E^{2}MEND: EVIDENCE-GUIDED EVOLUTION FOR MEMORY ENHANCEMENT VIA BOTTLENECK DIAGNOSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory failures are heterogeneous: evidence may be lost, unretrieved, insufficiently structured, or available while the answer model still fails; a uniform update can target the wrong bottleneck and introduce regressions. We present E^{2}MEND, a diagnosis-guided framework for bounded memory-strategy evolution: it estimates the bottleneck, mutates versioned strategy artifacts through a bounded LLM evolver under a locked runtime, and adopts candidates via paired evaluation, negative evidence, and rollback. It matches failure modes to bounded edits—extraction or provenance for ingestion gaps, support selection for retrieval gaps, typed state for state construction, and no memory mutation for answer-model limitations. On a fixed 1,540-question LoCoMo slice, a staged trajectory raises accuracy from 0.756 to 0.947 (final paired checkpoint: 16 repairs, 0 regressions); on MemBench, matched strategies yield a descriptive $0.770 \rightarrow 0.863$. We frame these as diagnostic and strategy claims on fixed slices, not public-task superiority.

1 INTRODUCTION

Long-term memory is central to language-agent systems, but an incorrect answer does not reveal what should change; it exposes a memory-attribution gap between an endpoint failure and the answer-bearing stage where an edit is needed. Evidence may be missing, unretrieved, insufficiently structured, or available to an answer model that still fails; these four cases call for different, sometimes opposing, fixes. More retrieval cannot recover discarded information; adding more retrieved chunks cannot resolve a state-construction gap; and memory mutation alone does not repair an answer-model limitation.

Instead of a uniform memory strategy, such as a Mem0-style extraction pipeline Chhikara et al. (2025) or a retrieval-only RAG pipeline, we study whether a memory system can diagnose the bottleneck, select a matched strategy, and adopt it without regressing previously correct behavior. E^{2}MEND implements this with five mechanisms: Layer-A routes failures along the answer-bearing path (ingestion, retrieval, state construction, and answering) to the earliest actionable stage; a fixed policy router restricts candidate paths; a bounded LLM evolver proposes concrete, schema-legal edits from the diagnosis, the implicated paths, and prior attempts; a locked runtime leaves only versioned strategies mutable; and paired non-regression checks control adoption. The runtime executes ingestion, storage, retrieval, reranking, context assembly, structured-state execution, and answering, while the versioned strategy controls when and how those capabilities are invoked. Evidence granularity is diagnosis-conditioned, not fixed: source windows preserve retrieved evidence, whereas typed state targets post-retrieval state construction.

We evaluate on fixed slices; candidate adoption uses paired, non-regression comparisons, while configuration sweeps and reference rows are separately scoped. The experiments in Section 6 answer three questions: can the probes produce actionable bottleneck hypotheses, do matched strategies improve the intended slice, and does paired verification filter no-effect or regressive candidates? Section 2 formalizes the fixed/mutable boundary, Section 3 describes strategy decomposition, and Sections 4 and 5 present diagnosis/planning and verification.

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

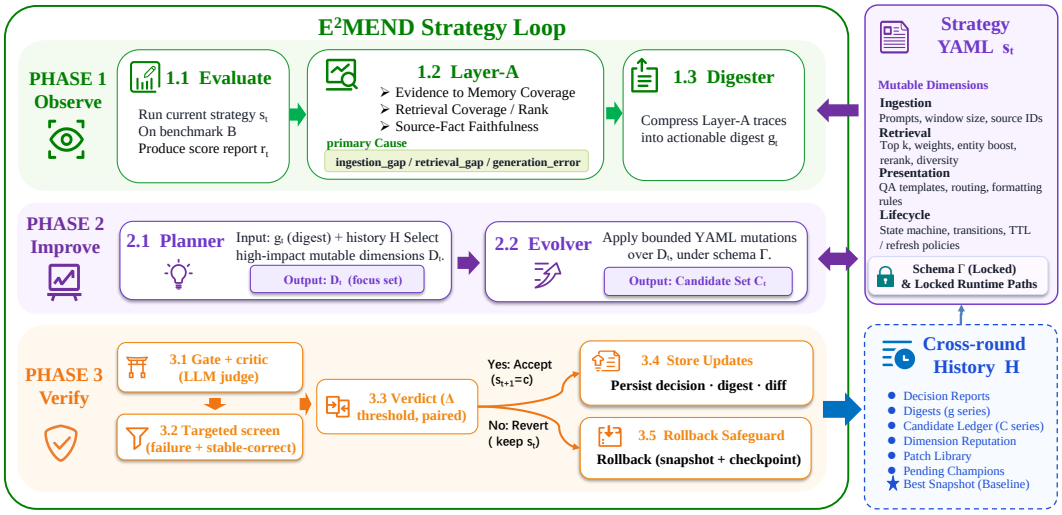


Figure 1: One E^{2}MEND round: diagnose, propose a schema-legal strategy diff, verify it with paired checks, then apply strict or practical acceptance, retain as pending, reject, or, on later drift, roll back.

This paper makes three contributions. First, we formulate memory-strategy evolution as a four-way bottleneck diagnosis—ingestion, retrieval, state construction, or answer-model failure—each matched to a distinct bounded strategy family, not a single universal update. Second, we introduce a locked-runtime, versioned-strategy architecture for bounded, attributable, and checkpoint-reversible mutations. Third, we develop a noise-aware adoption protocol based on targeted replay, paired non-regression, negative evidence, and rollback.

2 Problem Setup: What Can Change?

Notation and mutable artifact. Let F be the locked runtime (ingestion, storage, retrieval, reranking, context assembly, structured-state execution, and answering), Γ the schema that defines legal strategy fields, $\mathcal{B}$ the fixed evaluation slice (the questions and scoring evidence used for comparison), s_t the served strategy at round t, and $\mathcal{H}_t$ the persistent history of candidate diffs, paired outcomes, rejected directions, and rollback checkpoints.

For each question i, let $Q_i \in \{0, 1\}$ be a fixed per-question state-construction predicate. It equals 1 when the answer must be assembled from typed state through comparison, aggregation, or temporal update, and 0 when the evidence should be answerable directly from retrieved source text. The predicate only routes generation-level failures; its assignment procedure is currently specified semantically, not operationally, and we make this boundary explicit before revisiting it in Appendix G. Let $Q = (Q_i)_i$ denote the fixed predicate map. Then F, Γ, $\mathcal{B}$, and Q are fixed, while s_t—versioned parameters, prompts, routing rules, and feature switches—is the only mutable artifact. Fixing everything but s_t keeps each gain or regression attributable to an explicit strategy diff.

In one round, E^{2}MEND diagnoses failures on $\mathcal{B}$, proposes legal edits to s_t, screens the resulting candidates, and compares the selected candidate with the current strategy on the configured pairing key (question key or run–question key; Section 5). A candidate is promoted to s_{t+1} only under StrictAccept or PracticalAccept; a Pending candidate is retained for later confirmation, and otherwise the system keeps s_t and records the failed direction. If drift later occurs, the system restores a prior checkpoint. Figure 1 expands this process.

One-round decision. Given the failure digest and prior history, the evolver generates schema-legal strategy edits and applies them to s_t to produce a candidate list. Search scores determine evaluation order only. A candidate that reaches full verification receives one of StrictAccept, PracticalAccept, Pending, or Reject, as defined in Section 5; before thresholds are applied, four checks must pass: legality under Γ, a positive full-slice net flip count, no protected-category regression beyond tolerance, and a stable-correct replay count within its regression limit.

Table 1: Representative evolvable strategy slices.

Group	Example fields	Guardrail
Ingestion	prompt version, max facts per window, dedup threshold, conditional ingest patch	Provenance and no duplicate turns
Retrieval & state	top_k, entity threshold, rerank size, source window, *intent routing* (reasoning-type parser, top-k multiplier), *structured state* (slot schema, typed lookup)	Legal ranges, fallback rules, schema validation
Lifecycle	pruning threshold, decay half-life, feedback tracking	No deletion of active constraints
Presentation	citation mode, synthesis prompt, output format	Prompt-integrity checks
In-phase	enabled toggle, provision interval, short-term context window	Must not interfere with long-term retrieval
Feature toggles	failure correction, conflict handling, conditional triggers, graph extraction	Declared trigger scope and rollback key

Fixed authority boundary. E^{2}MEND may instantiate only fields declared by Γ; it cannot modify F's code, the storage schema, $\mathcal{B}$'s composition, or the judge policy. A change outside this boundary requires manual review. We choose this separation because tuning the runtime and strategy jointly would couple improvements and obscure which change caused a gain or regression; a versioned strategy keeps every accepted gain attributable to an inspectable field-level diff.

3 Framework and Strategy Decomposition

This section defines the evolvable surface and the vocabulary used later to map bottleneck diagnoses to bounded edit families. Sections 4 and 5 formalize diagnosis and verification; the experiments test whether this boundary produces attributable, non-regressive improvements. The runtime provides capabilities, while the strategy controls when and how to invoke them: activation, scope, thresholds, and prompts are versioned fields, not hidden runtime edits.

