the structural filter is enforced across all channels**: category and period conditions and the exclusion of supersede-invalidated records apply at the store-query level; embedding similarity cannot resurrect invalidated records, so the freshness guarantee holds regardless of retrieval method. Second, **choose by relevance, present by recency**: the fused ranking decides only which records to pick, and the final presentation reverts to reverse-chronological order, because relevance-ordered presentation measurably pushes stale values

to the top when old and new records coexist without a declared supersede. As an auxiliary device, when a category-filtered search returns zero results, a **deterministic relaxation fallback** drops the category and retries once.

The third discipline is the **active recall loop**, responding to response avoidance at scale (§6.3). A query turn passes through two stages before response generation: **pre-search**, in which deterministic code guarantees one hybrid search with the verbatim question (prompt discipline alone collapses into no-lookup abandonment as records grow, by measurement), and a **dedicated retrieval stage** whose sole mission is evidence collection, re-searching from different angles when initial results are thin and passing only question-relevant evidence to the response stage. Code guarantees the search, the LLM supplies the strategy: the write-path principle of "the LLM proposes, code executes" applied to the read path. IRCoT (Trivedi et al., 2023), Self-RAG (Asai et al., 2024), and MRAgent (Ji et al., 2026) share this framing of interleaving retrieval with reasoning, and the dedicated stage follows the active reformulating retrieval of MRAgent; what distinguishes ours is that the first search is guaranteed by deterministic code rather than prompt discipline.

4.7 Safety invariants

All writes are performed by deterministic code; the LLM goes no further than proposing; originals are immutable. Guards validate sibling conflicts, parent existence, cluster size, supersede (both sides exist, no self-supersede, idempotent re-invalidation), and the existence of citations in promoted summaries (hallucinated citations removed), cutting off false confirmations at the source. The prompt-surface/storage separation of §4.4 is another application of the same isolation invariant.

5 Evaluation Protocol

Structural-health metrics: what recall yardsticks cannot see. The case for the metrics of §3.3 is twofold. First, structural pathologies do not register in recall scores: even in a tree with a 64% magnet, retrieval still succeeds as long as the answer sits *inside* that broad node (in §6.1, MemTree records 82.4% recall at 42% magnet occupancy). Second, the pathologies nevertheless carry real costs: when one branch holds every topic, branch-level context injection collapses, and the coherence of multi-resolution summaries and navigability degrade with it—costs invisible to recall, requiring a separate yardstick that measures the structure itself.

Writer attribution (oracle/e2e). To isolate the contribution of the accumulator (the writer), we compare two cells: the **oracle** cell injects raw records into a ground-truth tree (an "ideal placement only" condition), and the **e2e** cell ingests through the full pipeline. This inherits the RoboMME protocol; combining it with structural metrics is our extension. The gap separates writer-induced loss from retrieval loss: on the primary English suite it is -11.8 points (oracle 96.1% vs. e2e 84.3%, medians of 3 runs). The sign is itself informative: on the original backbone the needle-condition gap was *negative* (oracle 50% vs. e2e 66.7%)—perfect placement of raw text was insufficient, and the writer's summary refinement dominated (Appendix G). Queries run in a session channel separate from ingestion; conversational residue otherwise inflates e2e by about 10 points (measured).

Fixture suite. The evaluation suite consists of eight English fixtures (e1–e8) spanning temporal, entity, locus, procedural, multi-topic (structural-pathology regression), abstention, knowledge-update (with stale-value citations tallied separately as a forbidden failure rate), and noisy-history (LongMemEval-style) conditions (Appendix B); the oracle cell also receives ground-truth invalidation, keeping the ideal-placement condition consistent. The suite is a structure-preserving English translation of the Korean originals (k1–k8) under machine-checked isomorphism (Appendix B); both suites ship in the reproduction package, and the originals serve the cross-lingual check of Appendix G.

6 Experiments

6.1 Setup and baseline comparison

Setup. The primary evaluation is in English: a multi-topic workload translated from real usage (a 31-message QNA corpus) and the fixture suite e1–e8 (51 questions, §5); the original Korean workload

Table 1: Recall and freshness on the English suite (51 questions / 18 forbidden; `gemini-flash-latest`; medians of 3 runs, per-run forbidden counts in parentheses). "Avoid." = response avoidance, counted as failure.

System	Recall	Stale-value cit.	Avoid.	Update mechanism
Gardener (oracle bound)	96.1%	0.0%	3.9%	ground-truth invalid.
Gardener (e2e)	84.3%	**0.0%** (0, 0, 0)	15.7%	determ. supersede
Unstructured flat	84.3%	0.0% (0, 0, 0)	15.7%	supersede kept
MemTree reimpl. (paper)	82.4%	**50.0%** (8, 9, 9)	0%	none
Mem0 (official OSS)	82.4%	50.0% (11, 8, 9)	0%	LLM update/delete
A-MEM (official)	80.4%	55.6% (11, 10, 9)	0%	evolution rewriting

Table 2: Structural health (31-message translated QNA; `gemini-flash-latest`; tree systems only; medians of 3 runs except cells marked n=1).

System	Magnet M	Branches	Fragm. F	Notes
Gardener (intake → curated)	**31%** (27–55) → **22%**	13 → 11	0	unfiled 33→4%
Unstructured flat	—	0	—	100% unclassified
MemTree (paper θ_0=0.4, λ=0.5)	42% (32–45)	6	0	sentence-like titles
MemTree (θ_0=0.3, n=1)	55%	5	0	low θ_0 worsens magnet
MemTree (λ=0.1, n=1)	19%	**24**	0	**deep chains** (depth 8)
Mem0 / A-MEM	undefined	—	—	non-tree

serves as a cross-lingual check (Appendix G). Baselines: an **unstructured flat** configuration (ours, with hierarchical placement turned off), **Mem0** (official OSS 2.0.12), **A-MEM** (official code, default embedder), and **MemTree** (no official code is available, so ours is a stated **reimplementation** from the paper's formula $\theta(d) = \theta_0 e^{\lambda d}$, with a sensitivity sweep). All systems share the same backbone (`gemini-flash-latest`; the original deployment backbone was retired upstream mid-study, which we treat as a measured condition, §6.4), ingestion order, questions, short-answer response adapter, and deterministic scoring; each system's recall path uses its own defaults. Main comparison cells are **medians of 3 runs each**; sensitivity-sweep variants are single runs, marked as such. Appendix F states the comparison conditions in full.