Evolvable strategy space. The strategy artifact contains versioned extraction prompts, fact budgets, deduplication thresholds, retrieval weights, rerank limits, source-window sizes, and feature toggles. A source window is the contiguous original text retained for retrieval or answering; *session chunks* align source windows to dialogue sessions in the LongMemEval protocol. Prompt edits are permitted only when they are versioned; implicit prompt edits inside framework code remain outside the boundary. Declarative artifacts can be searched, bundled, reverted, and compared across rounds.

Table 1 organizes representative evolvable dimensions into six conceptual strategy sections. The underlying implementation may expose additional schema-validated fields, but all candidates remain within the declared strategy boundary.

Intent routing targets retrieval-bound cases, structured state targets state-construction cases, and conditional ingest patches address ingestion. Below, *query routing* denotes the retrieval-side use of intent-routing fields, and *typed state* denotes answer-time execution of structured-state fields. Structured-state fields are answer-time views over the locked store; a candidate may materialize existing evidence fields but cannot alter the persistent storage schema.

Locked base harness. The locked harness contains capability code paths (ingestion, storage, retrieval, reranking, context assembly, structured-state execution, answering). E^{2}MEND can configure these capabilities through strategy artifacts, but cannot rewrite the harness.

4 Diagnosis-Guided Strategy Selection

Each round follows an Observe–Improve–Verify loop ordered around one question: *where does the answer-bearing path first fail?* Diagnosis estimates the stage, strategy selection maps it to a bounded family and the evolver proposes a concrete edit, and verification decides whether to adopt it. Candidate generation is restricted to schema-legal primitive or bundle moves, and no strategy is hot-reloaded before targeted screening and a full paired non-regression check.

Operationally, E^{2}MEND keeps three evidence streams explicit: per-failure diagnosis records the first failing stage; a bounded digest summarizes dominant failure patterns; and paired decision records preserve which edit was tried, what changed, and why it was adopted or rejected. Together they make search auditable without expanding the mutable runtime surface.

Algorithmic decomposition. Diagnosis and digest compression turn failure evidence into a bounded focus summary g_t; a bounded LLM evolver proposes schema-legal edits from g_t, the eligibility mask, and history; candidate verification applies gates and paired checks; and round orchestration handles execution, logging, and rollback. Candidate search orders these proposals with upper confidence bound (UCB) selection described below. Detailed pseudocode is in Appendix Algorithms 1–3.

4.1 Layer-A Diagnosis

Layer-A routes each failure to the earliest actionable failing stage and returns one of *ingestion gap*, *retrieval gap*, *generation error*, or *uncertain*. The policy router then refines *generation error* into *state-construction gap* or *answer-model gap* using Q_i. Appendix Algorithm 1 checks evidence-to-memory coverage, retrieval rank, and then an LLM-assisted source/fact faithfulness judgment. The first two checks are deterministic only when evidence identifiers, stored records, and retrieved records are fixed; Layer-A is therefore a routing signal, not definitive causal attribution. Diagnosis selects *what to try*, while paired verification decides *whether to adopt*. A worked lifecycle example is provided in Appendix B.1.

Digest compression turns these per-item routes into a bounded focus summary g_t containing severity and failure-type counts, representative examples, and implicated strategy dimensions. This lossy routing signal is not a new judge; candidate generation consumes it with the eligibility mask, while paired verification retains authority over adoption.

4.2 Bottleneck-to-Strategy Decision Rule

The policy router maps each diagnosis to a strategy family before any candidate is generated. Table 2 summarizes the central design choice: evidence granularity is selected by failure mode, not by optimizing one aggregate representation. Source windows preserve lexical and local context for retrieval-bound cases; typed state externalizes fields needed for comparison, aggregation, or temporal update. Neither is universally superior, and both remain subject to verification.

Table 2: Bottleneck diagnoses and matched strategy families.

Diagnosis	Observable condition	Eligible strategy family	Disallowed shortcut
Ingestion gap	No faithful item is stored, or extraction loses critical detail	Extraction, provenance, conditional ingest, deduplication, or fact budget	More retrieval only
Retrieval gap	A faithful item exists but is not retrieved	Source windows, query routing, top_k, reranking, or support selection	Compressing the evidence
State-construction gap	Retrieved evidence requires comparison, aggregation, or temporal update	Typed state and answer-time state lookup	Adding chunks by default
Answer-model gap	Retrieved evidence is judged sufficient, but the answer remains wrong	Manual review of answer prompt, model, or protocol	More memory mutation
Uncertain	Coverage or faithfulness cannot be established	Conservative replay or manual review	Broad multi-field edits

Layer-A produces a low-level evidence diagnosis λ_i, which the policy router maps to a bottleneck b_i and, when available, a sub-cause ξ_i. For failed item i, the router applies the fixed predicate Q_i and routes the item to state construction only when $Q_i = 1$; the eligibility filter suppresses typed-state paths when typed state is not schema-legal for that item. The mapping is

$$b_i = \begin{cases} \texttt{state_construction_gap}, & \lambda_i = \texttt{generation_error} \wedge Q_i = 1, \\ \texttt{answer_model_gap}, & \lambda_i = \texttt{generation_error} \wedge Q_i = 0, \\ \lambda_i, & \text{otherwise.} \end{cases} \tag{1}$$

Equation 1 is a routing rule, not an adoption rule. Items in the state-construction row permit typed-state interventions over evidence-supported fields; items in the answer-model row default to manual

review, not memory mutation. This split distinguishes a state-construction gap from an answer-model limitation without reclassifying either as missing evidence.

As an illustration, a failed question whose gold facts are stored at rank $r = 35$ under $top_k = 30$ is routed to retrieval-side changes, not extraction; a failed question whose retrieved facts are complete but require temporal comparison is routed to typed state only when $Q_i = 1$. These routes determine what may be tried, not what will be adopted.

The implementation turns this table into a hard eligibility mask. For the set $\mathcal{W}_t$ of items wrong under the current strategy, bottleneck b_i, and optional sub-cause ξ_i, the candidate strategy paths are

$$\mathcal{E}_t = \mathrm{Filter}_{\Gamma,\,\mathrm{impl},\,\mathcal{H}_t}\left(\bigcup_{i\in\mathcal{W}_t} \mathcal{R}(b_i, \xi_i)\right). \tag{2}$$

Here $\mathcal{R}$ maps each diagnosis to the strategy paths in Table 2; the filter keeps only schema-legal, implemented paths and suppresses directions exhausted in $\mathcal{H}_t$, with high-confidence sub-causes applied before the broader bottleneck route. A retrieval gap with unknown rank does not automatically open top_k; that path requires an observed or inferred out-of-window rank.

4.3 Candidate Generation and Hypothesis Selection

Given the digest g_t, the eligible paths $\mathcal{E}_t$, and the history $\mathcal{H}_t$, the evolver proposes a small set of concrete strategy diffs—each a primitive scalar move or a multi-field bundle, with an impact, rationale, and risk—under the prompt in Appendix B.2. The planner then converts g_t and $\mathcal{E}_t$ into a focus set D_t keyed by strategy path and direction (e.g., `retrieval.top_k::increase`); inactive, refuted, confirmed, or exhausted nodes are skipped. For an active direction d with visits $n_t(d)$, parent visits N_t, and mean normalized paired outcome $\bar{r}_t(d)$, selection uses

$$\mathrm{UCB}_t(d) = \begin{cases} +\infty, & n_t(d) = 0, \\ \bar{r}_t(d) + \eta(a_t)\sqrt{\dfrac{\log \max(N_t, 1)}{n_t(d)}}, & n_t(d) > 0, \end{cases} \tag{3}$$

where actionability $a_t \in [0, 1]$ derives from failure severity, persistence, and distance to target, and $\eta(a_t) = \eta_0\,\mathrm{clip}(1.5 - a_t,\, 0.5,\, 1.5)$ for positive scale η_0. Unvisited directions are placed at the front of the queue; UCB only orders diagnosis-derived directions, and adoption authority remains with paired verification (Appendix Algorithm 2).

4.4 Direction Policies

The evolver's proposals are restricted to two shapes. Primitive scalar mutations move one focused dimension by a single diagnosis-selected step within its legal interval; they are a deterministic, easy-to-attribute fallback and the default for a narrow implicated failure surface. Strategy bundles change multiple related fields under one declared hypothesis and are compared atomically with their single-field parent.

5 Noise-Aware Verification

Direction policies determine *what* to search; verification determines *whether evidence is strong enough to adopt*. Automatic memory evolution is vulnerable to noisy judges, unequal question difficulty, and category seesaw effects. The adoption decision uses only within-unit paired flips, so unchanged units contribute no evidence. E^{2}MEND therefore verifies each candidate in stages: reject invalid diffs cheaply, test implicated failures and stable-correct controls, then run a full paired comparison before adoption. Appendix Algorithm 2 summarizes this decision.

5.1 Staged Paired Verification

Fail-fast gates and critic. Before model evaluation, structure and range gates reject schema-external or out-of-bounds edits; prompt-integrity gates preserve required variables, output markers,

provenance clauses, and grounding instructions; and trigger-scope gates require conditional patches to declare activation and rollback scope. An independent critic may block duplicate, risky, or over-broad candidates but cannot override a failed hard gate. These checks reduce evaluation cost; they are not evidence of task improvement.

Targeted replay. For a subset A of aligned comparison units, let $\mathcal{I}_A(c)$ contain baseline-wrong/candidate-correct keys and $R_A(c)$ the baseline-correct/candidate-wrong keys, with $\Delta_A(c) = (|\mathcal{I}_A(c)| - |R_A(c)|)/|A|$; subscripts are omitted for the full slice, and Δ_k denotes protected group k. Let $\mathcal{T}_t$ be the failures implicated by the current digest and $\mathcal{S}_t$ the stable-correct questions that should not change. The candidate passes targeted screening only if

$$\text{Screen}(c) = 1 \quad \Longleftrightarrow \quad \Delta_{\mathcal{T}_t}(c) > 0 \ \wedge \Delta_{\mathcal{T}_t \cup \mathcal{S}_t}(c) > 0 \ \wedge |R_{\mathcal{T}_t \cup \mathcal{S}_t}(c) \cap \mathcal{S}_t| \leq \tau_{\text{stable}}. \quad (4)$$

The first condition requires a positive net effect on implicated failures, not merely on the combined target-and-stable set. This stage detects candidates that repair a local issue while damaging previously solved behavior, and avoids a full-slice run for clearly harmful edits.

Paired non-regression. Full evaluation aligns baseline and candidate outputs by pairing key (question key for single runs, run–question key for pooled reruns). Let $w = |\mathcal{I}(c)|$ be baseline-wrong/candidate-correct flips, $\ell = |R(c)|$ the reverse flips, and n the number of aligned units; the paired effect and one-sided exact McNemar probability are

$$\Delta(c) = \frac{w - \ell}{n}, \qquad p_{\text{McN}}(c) = 2^{-(w+\ell)} \sum_{j=w}^{w+\ell} \binom{w+\ell}{j}. \quad (5)$$

These statistics use only within-unit flips. A protected group is any prespecified partition (e.g., question type) whose regression should not be hidden by an aggregate gain, and protected categories require $\Delta_k(c) \geq -\epsilon_k$. When the judge path is stochastic, repeated passes use majority or rerun-stability checks.

Decision outcomes. After the stable-replay and protected-group gates pass, strict/practical/pending/reject outcomes follow fixed thresholds on $\Delta(c)$ and, for strict acceptance, on $p_{\text{McN}}(c)$ (Appendix B). The exact test is confirmatory only for a prespecified candidate; unadjusted p-values after adaptive multi-candidate search are exploratory unless confirmed on untouched data or with a multiplicity correction.

5.2 Observed-Slice Property and Rollback

Observed-slice acceptance invariant. By definition, strict acceptance on aligned keys passes the stable-replay gate and satisfies $\Delta(c) \geq \tau_{\text{full}}$, $p_{\text{McN}}(c) < \alpha$, and $\Delta_k(c) \geq -\epsilon_k$ for every protected group. This invariant concerns only the evaluated slice. Judge errors, incomplete labels, adaptive selection, and distribution shift preclude a population or deployment guarantee.

Negative evidence and rollback. Rejected, pending, and accepted candidates all update persistent history. Rejections lower future preference for repeated weak directions; accepted candidates remain checkpoint-reversible. If current serving accuracy falls more than the configured rollback threshold (three percentage points, 3 pp, in the reported setup) below the best accepted checkpoint, E^{2}MEND restores that checkpoint. This provides a safeguard against cumulative drift that a single-round verifier may not expose.

5.3 Claim Scope and Reporting Conventions

We present only diagnostic claims (bottleneck hypotheses) and strategy claims (paired improvement on a fixed slice), not benchmark claims of public-task superiority. Cross-paper scores are comparable only when backbone, judge, protocol, variance, and significance are controlled; each result table states its row scope.

When a flip count is reported as $w/-\ell$, w counts baseline-wrong/candidate-correct flips and ℓ counts reverse flips under the configured key; the final LoCoMo count is question-key-level, while the pooled MemBench count is run–question-level.

Table 3: Results by fixed slice, with per-row result status and claim scope.

Fixed slice	Metric	Baseline	E^{2}MEND	Δ	Result status / claim scope
LoCoMo, $n = 1540$	accuracy	0.756	**0.947**	$+0.191$	Staged development; pending-tier final paired checkpoint
HotpotQA distractor, $n = 300$	both@2 retrieval	0.320	**0.637–0.657**	$+0.317$–0.337	Config sweep; not independent reruns
MemBench, $n = 200$	accuracy	0.770	**0.863**	$+0.093$	Three paired runs; 600 run–question pairs
LongMemEval-S QA, $n = 470$	accuracy	–	0.950	–	Reference-only

6 EXPERIMENTS

We evaluate on four primary fixed evaluation slices: the 1,540 non-adversarial LoCoMo questions Maharana et al. (2024), the first 300 HotpotQA distractor questions Yang et al. (2018), the 200-question MemBench slice Tan et al. (2025), and the 500-question LongMemEval-S QA split Wu et al. (2025), whose displayed QA metrics use the 470 non-abstention questions after the official 30-instance abstention exclusion.

We additionally use the 500-question LongMemEval-M retrieval split as a diagnostic evidence-preservation probe. Its per-question histories contain roughly 500 sessions, and its displayed retrieval metrics use the 470 non-abstention instances. Throughout, "the LoCoMo slice" and similar phrases refer to these fixed slices.

For HotpotQA, both@2 denotes retrieval success only when both gold paragraphs appear among the top two retrieved groups. The HotpotQA slice is fixed at development time; its dataset version/order and 300-question selection manifest are documented as open reproducibility gaps in Appendix 4. The GroupRank values 0.5, 0.7, and 1.0 are retained implementation labels and are not presented as formal GroupRank hyperparameter definitions.

Evaluation budget and model allocation. Each round uses one full-slice LoCoMo evaluation, up to three subset screens, and at most one extra full verification; baseline rows are reused. Detailed token and reproduction-chain counts are in Appendix C.1; Appendix Table 4 summarizes the fixed protocol.

Table 3 summarizes the fixed-slice rows and their per-row result status used below.

For the LoCoMo row, the displayed Δ is the staged endpoint difference ($0.756 \rightarrow 0.947$), not the final paired-checkpoint effect; the final paired-checkpoint effect is listed separately as $+16/-0$ (approximately $+0.010$). The LongMemEval-S QA row is reference-only: no matched comparator uses the same backbone, judge path, and protocol on the same 470 non-abstention questions, so this row is context, not paired evidence.

Evolution trajectory on LoCoMo. The trajectory is a staged development record from 0.756 to **0.947** (Figure 2(a)); only the final $0.937 \rightarrow 0.947$ update has an independently recorded paired checkpoint ($+16/-0$, question-key-level). Its full-slice paired effect is $\Delta = 16/1540 \approx 0.010$, which falls in the pending range under the stated implementation defaults (Appendix B); we therefore present it descriptively, not as strict or practical acceptance. Intermediate points are cached summaries (Appendix Table 10).

6.1 **RQ1**: CAN PROBES PRODUCE ACTIONABLE BOTTLENECK HYPOTHESES?

Question. When retrieval is removed or evidence representation is varied, do the probes separate evidence availability from remaining answer-path limitations?

Oracle evidence. Supplying complete gold source text directly as memory for all 1,540 LoCoMo slice questions leaves the unchanged answer path at 0.753 accuracy (0.875 single-hop, 0.486 tempo-

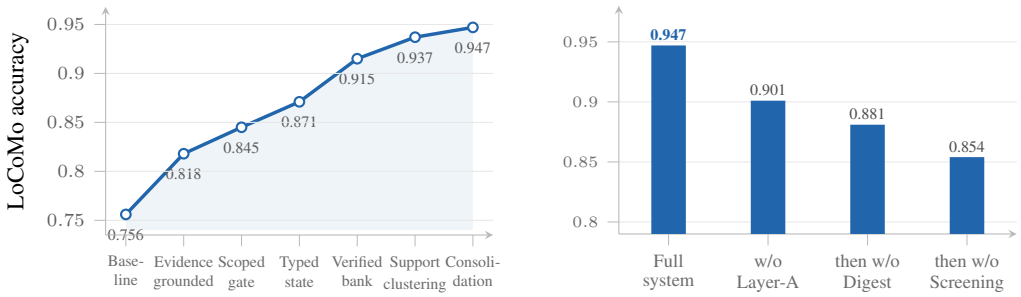


Figure 2: (a) Recorded LoCoMo checkpoint trajectory (0.756 → 0.947); the final point is a documented pending-tier paired checkpoint, and intermediate points are development summaries. (b) Cumulative pipeline ablation on the fixed LoCoMo slice (Appendix Table 9); bars remove components cumulatively left to right.

ral), 0.003 below the 0.756 staged baseline. Without paired outputs or a complete run manifest, we do not interpret this small difference. The probe supplies the source text directly, bypassing ingestion and retrieval as evaluated failure sources, but adds no structured organization; it leaves answer-path and state-construction failures for later routing, not for distinguishing a typed-state opportunity from an answer-model limitation.