The results split along three axes.

(1) Recall—parity, with failures of a different kind. In Table 1, Gardener's e2e recall (84.3%) sits at the top of the embedding-baseline band (80.4–82.4%) and level with the flat control; recall is not the axis on which these systems separate. What separates them is the *kind* of failure: every Gardener failure in every run is a response avoidance—among answered questions, accuracy is 100%—whereas the baselines answer everything and are wrong silently, half the time citing a stale value. The oracle bound (96.1%) attributes the residual gap to the writer (§5).

(2) Freshness—decided by the presence and determinism of invalidation. Median stale-value citation is 50.0% for MemTree (no invalidation), 50.0% for Mem0 (discretionary UPDATE), 55.6% for A-MEM, and **0.0%** for Gardener in every run. The flat control also records 0.0%: freshness protection comes from the deterministic supersede write guard, not from the tree—an attribution the ablation-style contrast makes for free. Supersede is stable across runs, and unlike destructive updating it still answers "what was the original value" queries.

(3) Structural health—the only tree-to-tree comparison, and the role of maintenance. In Table 2, MemTree's paper-value configuration records a median magnet of 42% (32–45%) against Gardener's 31% at intake, which one curation pass lowers to 22% while cutting unfiled records from 33% to 4%: on this backbone the intake itself is noisier (§6.4), and the maintenance loop is what restores structural health—the thesis in miniature. The sensitivity sweep (Figure 4, Appendix F) answers the objection "isn't $\theta(d)$ enough?": lowering θ_0 deepens the magnet (55%), and lowering λ escapes it only by fragmenting into deep chains (19% at 24 branches, depth 8); the depth axis cannot control

magnet and fragmentation at once. For non-tree systems (Mem0, A-MEM) the structural metrics are not even defined; the asymmetry itself corroborates that structural health is an axis existing systems have no notion of measuring.

6.2 Main experiment: structural health of accumulation

The main structural-health results are three. First, **guarded accumulation stays within the health gate** on the primary English workload (magnet 31% at intake against an unguarded historical collapse of 64%), with the anti-preferential bound enforced in code (§6.4). Second, **curation repairs what intake degrades**: one pass lowers the magnet from 31% to 22% and cuts unfiled records from 33% to 4%, originals untouched (Figure 3, Appendix D)—on the newer backbone, whose intake is noisier, the maintenance loop carries more of the health budget: the thesis operating as designed. Third, **hierarchy formation is backbone behavior above a deterministic floor**: the newest flash-class backbone keeps the 31-message tree flat, an opus-class backbone forms two levels on the same corpus (Appendix E), and four levels formed in live-server accumulation on the original backbone. Figure 5 (Appendix G) contrasts the two system states after ingesting the same 31 messages on the original backbone: all guards off, one magnet node absorbs 27 records in an effectively flat tree (64%); Gardener forms a hierarchy of 12 branches with at most 8 records each (20%).

6.3 Recall preservation

We verify that designing for structure does not cost recall. On the primary English suite, e2e recall with hybrid retrieval (§4.6) is **84.3%** (median of 3 runs) against a 96.1% oracle bound, level with the flat control and the embedding baselines (§6.1); every residual failure is a response avoidance. On the LongMemEval-style port (e8), e2e holds at 10/12 and abstention (e6) at 7/9 (medians). The hybrid layer itself was introduced through a measured gate on the original Korean workload, where it lifted lexical-only recall from 74.5% to 90.2% (Appendix G).

Three progressively harder out-of-domain probes test generalization (full anatomy in Appendix A). On a LoCoMo pilot (English personal chat), hybrid recall and response discipline turn an initial total loss (0/22) into 10/22. Scaling to the full corpus (10 dialogues, 4,972 messages, 1,984 questions) shows that **the structural axis generalizes out of domain**: all ten dialogues meet the §3.3 gates. The scale-up also surfaced a response-avoidance bottleneck, motivating the active recall loop (§4.6); a controlled re-run on the same accumulated state attributes the improvement to the loop (61.8%→25.0%; 21.1% corpus-wide), with abstention on unanswerable questions unchanged. Finally, on the original English LongMemEval knowledge-update and abstention subsets (102 questions, haystacks ingested session by session into isolated stores), the system scores 52/102 by LLM judge under our own backbone and judge conditions; 44% of the knowledge-update failures trace to the writer's recording selectivity rather than to recall, and all 102 independent accumulations reconfirm the structural axis (gate met on 100 of 102).

6.4 Ablations: establishing defense in depth

We toggle the six accumulation mechanisms (anti-preferential scoring, sibling merge, ask-back, hierarchy nudge, title discipline, bloat split) off individually and in combination over the 31-message corpus, on both backbones; Figure 2 (Appendix C) summarizes.

On the original deployment backbone (Korean workload), the layers behave as a defense in depth: **no single ablation breaks the structure** (any one mechanism off stays at 18–24%, indistinguishable from the 20% baseline), removing four in combination holds at 23%, and **only removing every layer collapses the structure** (magnet 64%, 6 nodes, effectively flat), reproducing the historical pre-improvement measurement exactly. The interpretation is a complementary double defense: title discipline blocks the *formation* of broad nodes, anti-preferential scoring blocks their *growth*; rich-get-richer returns only when both are removed. On a small corpus (13 messages, same backbone), nudge-off alone eliminated depth-1 nodes: hierarchy formation is a causal effect of the nudge.