Evidence preservation. On the 470 non-abstention instances of the 500-question LongMemEval-M retrieval split, session chunks reach Recall@5 (R@5) $= 0.850$ and Recall@10 (R@10) $= 0.910$. This is consistent with preserving contiguous source text for lexical answer-bearing spans (Appendix D.1, Table 5). Together, the two probes delineate evidence-availability conditions under oracle and retrieval-preserving settings; neither directly measures the causal contribution of a strategy change.

6.2 **RQ2**: Do Matched Strategies Help?

Question. Does support selection improve retrieval-isolated performance on HotpotQA, and does typed state improve the evaluated MemBench slice?

We evaluate two diagnosis-matched interventions: support selection on retrieval gaps (HotpotQA) and typed state on state-construction gaps (MemBench); Table 2 gives the full diagnosis-to-strategy policy.

On HotpotQA, the retrieval-only both@2 metric isolates support retrieval from answer extraction Zhang & Ma (2026), and the GroupRank sweep Long et al. (2026a) settings are not independent reruns (per-setting breakdown in Appendix Table 6). On MemBench Tan et al. (2025), the three runs are repeated measurements over the same 200 questions, so the pooled gain is descriptive; the separate 0.930 staged checkpoint is excluded (Appendix Table 7). The LongMemEval-S 0.950 row (Table 3) is reference-only and does not isolate either strategy (Appendix Table 8).

6.3 **RQ3**: Can Verification Reject No-Effect or Harmful Candidates?

Question. Do candidate traces and the pipeline ablation show that verification rejects no-effect or regressive directions while retaining accepted changes?

Figure 2(b) ablates the fixed LoCoMo pipeline cumulatively: removing routed diagnosis, then digest compression, then screening reduces accuracy to 0.901, 0.881, and 0.854 (consecutive drops -0.046, -0.020, -0.027). These are pipeline-sensitivity signals, not isolated causal effects.

Candidate-decision and bundle evidence are in Appendix Tables 10 and 11. Rollback was not triggered in the recorded trajectory, so it remains an implemented safeguard.

7 Related Work

Memory evaluation increasingly separates coverage, retrieval, state construction, and serving cost Long et al. (2026b); Pollertlam & Kornsuwannawit (2026), while adaptive systems evolve memory content, schemas, or operational policies Liu et al. (2026a); Jiang et al. (2026); Zheng et al. (2026). The remaining gap is the memory-attribution gap from Section 1: aggregate gains alone rarely show which stage changed and why.

Closest designs separate routing from storage Tobkin & Yang (2026) or gate schema-constrained edits under regression and rollback constraints Liu et al. (2026b;c). E^{2}MEND differs by locking the runtime and evidence slice, mutating only schema-legal strategy fields, routing by the earliest failing stage, and gating adoption on within-unit paired flips plus protected-group and stable-replay checks; Appendix F details the comparison.

8 Discussion and Limitations

E^{2}MEND is designed for bounded offline evolution, not unrestricted self-modification. Evidence comes from fixed slices and adaptively selected candidates, so search-stage significance is exploratory without untouched confirmation data or multiplicity correction Xu et al. (2026); Ye et al. (2026).

Layer-A and UCB are routing and ordering mechanisms, not claims of definitive causal attribution; rollback is checkpoint-level, not action-level repair, and deployment without online labels remains unvalidated. Equal-budget search, graded diagnostic observability (DCR), a localizability score for diagnostic traces Liu et al. (2026b), and visible protected-group deltas remain future work. Overall, the results support a bounded, attributable strategy-search claim on fixed slices, not a production deployment guarantee.

9 Conclusion

E^{2}MEND turns memory improvement into bounded, checkpoint-reversible strategy search. The reported probes separate evidence availability from remaining answer-path limitations; fixed-slice experiments associate gains with support selection and typed-state candidates, narrowing the memory-attribution gap on those slices. Future work should add held-out confirmation, equal-budget search, graded DCR Liu et al. (2026b), visible protected-group deltas, complete release artifacts, and action-level rollback evaluation Yu et al. (2026).

Reproducibility statement

Protocol settings, fixed-slice scopes, model roles, and pairing units are summarized in Appendix Table 4; verification thresholds and algorithms appear in Appendix B. Source code and protocol documentation are provided as supplementary material. As recorded in the reproducibility snapshot, the executable code covers only a subset of the reported components, and fixed-slice selection manifests remain unreported.

Ethics statement

This work evaluates memory systems on public benchmark slices and does not deploy models or collect user data. Memory systems can nevertheless process personal or sensitive information in deployment, so privacy, consent, and platform-safety controls remain necessary.

AI use statement

During manuscript preparation, generative AI tools assisted with language editing, structural feedback, formatting, and drafting portions of the appendix, including the worked lifecycle-diagnosis example and related explanatory text. The authors reviewed and verified all AI-assisted material and take responsibility for the paper's claims, evidence, and final content.

REFERENCES

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. URL https://arxiv.org/abs/2504.19413.

Eric Hanchen Jiang, Zhi Zhang, Yuchen Wu, Levina Li, Dong Liu, Xiao Liang, Rui Sun, Yubei Li, Edward Sun, Haozheng Luo, Zhaolu Kang, Aylin Caliskan, Kai-Wei Chang, and Ying Nian Wu. Memory as a controlled process: Learned adaptive memory management for LLM agents. *arXiv preprint arXiv:2607.13591*, 2026. URL https://arxiv.org/abs/2607.13591.

Qingshan Liu, Guoqing Wang, Wen Wu, Jingqi Huang, Xinqi Tao, Dejia Song, Jie Zhou, and Liang He. MemPro: Agentic memory systems as evolvable programs. *arXiv preprint arXiv:2606.00619*, 2026a. URL https://arxiv.org/abs/2606.00619.

Xule Liu, Yijun Liu, Chao Li, and Kun Shao. D^2ACCI: A dual-loop diagnostic protocol for evidence-preserving agent memory. *arXiv preprint arXiv:2608.17756*, 2026b. URL https://arxiv.org/abs/2608.17756.

Xule Liu, Hanlin Teng, Chao Li, Yanan Ni, Shuo Lu, Audrey Wang, Yijun Liu, Yunfei Wang, Xiaofeng Li, Xian Yi, Yuanfa Li, Kang Zhao, Jian Liang, Yuxuan Chen, Jinyuan Chen, Heng Qu, Kun Shao, and Jian Luan. Mi-Memory: A lifecycle memory framework for personal AI. *arXiv preprint arXiv:2607.18975*, 2026c. URL https://arxiv.org/abs/2607.18975.

Meixiu Long, Duolin Sun, Dan Yang, Yihan Jiao, Lei Liu, Jiahai Wang, Binbin Hu, Yue Shen, Jie Feng, Zhehao Tan, Junjie Wang, Lianzhen Zhong, Jian Wang, Peng Wei, and Jinjie Gu. GroupRank: A groupwise paradigm for effective and efficient passage reranking with LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25165–25186, San Diego, California, United States, July 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1260. URL https://aclanthology.org/2026.findings-acl.1260/.

Xianxuan Long, Zhikai Chen, Shenglai Zeng, Shouren Wang, Kai Guo, and Jiliang Tang. MemTrace: Probing what final accuracy misses in long-term memory. *arXiv preprint arXiv:2606.17328*, 2026b. URL https://arxiv.org/abs/2606.17328.

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 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2402.17753.

Natchanon Pollertlam and Witchayut Kornsuwannawit. Total recall at what cost? benchmarking the serving cost of agentic memory systems. *arXiv preprint arXiv:2608.11879*, 2026. URL https://arxiv.org/abs/2608.11879.

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.

Linfang Shang, Ming Xu, Yiding Sun, Tianle Xia, Lingxiang Hu, Lan Xu, and Ning Zheng. When self-evolution backfires: Pre-commit gating against skill contamination in LLM agents. *arXiv preprint arXiv:2608.05810*, 2026. URL https://arxiv.org/abs/2608.05810.

Yongye Su, Wujiang Xu, Chaoji Zuo, and Elisa Bertino. ChronoMem: Version control and semantic rollback for large language model agent memory. *arXiv preprint arXiv:2607.27773*, 2026. URL https://arxiv.org/abs/2607.27773.

Supermemory. Supermemory is the new state-of-the-art in agent memory. https://supermemory.ai/research/longmembench/, 2026. Accessed 2026-07-27.

supermemoryai. supermemory github repository. https://github.com/supermemoryai/supermemory, 2026. Accessed 2026-07-29.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://arxiv.org/abs/2506.21605.

Joshua Tobkin and David Yang. Supra cognitive modes: A routed architecture for agent memory. *arXiv preprint arXiv:2607.19096*, 2026. URL https://arxiv.org/abs/2607.19096.

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*, 2025. URL https://arxiv.org/abs/2410.10813.

Cheng Xu, Nan Yan, Liming Chen, and M-Tahar Kechadi. Phantom gains: Auditing self-improvement against a measured null. *arXiv preprint arXiv:2608.20290*, 2026. URL https://arxiv.org/abs/2608.20290.

Yi Yang, Zhennan Chen, Yihong Zhuang, Tiehan Fan, Yinan Chen, Jian Li, Jian Yang, and Ying Tai. RoMeRL: Balancing feedback coverage and the memory-reward trap in self-evolving agent memory via reduced-order utility states. *arXiv preprint arXiv:2608.02508*, 2026. URL https://arxiv.org/abs/2608.02508.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://aclanthology.org/D18-1259/.

Qinyuan Ye, Yu Li, Yada Pruksachatkun, Jiaxin Zhang, and Chien-Sheng Wu. On the fragility of self-improving agents: Variance, task order, and underspecification. *arXiv preprint arXiv:2608.18066*, 2026. URL https://arxiv.org/abs/2608.18066.

Caili Yu, Yiqi Wang, Jiaqi Zhang, Yiqun Duan, Mingkai Zheng, Zhangkai Wu, Kaize Shi, and Taotao Cai. From faulty memories to corrected actions: Dependency-guided rollback repair for memory-augmented agents. *arXiv preprint arXiv:2608.10502*, 2026. URL https://arxiv.org/abs/2608.10502.

Boqin Yuan, Yue Su, and Kun Yao. Diagnosing retrieval vs. utilization bottlenecks in LLM agent memory. *arXiv preprint arXiv:2603.02473*, 2026. URL https://arxiv.org/abs/2603.02473.

Mingyu Zhang and Ying Ma. HKVM-RAG: Key-value-separated hypergraph evidence organization for multi-hop RAG. *arXiv preprint arXiv:2606.07218*, 2026. URL https://arxiv.org/abs/2606.07218.

Lei Zheng, Weinan Song, Daili Li, and Yanming Yang. To know is to construct: Schema-constrained generation for agent memory. *arXiv preprint arXiv:2604.20117*, 2026. URL https://arxiv.org/abs/2604.20117.

APPENDIX

A ALGORITHMIC PROCEDURES

The three algorithms cover diagnosis, candidate verification, and round orchestration with rollback. Candidate search is described in the main text; separating search from verification keeps adoption traceable to a candidate, its paired decision, and the round record.

Algorithm 1 Layer-A diagnosis.

Require: wrong-answer sample $x = (q, a^*, E)$; provider P; memory store $\mathcal{F}$
Ensure: routing label λ_i
Given fixed stored and retrieved records, identifier coverage and per-evidence ranks are deterministic when E is complete; only Phase 3 uses an LLM.
if E is incomplete or its completeness cannot be established **then**
  return `uncertain`
end if
Phase 1: evidence-to-memory coverage.
$(D_E, W_E) \leftarrow$ dialogue ids and ingestion-window ids implied by E
$\mathcal{F}_E \leftarrow \textsc{EvidenceFacts}(D_E, W_E, \mathcal{F})$ *source-id overlap establishes provenance only*
if $\mathcal{F}_E = \emptyset$ **then**
  return `ingestion_gap`
end if
Phase 2: retrieval coverage.
$(I, X) \leftarrow \textsc{Retrieve}(P, q)$ *retrieved ids and texts*
$j \leftarrow \textsc{Rank}(\mathcal{F}_E, I, X)$ *best 1-based rank/index into I; -1 if absent*
if $j < 0$ **then**
  return `retrieval_gap`
end if
Phase 3: source/fact faithfulness judgment.
$u \leftarrow \textsc{SourceWindow}(I_j)$ *original source window*
$\kappa \leftarrow \textsc{LLMClassify}(q, a^*, u, X_j)$ *Full_Coverage / Source_Only / Uncertain*
if $\kappa =$ `Full_Coverage` **then**
  return `generation_error`
else if $\kappa =$ `Source_Only` **then**
  return `ingestion_gap`
end if
return `uncertain`

B DIAGNOSTIC AND VERIFICATION DETAILS

Layer-A implementation note. Layer-A follows the three-phase stage-resolved rule in Algorithm 1. It produces low-level evidence diagnoses: empty evidence-to-memory coverage is *ingestion gap*, a missing retrieval rank is *retrieval gap*, and the source/fact faithfulness judgment after retrieval yields *generation error* (`Full_Coverage`), a second *ingestion gap* signal (`Source_Only`), or *uncertain*. Before candidate generation, ROUTEDIAGNOSIS applies the task predicate Q_i: tasks requiring comparison, aggregation, or temporal update are routed from *generation error* to *state-construction gap*, while other tasks are routed to *answer-model gap*. Coverage and rank checks are deterministic only with complete evidence ids and fixed stored/retrieved records; otherwise Layer-A should be treated as an audit signal, not a deterministic causal attribution.

B.1 WORKED LIFECYCLE DIAGNOSIS

Figure 3 gives a concrete example of the routing rule above. The relevant dialogue turns are retrieved, but the raw-concatenation baseline leaves the lifecycle relation implicit and does not normalize the relative completion time. The diagnosis therefore selects a query-specific lifecycle state, after which the structured-evidence strategy resolves the relation between the observed start and completion events.

Adaptive-search interpretation. The one-sided exact McNemar test is calibrated for a prespecified candidate on a fixed paired slice. Because the search procedure evaluates multiple candidates and retains promising outcomes, unadjusted search-stage p-values do not control family-wise error. We therefore treat them as

Algorithm 2 Staged paired verification and adoption.

Require: current strategy s_t, candidate c, target set $\mathcal{T}_t$, stable set $\mathcal{S}_t$, full slice $\mathcal{B}$, protected groups $\mathcal{K}$, thresholds $\tau_{\text{full}}, \tau_{\text{pr}}, \tau_{\text{pe}}, \tau_{\text{stable}}, \alpha$, and protected-group tolerances $\{\epsilon_k\}$
Ensure: StrictAccept, PracticalAccept, Pending, or Reject

if a hard gate fails or the critic blocks c **then**
  return Reject and record negative evidence
end if
Evaluate c on $\mathcal{T}_t \cup \mathcal{S}_t$ and align outputs by the configured pairing key
if keys mismatch or Eq. 4 fails **then**
  return Reject and update the direction history
end if
Run the full paired evaluation; compute Eq. 5 and every $\Delta_k(c)$
if any protected group or stable-replay gate fails **then**
  return Reject and persist the regressed comparison keys
else if $\Delta(c) \geq \tau_{\text{full}}$ and $p_{\text{McN}}(c) < \alpha$ **then**
  return StrictAccept
else if $\Delta(c) \geq \tau_{\text{pr}}$ **then**
  return PracticalAccept with descriptive evidence and rollback protection
else if $\tau_{\text{pe}} \leq \Delta(c) < \tau_{\text{pr}}$ **then**
  return Pending champion for later confirmation
end if
return Reject

Algorithm 3 Detailed E^{2}MEND round and history update.

Require: strategy s_t, schema Γ, slice $\mathcal{B}$, history $\mathcal{H}_t$, task predicate Q, provider P, rollback threshold τ_{rollback}
Ensure: served strategy s_{t+1}, candidate decision $decision_t$, and history $\mathcal{H}_{t+1}$

Evaluate s_t on $\mathcal{B}$ and align outputs by the configured pairing key.
Let $\mathcal{W}_t \subseteq \mathcal{B}$ be the wrong-answer samples under s_t; set $Z_t \leftarrow \{\text{LayerA}(x)\}_{x \in \mathcal{W}_t}$ (Algorithm 1).
$\tilde{Z}_t \leftarrow \text{RouteDiagnosis}(Z_t, Q)$; $g_t \leftarrow \text{Digest}(\tilde{Z}_t)$.
Construct the target set from $\tilde{Z}_t$, the stable-correct set from the current aligned evaluation, and protected sets from the fixed protocol.
From digest g_t and history $\mathcal{H}_t$, materialize eligible paths (Eq. 2), rank them by Eq. 3, and form candidate list $\mathcal{C}_t$.
Initialize $\hat{s}_{t+1} \leftarrow s_t$, $decision_t \leftarrow$ Reject, and $c^\star \leftarrow \emptyset$.
for $j = 1, \ldots, \min(3, |\mathcal{C}_t|)$ **do**
  Evaluate candidate c_j with the targeted screen in Eq. 4.
  if c_j passes the screen **then**
    $c^\star \leftarrow c_j$; **break**
  else
    update the direction history with the screened failure
  end if
end for
if $c^\star \neq \emptyset$ **then**
  Run full paired verification for $c^\star$ with Algorithm 2; set $decision_t$ and update $\hat{s}_{t+1}$ from its decision.
end if
if $decision_t \in \{\text{StrictAccept}, \text{PracticalAccept}\}$ **then**
  persist decision report, diff, paired metrics, and candidate; set $s_{t+1} \leftarrow \hat{s}_{t+1}$.
else if $decision_t =$ Pending **then**
  persist the pending champion and confirmation evidence; keep $s_{t+1} \leftarrow s_t$.
else
  persist the rejected-round summary; keep $s_{t+1} \leftarrow s_t$.
end if
if serving accuracy falls more than τ_{rollback} below the best accepted checkpoint **then**
  restore best-ever strategy s^* and persist the rollback patch.
end if
return $s_{t+1}, \mathcal{H}_{t+1}$