On the current backbone (English workload), the same removals produce a different picture: single ablations land at 25–42%, the 4-way removal at 36%, and all-guards-off drifts only to 33%—no ablation reproduces the collapse. We report this contrast deliberately: **the collapse magnitude is a property of the backbone's placement temperament, not of the guard stack**. Left to model

discretion, structure varies with backbone generation in both directions, and a system cannot ship a temperament. What the guards contribute is the part that does not vary: a volume-blind candidate bound enforced in code, plus the curation loop that repairs whichever degradation the current backbone produces (§6.2).

6.5 Cost, scalability, and model portability

The ask-back trigger rate is backbone-dependent: 0–3/31 on the original backbone, 3/31 for opus-class, and 9/31 (median) for the newest flash-class backbone, which asks far more liberally. The rate is thus a measured property of the backbone—bounded in all cases by the pending queue and TTL, and adjustable through the confidence threshold—not a fixed standing burden. Curation costs one automatic firing per session within a 6-LLM-call budget (§6.2); a no-progress backoff cuts off pointless repeats.

Model portability. Flash-class backbones of two generations and an opus-class backbone all meet the structural-health gates: defense against structural pathologies is **model-independent**, with capability and generation surfacing as style rather than deciding the floor (Appendix E).

7 Discussion & Limitations

Domain, workload, and language. Our evaluation is a single-domain (work journal), single-user workload; multi-user conditions are future work. The primary evaluation runs in English on a workload translated from the Korean originals (machine-checked isomorphism, neutralized entities, author-reviewed); the original Korean real-usage workload serves as a cross-lingual check, and on the shared backbone the two languages behave equivalently (Appendix G). One asymmetry is disclosed: Gardener's internal prompts operate in Korean (Appendix I), so stored titles and roll-ups remain Korean even for English input (measured), while reader answers follow the question language through a response adapter; the guards and scoring are language-independent code. The tree-plus-soft-link structure has a practitioner prototype in note tools; our contribution is to **formalize, automate, and validate** it.

Backbone dependence. The original deployment backbone was retired upstream mid-study, and unguarded collapse magnitude (64%→33%), ask-back appetite, hierarchy formation, and intake noise all shifted with the replacement (§6.4–§6.5). We report both backbones and read the shift as evidence *for* deterministic guards, but claims tied to one backbone—the 64% collapse above all—are labeled as such and should not be projected onto other models.

Reader recall and hybrid retrieval (honest reporting). The initial system's recall was lexical only; the measured shortfall on the Korean workload gated the introduction of hybrid recall (§4.6), which closed the gap (Appendix G). Remaining limits: 15.7% of primary-suite questions end in response avoidance (counted as failure, Table 1); procedural queries retain run-to-run amplitude; abstention erosion by embedding semantic neighbors was observed twice; LoCoMo response avoidance fell to 21.1% but does not vanish; compositional reasoning over retrieved facts (token F1 0.228) is the next weakest axis. Our contribution axes remain **orthogonal** to retrieval: a retrieval improvement can be added to any system, a structural maintenance loop cannot.

8 Conclusion

The agent-memory literature has reached online insertion; maintenance of inserted structure stops, at best, at repartitioning a single top layer. On a real-usage workload, undisciplined insertion produced preferential attachment (a 64% magnet), fragmentation, and flatness; ablation establishes the guard layers as a defense in depth, and a mid-study backbone change sharpened the point: unguarded behavior varies by generation, while the deterministic bound and the curation loop hold structure within the health gate in both regimes. Structural-health metrics and writer attribution measure this axis; same-condition comparisons show the advantage on the structural and freshness axes. *Accumulation is maintenance, not placement*: the maintenance loop deserves to be a first-class design object alongside the insertion algorithm.

REFERENCES

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Mikhail Burtsev, and Evgeny Burnaev. AriGraph: Learning knowledge graph world models with episodic memory for LLM agents. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 12–20, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999.

Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchi Li, Feiyu Xiong, and Zhiyu Li. HaluMem: Evaluating hallucinations in memory systems of agents. *arXiv preprint arXiv:2511.03506*, 2025.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009.

Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. RoboMME: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations (ICLR)*, 2026a.

Zhanghao Hu, Qinglin Zhu, Runcong Zhao, Di Liang, Hanqi Yan, Yulan He, and Lin Gui. Beyond RAG for agent memory: Retrieval by decoupling and aggregation. *arXiv preprint arXiv:2602.02007*, 2026b.

Shuo Ji, Yibo Li, and Bryan Hooi. Memory is reconstructed, not retrieved: Graph memory for LLM agents. In *International Conference on Machine Learning (ICML)*, 2026.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E. Gonzalez. Sleep-time compute: Beyond inference scaling at test-time. *arXiv preprint arXiv:2504.13171*, 2025.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 13851–13870, 2024.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2023.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for LLMs. In *International Conference on Learning Representations (ICLR)*, 2025.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*, 2024.

Alina Shutova, Alexandra Olenina, Ivan Vinogradov, and Anton Sinitsin. Evaluating memory structure in LLM agents. *arXiv preprint arXiv:2602.11243*, 2026.

Haoran Sun, Shaoning Zeng, and Bob Zhang. H-MEM: Hierarchical memory for high-efficiency long-term reasoning in LLM agents. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 10014–10037, 2023.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Sizhe Zhou, Sheldon Yu, Hui Wei, Junda Wu, Siru Ouyang, Yizhu Jiao, Shijia Pan, Julian McAuley, Yu Zhang, Tong Yu, and Jiawei Han. Filesystem-based memory for LLM agents: Organization, evolution, and sustainability. *arXiv preprint arXiv:2607.26637*, 2026.