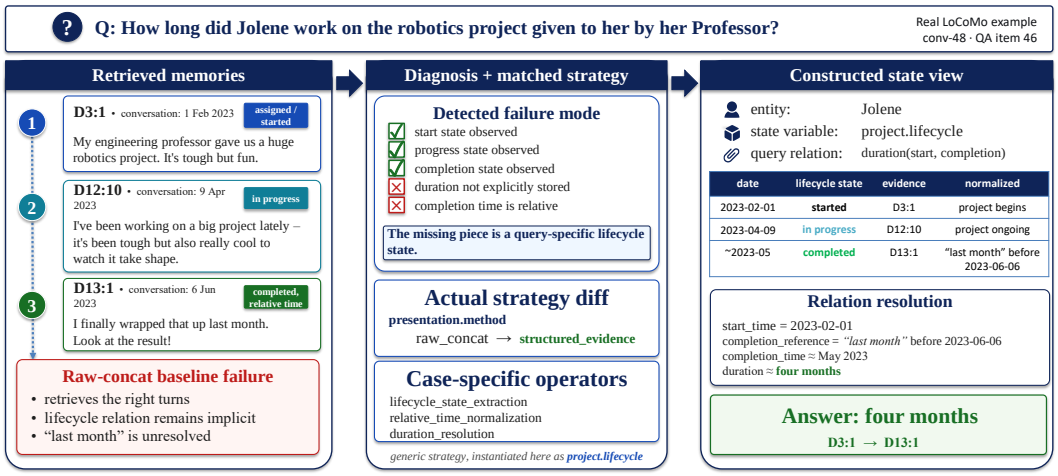


Figure 3: Worked example of query-specific lifecycle diagnosis and typed-state construction. The retrieved turns are preserved, the missing lifecycle relation is diagnosed, and a structured state view resolves the relative completion time into the answer "four months."

exploratory unless the candidate is evaluated on untouched confirmation data or under a stated multiplicity correction.

Decision thresholds. After the stable and protected-group gates pass, the implementation defaults are: strict accept if $\Delta(c) \geq 0.03$ and $p_{\text{McN}}(c) < 0.05$; practical accept if $\Delta(c) \geq 0.02$; pending if $0.01 \leq \Delta(c) < 0.02$; and reject below 0.01. In the reported setup, $\tau_{\text{full}} = 0.03$, $\tau_{\text{pr}} = 0.02$, $\tau_{\text{pe}} = 0.01$, $\tau_{\text{stable}} = 1$, $\tau_{\text{rollback}} = 0.03$, and $\alpha = 0.05$. Practical and pending outcomes are descriptive. The stable-replay tolerance $\tau_{\text{stable}} = 1$ allows one stable-correct regression and is an implementation default, not a distributional guarantee.

B.2 Evolver Prompt

The evolver is a bounded-diff generator, not a free-form code editor. Its prompt combines strategy and schema context, legal paths, Layer-A evidence, history, and mutation budget, and requires an auditable JSON diff with impact, rationale, and risk. The listing below is the Evolver prompt/output schema: braces denote per-round placeholders and the JSON object illustrates the required format, not a verbatim model response from a particular run. Representative real candidate values are reported separately in the development trace when available.

```
You are a memory system optimizer. Generate strategy modifications to fix
     specific failures.

## Current Strategy
```yaml
{current_strategy}
```

Evolution Landscape
- Themes: {themes}
- Focus Areas: {focus_areas}
- Reputation Scores: {reputation}

Active Hypothesis (UCB-selected, may be empty)
{hypothesis_context}

Schema Constraints (legal values only)
```yaml
{schema_snippets}
```

Dimension Routing (MANDATORY)
```

```
Each bottleneck maps to specific strategy paths. You MUST only suggest
    changes in the allowed paths for the dominant failure category:

{allowed_paths}

## Previously Attempted (DO NOT repeat these exact changes)
{rejection_history}

## Error Samples with Layer-A Data
{error_samples}

## {budget_hint}

## Your Task
Generate {K} candidate strategy modifications. CRITICAL RULES:
1. **Path MUST be in the allowed list above** --- do NOT suggest changes
    outside these paths. Distribute candidates according to the Mutation
    Budget above --- each category MUST get at least its allocated slots.
2. Use **Layer-A data** (extraction_coverage, retrieval_rank) to pick the
     RIGHT magnitude:
   - If retrieval_rank is just outside top_k (e.g., rank=35, top_k=30),
    small increase (+5 to +10)
   - If retrieval_rank is far (e.g., rank=80, top_k=30), consider large
    increase or method change
   - If extraction_coverage=0.0, extraction_prompt needs modification,
    NOT a numeric tweak
3. **Specify exact YAML path** (must be in allowed paths)
4. **Old value -> New value** (must respect schema constraints)
   - For `presentation.method`: the ONLY legal values are `raw_concat`
    and `structured_evidence`. Do NOT invent variants such as `
    structured_evidence_cards`, `evidence_table`, or ranking-specific
    names; use `structured_evidence` instead.
   - For `ingestion.extraction_prompt`: provide ONLY the new rules to
    APPEND (not the full prompt). Write them as numbered rule lines. The
    system will insert them into the existing prompt automatically. NEVER
     replace the entire prompt.
5. **Predicted impact** (which error categories does this fix?)
6. **Rationale** (1-2 sentences, cite specific Layer-A data)
7. **Regression risk** (low/medium/high)

Output JSON:
{{"candidates": [{{"path": "retrieval.top_k", "from": 30, "to": 40, "
    impact": "Fixes retrieval_gap when a gold fact falls just outside
    top_k", "rationale": "Layer-A shows gold facts at rank 35, outside
    current top_k=30", "risk": "low"}}]}}
```

The prompt is deliberately structured as an auditable interface rather than an open-ended instruction to rewrite the memory system. The current strategy and schema snippets define the legal state space; the dimension-routing block narrows the candidate paths using the diagnosed bottleneck; and the rejection history prevents the evolver from repeatedly proposing edits that have already failed verification. Layer-A evidence supplies the magnitude signal, so a candidate can distinguish a small retrieval adjustment from a representation change or an extraction-rule amendment.

The required JSON fields also separate proposal from adoption. A candidate must state the exact path and value transition, the intended error category, the evidence-based rationale, and its regression risk, but it does not become active merely because it is syntactically valid. Hard gates, targeted screening, full paired verification, and the stable/protected-group checks in Algorithm 2 remain the authority for accepting, retaining, or rejecting the proposed diff.

C Protocol and Reproducibility

The following table summarizes the fixed model roles, pairing unit, and slice scope used across experiments.

Table 4: Reproducibility snapshot.

Protocol item	Documented setting
Ingestion / intent parsing	`Qwen3.5-9B`
Evolution	`GPT-5.5`
Evaluation / judge path	`GPT-4o`
Embedding	`text-embedding-3-small`
Pairing unit	Question key for single-run tests, or run–question key for descriptive pooled reruns, aligned before comparison
Claim scope	Diagnostic, strategy, and reference-only claims are kept separate
Configuration / rerun reporting	HotpotQA GroupRank reports three configuration settings; MemBench safe-combo reports three paired runs and a labeled pooled aggregate
Evidence visibility	Oracle-evidence probes use complete gold evidence; Layer-A uses annotations only when available
Protected groups	Question-type or task-family partitions are reused across baseline and candidate runs for regression checks
Slice scope	Fixed slices are used for LoCoMo (1540), HotpotQA distractor (300), LongMemEval-S QA (500 source split / 470 evaluated QA questions), LongMemEval-M retrieval (500 source split / 470 evaluated retrieval instances), and MemBench (200)
Slice provenance / labels	HotpotQA's 300-question selection manifest/order is not reported; GroupRank numeric labels are retained implementation labels, not formal hyperparameter definitions

Fixed-slice sizes are stated, but dataset versions/orders and selection manifests are not included. Executable code is provided in the supplementary material, though it covers only a subset of the reported components; exact hardware, random seeds, complete decoding/preprocessing settings, and model checkpoint identifiers are not reported. The snapshot therefore documents the reported protocol boundary, not a full reproduction package.

C.1 Evaluation Budget and Model Allocation

Each self-evolution round uses one full-slice evaluation over the LoCoMo slice, up to three subset screens, and at most one additional full paired verification for the first candidate that passes targeted screening. Baseline results are reused across rounds, so unchanged baseline rows do not incur repeat full-slice recomputation. The evolution loop generates roughly $K \approx 9$ candidates per round but evaluates at most three of them.

Across the reported runs, one self-evolution round consumes about 2,500,380 tokens on average, with roughly 2.405M prompt tokens and 0.095M completion tokens. These figures are a judge-side lower bound, since diagnosis and planning (GPT-5.5) and ingestion (Qwen3.5-9B) do not persist usage fields.

In the final LoCoMo reproduction chain, remaining-wrong candidate-bank generation is run on 97 questions and yields 64 supported candidates. The subsequent supported-candidate adoption stage is run on 53 questions and accepts 18 candidates. These counts describe generated or adopted candidate edits in the reproduction chain; the headline $+16/-0$ counts question-key-level baseline-wrong/candidate-correct flips for the final reported paired checkpoint, so the two are not one-to-one.