Reproducibility Statement

As a reproduction package we release: the evaluation harness (runner, structural-metric calculator, oracle injector, and the deterministic regrading script used for all aggregates), the synthetic fixture suites in both languages (e1–e8 and k1–k8, after anonymization checks), the MemTree reimplementation (formula and hyperparameters stated), execution scripts for the external baselines (Mem0 and A-MEM—public repository versions and clone procedures documented), figure-generation scripts, and a full appendix of LLM prompts including the response-language adapter (Appendix I). For every experiment, the backbone model, run count, and scoring rules (deterministic expected/forbidden string matching) are stated in the body and appendix. The real-usage corpus is withheld for privacy and business-information reasons in both its original Korean and translated English forms; we bound the reproducible scope to its statistics (length, topic count, interconnectivity) plus the synthetic fixtures, and state this explicitly.

Ethics Statement

The evaluation workloads are self-generated and contain no real-user data. The real-usage-derived corpus is handled as source-withheld. The user burden of HITL ask-back is bounded by the pending queue and TTL and reported as a measured, backbone-dependent trigger rate (§6.5), and the record-immutability principle structurally prevents unauthorized alteration of user records. We include in the discussion the privacy implications of long-term personal memory accumulation—the surface by which sensitive information could be exposed through structuring and summarization.

AI USE STATEMENT

Large language models were used in this work in four capacities, which we disclose including items in the recommended-disclosure category. (1) As the object of study: Gardener is an LLM-based memory system, and the backbone models used in every experiment are named in the body and appendix together with the measurement conditions. (2) Engineering assistance: the system and the evaluation harness were developed with LLM coding assistance; all write paths of the resulting system are deterministic code, the harness scores by deterministic string matching, and both were reviewed by the authors. (3) Synthetic data: the synthetic evaluation fixtures were authored with LLM assistance and verified item by item by the authors (expected and forbidden answers fixed by hand); the English mirrors of the fixture suites and of the workload corpus were likewise translated with LLM assistance under machine-checked structural isomorphism and reviewed item by item by the authors. (4) Writing assistance: the manuscript was drafted in Korean and translated and polished into English with LLM assistance; the claims, the related-work positioning, and every reported number were verified by the authors against primary sources and measurement logs (venue-level verification of citations, and a claim-boundary re-verification pass before submission). No LLM contributed to the mathematical formalization beyond editorial drafting, and no content was included without author verification. The authors take full responsibility for the entire content of this paper.

A EXTENDED OUT-OF-DOMAIN AND LONGMEMEVAL RESULTS

This appendix reports in full the three out-of-domain probes summarized in §6.3.

LoCoMo pilot. As an out-of-domain probe under extreme conditions we use a LoCoMo pilot (1 of the 10 public dialogues, a 3-session subset of English personal chat). With lexical retrieval alone, the 22 recall-type questions are a total loss (0/22, all abstentions without lookup). Re-running with hybrid recall and response discipline brings this to 10/22 under manual bilingual judging (12/22 lenient), and a control cell (channel off, response discipline only) records 8/22, decomposing the contribution: response discipline dominates and the embedding channel adds $+2$. In this corpus the English source text is stored verbatim, so the lexical channel already hits; the substance of the cross-lingual gap was not retrieval failure but response avoidance and detail selection. Remaining failures change character to relative-time anchor errors and detail-level abstention, and one erosion by embedding semantic neighbors under the persona-substitution adversarial reappears (§7). Harness adapters for response language and time anchoring (answer-in-question-language instruction, session timestamp backdating) were validated in a further pilot, where the relative-time anchor errors disappear and the score improves to 12/22.

Full-dialogue extension and the active recall loop. We then extended the probe to one full dialogue (19 sessions, 342 messages, 199 questions), with two results. First, the structural axis generalizes out of domain: over the long-run accumulation of 342 English chat messages, magnet 12%, depth 3, and unclassified 0% meet every §3.3 gate (the 33–39% at the 3-session point descends to 12% as accumulation and curation proceed). Second, the remaining recall bottleneck stalls at scale: on 61.8% of answerable questions the reader chose to abstain even after lookup—a limitation of response strategy, not of the retrieval channel, since selection over a wide candidate set exceeds what a single-requery discipline can handle. This measurement motivated the active recall loop (§4.6), whose effect we attribute with a controlled intervention that re-runs only the queries on the same accumulated state: with the loop, the avoidance rate halves from 61.8% to 25.0%, the token-F1 pass rate (≥ 0.5) on answerable questions rises from 10.5% to 29.6%, and abstention on unanswerable questions (34/47) is unchanged—multi-step recall did not erode abstention. A control cell in the same code with only the loop disabled reproduces the original measurement at 57.2%, attributing the improvement to the loop, and the in-house suite regression is harmless (e2e 94.1%, single run on the Korean suite, against its 90.2% median—Appendix G).

Full-corpus scale-up. After the intervention validation we scale to the full corpus (10 dialogues, 4,972 messages, 1,984 questions). The avoidance rate is 14.8–32.5% per dialogue (21.1% overall), reproducing the intervention effect on all ten dialogues; token F1 on answerable questions averages 0.331 (33.0% pass rate), and adversarial abstention holds at 81.8% (72–88% per dialogue). On the structural axis, all ten dialogues meet the §3.3 gates (magnet 5–26%, unclassified 0%, depth within

5), establishing that the structural health of §6.2 is a property of the corpus at large rather than an accident of one dialogue. The weakest remaining axis is open-domain composition questions (token F1 0.228), whose character is compositional reasoning over retrieved facts rather than recall breadth. These figures come from our own harness (official-style token-F1 reproduction, keyword abstention judging); the subset, scorer, and backbone differ from other work, so we do not compare the numbers across papers.

Original LongMemEval. Finally, we run the original English LongMemEval on the full knowledge-update and abstention subsets (102 questions: knowledge-update 78, abstention 30, 6 duplicates excluded). Each question initializes an isolated store, ingests the haystack sessions in full (47.7 sessions on average, on the order of 122K tokens per question) session by session, then queries through a separate reader channel: the same path as the noisy-history port (e8), scaled up. By LLM judge, knowledge-update scores 38/72 (52.8%), abstention 14/30 (46.7%), and 52/102 overall; since the backbone and judge (both `gemini-flash-latest`) differ from the official configuration (the original paper judges with gpt-4o), these are reported under our own conditions and not placed beside other papers' numbers. The failure anatomy is telling. Of the 34 knowledge-update failures, 44% (15) come not from recall but from the writer's recording selectivity: a secretary-style system that judges what is worth recording did not record needles embedded in free-form chat, and this difference in conditions from the port fixtures, which contain explicit recording requests, set the recall ceiling. Abstention failures were mostly assertions extrapolated from recorded adjacent facts; one question failed repeatedly at the response stage and is conservatively scored as wrong. As a by-product, all 102 independent accumulations reconfirm the structural axis: median magnet 5.7%, median unclassified 0%, and the structural gate (magnet $\leq$30%) met on 100 of 102—the two exceptions are trivial occupancy in minimal trees of 4 and 7 nodes.

B FIXTURE SUITE DETAILS

The eight fixtures of §5 are: temporal (schedule changes and cancellations), entity (namesake and organization resolution), locus (location tracking), procedural (procedures and rules), multi-topic (structural-pathology regression), abstention (where "unknown" or expired information is the correct answer), knowledge-update (multi-step changes, cross-updates, and original-value temporal queries, with stale-value citations tallied separately as a forbidden failure rate), and longmem-haystack (a port of the LongMemEval style: knowledge updates and abstention re-tested inside a noisy history of 24 messages across 7 topics, with remote update chains at placement distance 10–18 translating the multi-session condition). The English suite (e1–e8) mirrors the Korean originals (k1–k8) one-to-one: message counts, supersede chains, question types, negative flags, and forbidden sets are machine-checked identical; entities are neutral pseudonyms (reserved 555 phone range), amounts are converted to USD, and abstention questions grade against a uniform abstention key set augmented with per-question topic keys, calibrated against false positives before the suite was frozen.

C ABLATION FIGURE

D CURATION RUN DETAIL

E MODEL PORTABILITY MEASUREMENTS

Under controlled conditions (same corpus, isolated stores, same procedure) we compare backbones in two rounds. On the Korean workload, the original flash-class backbone (Gemini 2.5 Flash) and an opus-class backbone (Claude Opus 4.6) both meet the structural-health gates (magnet 24% vs. 20–25%, fragmentation 0, unclassified $\leq$10%), with the larger model using hierarchy formation (depth-1 share 42% vs. about 25%) and ask-back (3 vs. 0) more actively. On the translated English workload with the current flash-class backbone as reference (magnet 31%, unfiled 33%, ask-back 9, flat), the opus-class backbone again meets the gates and again spends capability on structure: magnet 22%, unfiled 8%, ask-back 3, and a two-level hierarchy on the same 31 messages. Across both rounds the pattern is stable: the guards guarantee the floor; model capability adds headroom, and generation shifts surface as style, not as gate violations.

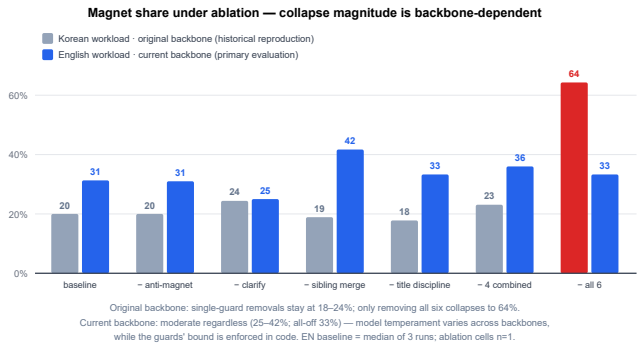


Figure 2: Magnet occupancy under ablation, on both backbones. Original backbone (gray; Korean workload): single ablations and the 4-way removal stay at 18–24%, and only removing all six layers reproduces the 64% collapse. Current backbone (blue; English workload, primary): moderate regardless (25–42%; all-off 33%). The collapse magnitude is backbone temperament; the guards' bound is enforced in code. EN baseline is the median of 3 runs; ablation cells are single runs.

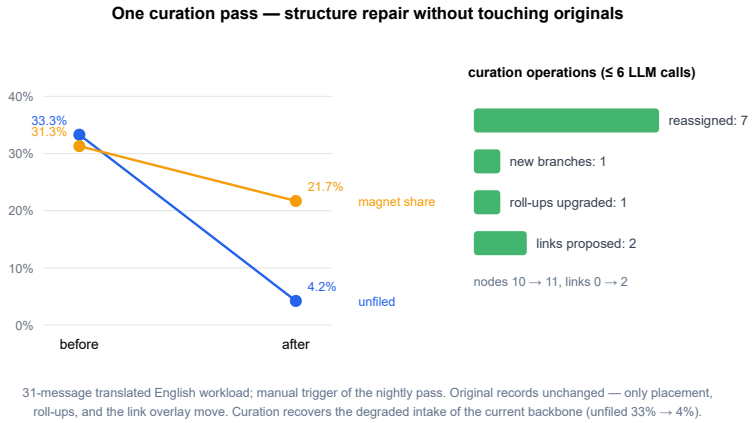


Figure 3: One curation run, before and after (English workload, current backbone). Unclassified 33.3%→4.2%, magnet 31.3%→21.7% (left); operation counts (right). Original records are untouched. The pass repairs the noisier intake of the current backbone (§6.2).