D Supplementary Benchmark Details

D.1 LongMemEval Retrieval Reference

The following table provides the retrieval-side reference comparison summarized in the main text.

In the table, $K = V$ denotes verbatim dialogue turns, $K = \text{fact}$ denotes extracted facts, and $K = V + \text{fact}$ stores both representations; "Round" and "Session" indicate the granularity of the retrieval unit.

Consistent with the official LongMemEval retrieval protocol Wu et al. (2025), the retrieval metrics exclude the 30 abstention instances because those questions do not have a ground-truth answer location. The R@5 and R@10 values in Table 5 are therefore computed over the remaining 470 non-abstention instances, while the slice label retains the 500-question source split.

These retrieval results are consistent with the lexical, answer-surface nature of many LongMemEval QA questions and with the longer contiguous evidence preserved by the session-chunk representation relative to the

Table 5: Retrieval reference on the 500-question LongMemEval-M split; each question has a roughly 500-session history, and R@5/R@10 use the 470 non-abstention instances.

Method / key design	Granularity	R@5	R@10
$K = V$	Round	0.582	0.692
$K = \text{fact}$	Round	0.530	0.654
$K = V + \text{fact}$	Round	0.644	0.784
$K = V$	Session	0.706	0.783
$K = V + \text{fact}$	Session	0.732	0.862
E^2MEND: embed_chunk	Session chunks	0.850	0.910

round-level representations in Table 5. In the reported examples, the question text often overlaps directly with the answer-bearing session chunk: for example, "What degree did I graduate with?" aligns with the chunk containing "I graduated with a degree in Business Administration," and "How long is my daily commute to work?" aligns with "45 minutes each way."

D.2 Retrieval-Bound Transfer on HotpotQA

The main text summarizes the retrieval gain; this table keeps the configuration details. On the first 300 HotpotQA distractor questions, the both@2 metric (both gold paragraphs appear among the top two retrieved groups) measures support retrieval without answer extraction. The hybrid baseline reaches 0.320 and a cross-encoder rerank 0.443, while GroupRank Long et al. (2026a) reaches 0.637–0.657 across three configuration settings (0.5, 0.7, and 1.0), a gain of 0.317–0.337. Each row reports a setting-specific range; the three rows are not independent reruns.

The manuscript does not report the dataset version/order or an explicit selection manifest for this 300-question slice; the numeric labels 0.5, 0.7, and 1.0 are retained from the implementation and are not presented as formal GroupRank hyperparameter definitions.

Table 6: HotpotQA distractor retrieval (first 300 questions).

Method	both@2	Δ vs. base
Hybrid baseline	0.320	–
Cross-encoder rerank	0.443	+0.123
GroupRank, setting 0.5	0.643–0.650	+0.323–0.330
GroupRank, setting 0.7	0.643–0.657	+0.323–0.337
GroupRank, setting 1.0	0.637–0.647	+0.317–0.327

D.3 MemBench Paired Results

The safe-combo experiment uses the same 200-question slice in three paired runs, yielding $0.785 \rightarrow 0.875$, $0.755 \rightarrow 0.845$, and $0.770 \rightarrow 0.870$. Pooling the run–question comparisons gives $462/600 = 0.770$ for the source-window baseline and $518/600 = 0.863$ for SlotTable, with $+62/-6$ run–question flips. Here 600 denotes run–question pairs, not 600 independent questions; because the three runs are repeated measurements over the same slice, the pooled aggregate is descriptive, not an estimate of population performance.

Because the slice is not explicitly restricted to state-construction questions, the paired gain supports the typed-state candidate on that fixed slice without isolating the state-construction pathway. SlotTable reads structured fields already present in MemBench events, normalizes units and dates, matches rows, and performs the requested comparison, count, or conditional lookup. The 0.930 staged checkpoint includes additional strategy components and is not part of the safe-combo aggregate.

The MemBench reference rows use the factual-memory Observation setting reported by Tan et al. Tan et al. (2025), with their own judge path and protocol. They are context, not matched comparators; the paired claim is only $0.770 \rightarrow 0.863$ against our source-window baseline.

D.4 LongMemEval-S QA Comparison

The main text keeps only the takeaway; the detailed protocol-specific comparison stays here. On the 500-question LongMemEval-S QA split, E^2MEND reaches 0.950 under a source-window strategy; following the official protocol, the displayed QA metrics use the 470 non-abstention questions and exclude the 30 abstention instances. LongMemEval-S uses approximately 115k-token histories, whereas the retrieval reference in Table 5 uses LongMemEval-M, whose per-question histories contain roughly 500 sessions; the two are distinct splits.

Table 7: MemBench results and protocol-specific reference points; reference rows are context, while the paired aggregate is the reported claim.

Method	Acc.	Source / status
GenerativeAgent	0.779	MemBench ref.
MemoryBank	0.721	MemBench ref.
MemGPT	0.789	MemBench ref.
SCMemory	0.529	MemBench ref.
RetrievalMemory	0.883	MemBench ref.
Source-window baseline (3-run pooled)	0.770	Ours, paired aggregate
E^2MEND SlotTable safe-combo (3-run pooled)	**0.863**	Ours, paired aggregate
E^2MEND staged checkpoint	0.930	Ours, broader system

Full-context and Zep are reference points on the same LongMemEval-S slice but are not matched to our backbone and judge path, while the Supermemory rows come from a public report with different backbones (GPT-4o, GPT-5, Gemini) and protocol, so they are reference-only context, not a matched comparison. In Table 8, SSU denotes single-session-user, SSA single-session-assistant, SSP single-session-preference, KU knowledge update, TR temporal reasoning, and MS multi-session. The Overall column uses the same 470 non-abstention questions; the table does not report per-category question counts, and the per-type values are reference-only and should not be interpreted as stable population estimates.

Table 8: LongMemEval-S accuracy by question type (reference-only comparison).

System	SSU	SSA	SSP	KU	TR	MS	Overall
Full-context	0.814	0.946	0.200	0.782	0.451	0.443	0.602
Zep	0.929	0.804	0.567	0.833	0.624	0.579	0.712
Our E^2MEND	1.000	1.000	0.733	0.949	0.962	0.940	0.950
Supermemory public report and repository (2026) Supermemory (2026); supermemoryai (2026)							
Supermemory (GPT-4o)	0.970	1.000	0.900	0.990	0.910	0.930	0.950
Supermemory (GPT-5)	0.971	1.000	0.767	0.872	0.812	0.752	0.846
Supermemory (Gemini)	0.986	0.982	0.700	0.897	0.820	0.767	0.852

D.5 Cumulative Pipeline Ablation

This table reports separately re-evolved ablation conditions. Each condition starts from the same cached initial strategy, so the comparison controls for the initial state while preserving the actual evolution process. The removals are cumulative, not factorial, because the mechanisms form a tightly coupled pipeline. In particular, UCB is not an independently decomposable component—it searches only over the eligibility mask that Layer-A defines and updates only from adoption outcomes that paired verification produces—so a UCB-specific ablation is not well-posed. The well-defined interventions are diagnosis, digest compression, and screening, which the cumulative removal measures.

Table 9: Cumulative pipeline ablation on the fixed LoCoMo slice.

Variant	Accuracy	Δ vs. Full	Interpretation
Full system	**0.947**	–	Cached artifact reference
w/o Layer-A	0.901	-0.046	Remove routed diagnosis
then w/o Digest	0.881	-0.066	Further remove digest compression
then w/o Screening	0.854	-0.093	Further remove screening checks

Because the mechanisms form a tightly coupled pipeline, cumulative removal measures pipeline sensitivity, not isolated contributions; the difference between consecutive rows describes the additional sensitivity when that component is removed on top of the prior removals, not an independent causal effect. The table is descriptive evidence for the three well-defined interventions (diagnosis, digest compression, and screening), not an estimate of a UCB-specific effect.

E DEVELOPMENT AND SEARCH EVIDENCE

E.1 REPRESENTATIVE STAGED CANDIDATE OUTCOMES

The following table reports a representative subset of the staged E^{2}MEND candidate outcomes summarized during development, not an exhaustive census of generated candidates. The IDs and intermediate checkpoint values are retained development labels and summaries; the current artifact independently retains the baseline and final paired checkpoint, but not every intermediate candidate artifact. Query examples are real LoCoMo questions used to make each direction concrete; unless noted as validation evidence, they are illustrative and do not imply that the candidate repaired that question.

The final column records the verification evidence available in the reported summary. Its flip counts are development-summary numbers and are not uniformly step- or baseline-relative, so they should not be compared directly with the displayed accuracy. In the *Dec.* column, `base` denotes the baseline row, `no-eff.` denotes no effect, `acc.` denotes an accepted development summary, `pend.` denotes pending-tier, and `rej.` denotes rejected.

Table 10: Staged candidate outcomes on LoCoMo (development summaries; not exhaustive).