F Baseline comparison conditions

The main comparison cells (hybrid, flat, and the three external systems in Table 1; the Gardener and MemTree paper-value configurations in Table 2) are medians of 3 runs; only the MemTree sensitivity variants (θ_0=0.3, λ=0.1) remain single runs for trend reading. QA for the external systems is single-call injection, which differs in interaction shape from a tool-using reader; we accept this under the each-system-default-usage principle; all readers receive the same short-answer response adapter (Appendix I). MemTree is a reimplementation (possibly underestimating the authors' implementation—defended by the sensitivity sweep and code release). A-MEM runs its official default embedder on the English workload; in the Korean-workload round (Appendix G) its embedder was swapped for a multilingual one, which we state explicitly.

G Cross-lingual results on the original Korean workload

The system was developed and originally evaluated on the Korean real-usage workload with the original deployment backbone (Gemini 2.5 Flash), which was retired upstream during the study. This

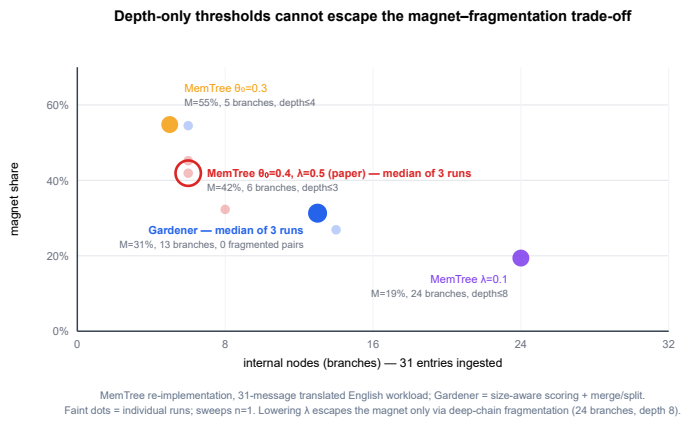


Figure 4: MemTree sensitivity scatter (English workload, current backbone). The depth axis (θ_0, λ) alone cannot escape the trade-off between magnet (upper left) and deep-chain fragmentation (lower right). Only Gardener (blue) sits outside it (magnet 31%, 13 branches, fragmentation 0; 22% after one curation pass). The two main configurations (MemTree paper values, Gardener) are medians of 3 runs, with faint dots for individual runs. Sensitivity variants (θ_0=0.3, λ=0.1) are single runs; lowering θ_0 worsens the magnet (55%, 5 branches), and lowering λ escapes it only by fragmenting into deep chains (19%, 24 branches at depth 8).

appendix preserves those measurements as the historical record and adds a same-backbone symmetric re-run that establishes language equivalence. Numbers in this appendix are **not comparable** to the primary tables (different backbone or different language); we never compare across these conditions in the body.

Language equivalence on the shared backbone. Re-running the Korean fixture suite (k1–k8) on the current backbone under the symmetric response adapter (Appendix I.6) gives oracle 98.0%, e2e 90.2%, stale-value citations 0/18 (0, 1, 0), and avoidance 5.9% (medians of 3 runs), against the English suite's 96.1%, 84.3%, 0/18, and 15.7% (Table 1); on the 31-message structure workload the Korean corpus records magnet 35%, unfiled 36%, and 11 ask-backs against the English corpus's 31%, 33%, and 9 (medians). The differences are within run-to-run spread except avoidance, where the reader avoids more on English input; correctness among answered questions is 100% in both languages. The backbone-behavior findings of §6.4 (moderate unguarded drift, liberal ask-back, flat intake) reproduce in both languages on the shared backbone, which is what attributes them to the backbone rather than to translation.

Historical results on the original backbone (Korean workload). Recall and freshness: Gardener with a lexical-only reader scored 74.5% with 5.6% stale citations; introducing hybrid retrieval lifted this to 90.2% and 0.0% (0, 1, 0), the intervention that gated the recall layer (§7); the flat control scored 84.3% and 11.1%; MemTree 88.2% and 33.3%; Mem0 88.2% and 16.7% (fluctuating 5.6–22.2% across runs); A-MEM (multilingual embedder) 94.1% and 11.1%. The +5.9-point structure contribution over flat reproduced under both lexical (68.6→74.5%) and hybrid (84.3→90.2%) conditions, concentrated in abstention (6/9→8/9) and freshness (3/6→5/6) questions. Structural health: Gardener magnet 20% (17–21) with 12–16 branches and hierarchy at depth 1–2; MemTree paper values 52% (39–55) with 5–7 branches; MemTree λ=0.1 escaped to 7% only via 29 branches for 31 leaves at 2.4× the LLM calls. The needle-condition writer gap was negative (oracle 50% vs. e2e 66.7%, median of 3): summary refinement dominated raw placement (§5). The ablation collapse belongs to this backbone (§6.4); Figure 5 shows the two system states.

Internal-language measurement. Under English input, stored node summaries remain Korean (96–100% of summaries contain Hangul, measured per cell from the store), reflecting the Korean-operating internal prompts (Appendix I); reader answers are English under the adapter. Recall on

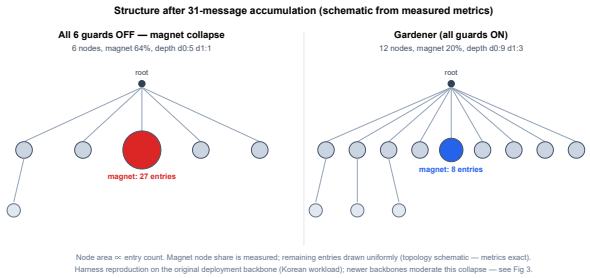


Figure 5: Structure after 31-message accumulation (schematic from measured metrics; original backbone, Korean workload). Left: all guards off—the magnet node absorbed 27 records (64%). Right: Gardener—12 branches, magnet 20%, depth-1 hierarchy. Newer backbones moderate the collapse (Figure 2).

the English suite is unaffected in aggregate (84.3% e2e), indicating that cross-lingual retrieval over Korean-summarized English originals is not the binding constraint at this scale.

H IMPLEMENTATION NOTE: CITATION SURFACES

Evidence citations must live only in the storage and browsing representations and be deterministically stripped from LLM-facing prompt surfaces (placement candidates, tree overviews), because metadata crowds out the summary window. In our measurements, recall dropped by 7 questions before this surface separation and returned to baseline after it.

I LLM PROMPT APPENDIX

Gardener operates in Korean. This appendix documents every LLM prompt surface that is active under the measured configurations, as faithful English translations by the authors; the verbatim Korean originals ship with the supplementary package. Dynamic injection slots are rendered as `<angle-bracket>` placeholders. Conditional prompt blocks tied to product features outside the scope of this paper (messenger-specific controls, background task delegation, dashboard links) are omitted here; none of them is triggered by the harness workloads. The write path itself—placement resolution, supersede marking, and all tree mutations—is deterministic code with no prompt surface (§4): the LLM only ever proposes, and a validator applies or rejects.

I.1 WRITER: INTAKE CLASSIFICATION AND PLACEMENT

Deterministically injected context. Today's date; the user message; a two-level outline of the current tree (title plus one-line summary); the top-5 placement candidates produced by the anti-preferential scorer of Eq. 1, one line each in the form "`[<id>] <path> (records <n>) [broad-branch flag] — <one-line summary>`" (evidence-citation tokens are stripped from this surface, §H; branches with $\geq$8 direct records are flagged *"broad branch — child branch recommended"*); up to three recent user placement corrections as one-line few-shots; and, when the previous turn asked the user a question (placement ask-back or amount ask-back), a pending block that binds the user's answer to a definitive update on this turn.

Core instruction (translation). "Classify the intent of the user message and output exactly one JSON object (no other text). Intents: `record`, `update`, `reorganize`, `link`, `edit_node`, `delete_node`, `delete_entry`, `query`, `stats`, `chat`. For `record`, fill: `raw_text` (the new message verbatim), `summary` (one line), `category`, `event_date` (resolve relative dates against today), `due_at`, `amount`, `tags`, `entities`. Also fill `kb_placement` as exactly one of: {`node_id`} — *only an id from the candidate list*; {`new_node: {parent_id, kind, title}`} when no candidate fits; or {`ambiguous: {candidates}`} to ask the user — plus a `confidence` of high/medium/low."

- **Title discipline.** "A branch title must be a short fixed class name — a noun phrase around the key entity or task, recommended $\leq$20 characters. No sentence-form titles; no enumerative qualifiers ('and/related/analysis/strategy'). Do not lump broad topics under one title — use separate branches per topic."
- **Hierarchy nudge.** "If a candidate is flagged as a broad branch and this record is a sub-detail of it, do not place into it directly — create a child branch under it (`new_node.parent_id` = that branch). Hierarchy extends naturally to 2–3 levels."
- **Ask-back first.** "If you cannot decide where the record belongs — several plausible candidates, only an irrelevant broad branch, or unclear whether a new branch is warranted — neither force a placement nor leave it unclassified: output `ambiguous` with the 2–3 most plausible branch titles. One ask-back is better than knowledge leaking into the wrong branch; the user's answer is bound to a definitive placement on the next turn."
- **Low-confidence deferral.** "If confidence is low and there is no candidate worth asking about, omit the placement — nightly curation retries unclassified records."
- **Freshness / supersede detection.** "If the message changes a *value* of an existing record (amount, schedule, owner), this is a new `record`, not an `update`: retrieve the old record, fill the entry with the new value, and set `supersedes_entry_id`. This covers both temporal change ('pushed to…') and error correction ('fix that to…'). The system keeps the old record intact and marks it superseded, so 'what was it originally?' remains answerable."

Deterministic resolution. The proposal is applied by code: hallucinated node ids fall back to unclassified; a low-confidence `new_node` is not created (unclassified); the depth cap promotes over-deep parents to an in-bound ancestor; an existing same-parent, same-title branch is reused idempotently.

I.2 Reader: active recall loop (retrieve)

Translation of the full base prompt (the pin-scope and ledger blocks are injected only when those features are active; neither is active in the harness):

> "You collect and select the record evidence needed to answer the question; answer writing is done by the next stage — here you *only gather evidence*. Today's date: `<date>`. Question: `<message>`. Initial search results (system-issued query from the question verbatim): `<presearch JSON>`. Procedure: (1) If the initial results already answer the question, output the final JSON without further search. (2) Otherwise re-search with `journal_query` from *changed angles* (at most 3 calls, never repeating a query): shorten to 1–2 key nouns; try synonyms and script variants (Korean↔English, aliases, abbreviations); try person, organization, place, or item names alone; add `date_from`/`date_to` when a period is implied; do not set a category filter; for past-state questions ('originally/at the time') set `include_invalid=true`. (2-1) When a topic (branch) search fits better, use `kb_search`, and descend with `kb_node_view`. (3) If details beyond a summary are needed, fetch the original with `journal_get(id)`. (4) Finally output exactly one JSON object: `{"evidence": [{"id", "date", "fact"}], "found": true|false, "notes"}`. Evidence must contain only records relevant to the question (max 8); `fact` must state only what the record says — no guessing or filling in. If multi-angle search finds nothing relevant, output empty evidence with `found=false`."

I.3 Reader: response synthesis (respond)

The respond stage receives the classification artifact, the persist result, and the retrieve stage's evidence JSON. Memory-relevant rules (translation; product-feature rules omitted):

- **No fabricated completion.** "Statements that an action was performed are allowed only if this turn's persist result or a tool return states it. Prior conversation is context, not grounds for completion claims."
- **Record confirmation.** "If `recorded=true`, confirm in one line; if placed (`kb.placed=true`), append the branch name; if a value was superseded, append a note that the previous value was marked outdated."