ID(s)	Strategy dimension	Candidate change	Query / example	Dec.	Acc.	Step Δ	Verification evidence
000	Initial memory strategy	Baseline before staged E^{2}MEND evolution	–	base	0.756	–	1164/1540
005	Semantic multi-hop reranking	Enable graph-based reranking for semantic multi-hop retrieval	Q: In what ways is Caroline participating in the LGBTQ community?	no-eff.	–	≈0.000	Reported target set showed no gain
027	Evidence-grounded context	Attach local dialogue evidence to answer context	Q: How does Gina describe the studio that Jon has opened?; preserve "amazing"	acc.	0.818	+0.062	Reported summary: +96 vs. baseline; early reverse flips unavailable
034	Scoped evidence gate	Use candidate only on evidence-supported cases	Q: How does Gina describe the studio that Jon has opened?; apply only to conv-30_q72	acc.	0.845	+0.027	Reported summary: +42 step; limited regression record
036	Raw-dialogue fallback	Expand raw-dialogue fallback search	Q: What state did Joanna visit in summer 2021?; fall back when support is weak	rej.	–	unstable	Reported screen summary: +10 / -7
047	Typed state abstraction	Convert evidence into temporal, location, plan, preference, slot, and ledger state	Q: What state did Joanna visit in summer 2021?; organize time/location fields	acc.	0.871	+0.026	Reported summary: +177 vs. baseline; early reverse flips unavailable
060	Verified candidate bank	Adopt only candidates passing support and non-regression checks	Q: What injury did Evan suffer from in August 2023?; accept after both checks	acc.	0.915	+0.044	Reported summary: +245 vs. baseline; early reverse flips unavailable
064	Support clustering	Use deterministic support clusters instead of selector choice	Q: What outdoor activities has John done with his colleagues?; cluster support snippets first	acc.	0.937	+0.022	Reported summary: +34 step / 0
067	Evidence-supported consolidation	Consolidate explicitly supported candidates and suppress low-confidence candidate classes	Q: What design is featured on Calvin's guitar?; merge supported, drop weak branches	pend.	**0.947**	+0.010	Final paired checkpoint: +16 step / -0

For row 067, the final full-slice paired effect is $\Delta = 16/1540 \approx 0.010$; under the reported implementation defaults in Appendix B, this is pending-tier, not strict or practical acceptance, so the paper reports it descriptively.

E.2 BUNDLE EVIDENCE SUMMARY

The following table groups bundle search outcomes by decision signal, separating positive, tied/no-op/weak, and negative candidates. It complements the staged candidate trace (Appendix Table 10) by showing that compound edits are judged under the same paired, non-regression adoption rule as single-field moves: a bundle is adopted only when it beats its single-field comparator and the gain repeats. These are development/search

signals; a displayed repeat belongs to that ablation and is not the three-run headline aggregate. No-op/weak rows are archived, and negative rows are rejected outright.

In each row the *Result* column reports the comparator → full-bundle outcome (left value = comparator, right value = full bundle), so its baselines are ablation-specific and deliberately differ from the headline baselines in Tables 6 and 7. In the Comparator column, `single-only` denotes the single-field parent without the bundle, and `trunk-only` denotes the strategy trunk before the named intervention. For readability, the table names each intervention and comparator by its functional role instead of exposing implementation-specific internal node names.

Table 11: Bundle search outcomes by decision signal.

Signal	Benchmark	Bundle / node	Comparator	Result	Takeaway
Positive	HotpotQA distractor, $n = 300$	balanced GroupRank	single-only	both@2 0.197→0.577 repeat 0.563	Gain repeats across runs
	MemBench development, $n = 200$	SlotTable safe-combo	single-only	acc. 0.360→0.705 repeat 0.725	Development bundle signal; not the headline aggregate
	MemBench comp., agg., cond.; $n = 120$	balanced intent planning	trunk-only	acc. 0.150→0.233	Positive but slice-specific
No-op / weak	HotpotQA catalog	entity-heavy (e1/e2)	trunk-only	tied / weak positive	Needs stronger paired evidence
	HotpotQA catalog	query matching (q1/q2)	trunk-only	no-op	Plausible direction is ineffective
	MemBench catalog	structured evidence (s1); recall-heavy intent planning	trunk-only	weak positive / noise-level	Archive noise-level gains
Negative	MemBench catalog	aggregation recall (a1)	trunk-only	negative	Reject harmful directions
	MemBench catalog	comparison evidence (c1/c2)	trunk-only	negative	Not universally beneficial

F Extended Related Work

This section preserves the detailed literature comparison summarized in Section 7.

Memory benchmarks and evaluation. Long-term-memory benchmarks test QA over extended histories Maharana et al. (2024); Wu et al. (2025), and HotpotQA isolates multi-hop retrieval Yang et al. (2018). Recent audits show that aggregate scores rarely identify whether a failure is in ingestion, retrieval, state construction, or the answer model Long et al. (2026b), and that accuracy and serving cost must be measured together Pollertlam & Kornsuwannawit (2026).

Memory architectures and adaptive memory. Storage systems address scalability, personalization, and temporal representation Chhikara et al. (2025); Rasmussen et al. (2025). Adaptive systems evolve memory content, schemas, or operational policies Liu et al. (2026a); Jiang et al. (2026); Zheng et al. (2026); Supra Cognitive Modes separates routing from storage but routes by query mode over a largely fixed policy Tobkin & Yang (2026). E^{2}MEND instead locks the runtime and makes the diagnosis-routed strategy artifact the only mutable object.

Diagnostic and regression-aware iteration. D^{2}ACCI introduces dual-loop diagnosis with protected-slice monitoring and graded observability Liu et al. (2026b), and controlled probes find that retrieval method dominates write strategy on LoCoMo Yuan et al. (2026). Skill-gating work further shows that post-hoc rollback may not undo contamination inherited from admitted skills, motivating pre-commit gates Shang et al. (2026).

Governance, stability, and relation to prior lifecycle design. A prior lifecycle-memory framework introduces schema-constrained evolution with gated adoption and checkpoint rollback Liu et al. (2026c); the present work develops that component into a standalone treatment with bottleneck diagnosis and paired non-regression verification.

In sum, the adjacent literatures converge on traceable iteration, paired evidence, and regression awareness. Most systems evolve the memory store, the end-to-end system, or a single retrieval stage; E^{2}MEND separates the diagnostic routing signal from the mutable strategy artifact and keeps answer-model limitations out of memory mutation.

G Extended Limitations

This section preserves the detailed limitation audit summarized in Section 8.

E^2MEND is designed for bounded offline evolution, not unrestricted self-modification. Evidence comes from fixed slices and adaptively selected candidates; without multiplicity correction or untouched confirmation data, search-stage significance is exploratory.

Prior transition-level audits also warn that differencing two noisy estimates can manufacture apparent gains when each statistic lacks a separately measured null Xu et al. (2026); fixed pairing and explicit flip counts reduce but do not eliminate such measurement artifacts. Re-evaluations of memory-based self-improvement further indicate that apparent gains are sensitive to run variance and task order Ye et al. (2026); the fixed slices and paired checks used here reduce but do not eliminate that fragility. Learning-based memory utilities can likewise become misleading when rewards are jointly assigned to co-retrieved memories, a memory-reward trap that bounded pairing-key flips mitigate but do not remove Yang et al. (2026).

Layer-A depends on evidence identifiers and an LLM-assisted faithfulness judgment, and UCB is treated as an ordering mechanism, not a claimed component; an equal-budget comparison against random or greedy search remains future work. Because bundle interactions and the served strategy change across rounds, no stationary-environment regret guarantee is claimed.

The fixed state-construction predicate Q_i is currently defined semantically but not operationally, so its assignment procedure is an additional reproducibility gap for independently reconstructing the diagnosis boundary. Although diagnosis produces per-item traces, the paper does not yet report a graded observability score (DCR) of the kind proposed by D^2ACCI Liu et al. (2026b); such a metric would strengthen the evidence that enriched traces support diagnostic agreement.

Post-hoc rollback is retained as a safeguard, but prior work suggests it may not fully undo contamination inherited by later skills Shang et al. (2026); separate version-control studies further suggest that semantic rollback should be tested counterfactually after subsequent updates Su et al. (2026). Dependency-guided rollback repair further shows that deleting a faulty memory leaves propagated claims and derived memories active, motivating selective provenance-guided replay Yu et al. (2026); the present work implements checkpoint-level rollback, not action-level repair. This reinforces the pre-commit gates and highlights an unreported evaluation for the current artifact.

The protected-group partitions and their per-group paired deltas are not fully reported in this artifact, so the protected non-regression gate is currently a protocol-level check, not an empirically visible slice audit. The reported token figures are judge-side lower bounds, not end-to-end serving costs, and serving-cost benchmarks indicate that accuracy and cost must be considered together Pollertlam & Kornsuwannawit (2026). As recorded in the reproducibility snapshot above, the executable code covers only a subset of the reported components, and exact hardware, random seeds, complete decoding/preprocessing settings, model checkpoint identifiers, dataset versions/orders, and fixed-slice selection manifests are not reported; the current artifact therefore supports protocol inspection, not end-to-end replication.

Deployment with sparse feedback or no online labels remains unvalidated, and quality gates do not replace privacy, consent, or platform safety controls.