- **Ask-back presentation.** "If placement was ambiguous, state that the record itself is saved and ask which branch it belongs to, offering the candidates as buttons."
- **Evidence-grounded answering.** "For queries, answer from the collected evidence — the loop has already searched from multiple angles. Fetch originals with `journal_get` when the exact text is needed; never reconstruct an original from summaries or conversational memory. If evidence came back `found=false`, say that no record exists — do not re-search, do not guess."
- **Grounding priority.** When user-selected context exists, the order is: referenced chip $\geq$ pinned active context $>$ in-scope recall evidence $>$ out-of-scope evidence (ask, do not assert) $>$ absence answer (§4.6).
- **Style.** Concise Korean honorific replies; internal JSON and field names must not leak into the reply.

I.4 Curation operators

Each nightly operator follows the same contract: deterministic code pre-selects the inputs, the LLM proposes in JSON, and a validator applies only verifiable proposals (§4.4). The first seven operators correspond to the seven curation steps of §4.4; the roll-up upgrade (item 8) runs after them in the same nightly pass. Translations:

1. **Unclassified-record reassignment.** Inputs: unclassified records, each with its top-3 scored candidate branches. "For each record, place it if it clearly belongs to one of its candidates; otherwise leave `null` (no forced placement). Output a JSON array of {`entry_id, node_id|null, confidence`}." Only `high`-confidence decisions whose `node_id` is inside the record's own candidate set are applied.
2. **Unclassified cluster creation.** Inputs: remaining unclassified records plus a two-level outline of existing branches. "Group $\geq$2 records of the same topic into a proposed new branch: short fixed class name ($\leq$20 chars); `parent_id` from the listed branches or null for root; do not group unrelated singletons." Groups below the minimum size are dropped; hallucinated or over-deep parents fall back to root.
3. **Title normalization.** Inputs: branch titles that look sentence-form (length or punctuation heuristics). "Rename each to a short fixed class name ($\leq$20 chars); skip items where renaming would lose information — details already live in the branch summary and records; the title is only a label." Renames are applied only to existing nodes and skipped on sibling-title collisions.
4. **Sibling merge.** Inputs: same-parent sibling groups (title, one-line summary, size). "Find siblings that denote the same task or topic but were split apart, and propose merges; only merge when clearly identical — no forced merges. Pick as canonical the branch with the most records or the most standard title; optionally propose a short `new_title`." The validator restricts merges to true siblings; records move, children re-parent, soft links re-wire to the canonical node, and the emptied node is deleted.
5. **Cross-link proposal.** Inputs: the branch list with paths and one-line summaries. "Propose at most k pairs that are not the same topic but closely related enough to deserve a reference link; never propose parent–child pairs (the tree already expresses them); no forced links — an empty list is fine; label each link in a few words." Existence, parent–child exclusion, duplicate exclusion, and a per-run cap are enforced in code.
6. **Procedural triage.** Inputs: records matching procedural heuristics. "Return only the ids of records that contain conversational procedure with no substantive information (requests, upload notices); exclude anything carrying a schedule, amount, fact, or decision." Confirmed ids are tagged `procedural` and excluded from summaries and recall surfaces.
7. **Oversize split.** Inputs: records of a branch that exceeds the direct-record cap. "Propose 2–5 child branches clustering the records by topic; titles follow the title discipline; every record goes to exactly one group." Proposed ids are validated; groups become child branches of the oversized node.
8. **Roll-up upgrade.** Inputs: valid (non-procedural, non-superseded) records of large branches, each line carrying its evidence-citation token. "Summarize the branch in 3–6 sentences:

> proper nouns over pronouns, absolute dates over relative ones, latest state first (reflect changes and cancellations), facts only. Keep the ⟨e:..⟩ citation token of the supporting record at the end of each sentence; write no sentence without support." Citations are validated against real record ids; hallucinated tokens are stripped (§4.4).

I.5 HARNESS JUDGE PROMPTS (LONGMEMEVAL)

The LongMemEval run (§6.3, Appendix A) uses an LLM judge; both prompts are operational verbatim (English in the original):

> QA: "I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no. [. . .] Is the model response correct? Answer yes or no only."

> Abstention: "I will give you an unanswerable question and a response from a model. Please answer yes if the model response indicates that the question cannot be answered from the available information, that no such information exists, or that it does not know. Otherwise (if the model fabricates or asserts an answer), answer no. [. . .] Does the model correctly abstain? Answer yes or no only."

All e1–e8 and k1–k8 fixture scoring, by contrast, is deterministic expected/forbidden string matching and uses no judge (§5).

I.6 RESPONSE-LANGUAGE ADAPTER

The English cells append one sentence to the reader query only (core prompts unchanged): "*(Answer in English with only the short answer phrase — no full sentence.)*" The Korean symmetric cells of Appendix G append the Korean mirror of the same sentence, so the two languages are measured under identical response conditions; external-baseline responders receive the equivalent instruction in their own answer prompts.

I.7 MEMTREE REIMPLEMENTATION PROMPTS

The reimplemented baseline (§6.1, Appendix F) uses two prompts (translation):

> Aggregate (ancestor summary update on insertion): "Here are a memory tree node's current summary and newly added content. Output only an updated summary of 1–3 sentences integrating both (in Korean; keep the source language if different). [If the node has more than two children:] Since there are many children, summarize more abstractly toward the parent topic. [Otherwise:] Produce a summary that covers both contents."

> Responder (QA over retrieved nodes): "You are the responder of a personal memory system. Answer the question grounded *only* in the retrieved memory nodes below. If the memories contain no grounds, you must answer 'there is no such record' (no guessing). [retrieved top-k nodes with similarity scores] Answer in 1–2 sentences."