MELO: A COGNITIVE SCIENCE–INSPIRED CLOSED-LOOP LONG-TERM MEMORY FRAMEWORK FOR AI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term agent memory must preserve information across sessions and revise it when later evidence changes what should be remembered. Many architectures support writing and retrieval, but do not explicitly connect a recalled memory to the outcome observed after that memory guides a response. Drawing on cognitive accounts of attentional bias, prediction error, and reconsolidation, we propose Melo, a closed-loop framework that makes this return path explicit. It implements selective encoding, outcome-triggered feedback admission, evidence-linked procedural consolidation, and time-dependent retention control. We also introduce CMLM-Assist, an author-constructed synthetic benchmark with 1,920 questions: 384 for development and 1,536 in a held-out test set. Across six baselines and three language-model backbones, Melo obtains the highest F1, BLEU-1, ROUGE-L, and METEOR scores in every evaluated block on both benchmarks. Exact match does not improve uniformly. LoCoMo supplies the independent external comparison, while CMLM-Assist supplies controlled diagnostic evidence. The results support coordinated memory use and revision under the declared configurations, not biological equivalence, equal-budget superiority, or general conflict resolution.

1 INTRODUCTION

AI agents increasingly mediate tasks whose state extends across conversations. A personal assistant may need to preserve a stable preference, replace an obsolete one, resume an unfinished task, or reuse a tool procedure whose outcome was observed earlier (Wang et al., 2024b; Zhang et al., 2025). Long context makes earlier records accessible, but it does not determine which record remains current. A useful long-term memory system must therefore select what to preserve, update the persistent state when later evidence supersedes it, and make the revised state available to subsequent interactions (Hu et al., 2025).

Existing approaches preserve experience for later planning (Park et al., 2023; Shinn et al., 2023), maintain persistent information outside the model context (Zhong et al., 2024; Packer et al., 2023), or coordinate extraction, consolidation, retrieval, and updating as a memory service (Chhikara et al., 2025; Kang et al., 2025). These systems demonstrate that long-term memory can be written and updated. The unresolved issue studied here is narrower: whether evidence observed after a memory-supported response is returned to the persistent state that will support later retrieval.

The problem is therefore not whether a system exposes an update operation. It is whether information used in one interaction remains connected to evidence observed afterward. Suppose a stored Monday deadline supports a reminder, but the user later corrects it to Friday. If the correction is written as an unrelated fact, both dates may remain retrievable. Similar failures occur when repeated successful tool sequences never become reusable procedures. We call a mechanism *closed loop* when (i) a response or action uses persistent memory, (ii) later user, tool, or task evidence is admitted as feedback, (iii) that evidence changes the persistent state, and (iv) the changed state is consulted by a later query. In this paper, *closed loop* names this return pathway. Accuracy, consistency, and retention are outcomes that must be evaluated separately.

The closed-loop return path defined above is guided by three cognitive principles: attentional bias, prediction error, and reconsolidation. *Attentional bias* gives priority to goal-relevant or salient

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.

information when multiple inputs compete, which motivates selective admission and retrieval rather than treating every event equally (Desimone & Duncan, 1995). *Prediction error* is the discrepancy between an expected consequence and the outcome that is observed. In human associative-memory studies, such a discrepancy can govern whether a reactivated trace becomes labile (Sevenster et al., 2013). *Reconsolidation* describes the subsequent updating and restabilization of that trace, during which relevant new information may be incorporated (Nader et al., 2000; Schiller et al., 2010; Lee, 2008). Perception–action accounts likewise describe memory as shaping action while later consequences provide new information (Fuster & Bressler, 2015; Fuster, 2004). Melo adopts this functional pattern: task relevance biases what enters or is recalled, and explicit corrections or outcomes indicate that a used memory may require revision. This is a conceptual correspondence, not a biological model; Melo uses declared sources and semantic tags rather than estimating neural attention or a scalar prediction-error signal.

We propose Melo (Memory Evolution and Layered Orchestration), a pluggable memory component that operationalizes this return path through selective admission, typed routing, evidence-linked procedural consolidation, and deterministic retention control. A user correction or task outcome is stored with provenance and can alter the state used to support later retrieval and generation; repeated successful workflows can be consolidated into a summary that retains the identifiers of its supporting events. Melo does not alter the host model's parameters or reasoning architecture.

We present CMLM-Assist, an assistant-memory benchmark with 1,920 controlled preference, task-update, and workflow questions: 384 for development and 1,536 in a user-disjoint held-out test set. LoCoMo (Maharana et al., 2024) provides public conversational QA but lacks the same predefined goal, preference, and workflow subsets in the annotations used here. Each held-out CMLM-Assist question is linked to supporting memory identifiers, allowing answer quality and evidence access to be checked together. Its programmatic construction provides exact provenance, while the shared templates and author-designed metadata create evaluation dependence that we state explicitly below.

The main contributions of this work are as follows:

1. We propose Melo, a cognitive science–inspired closed-loop memory framework that coordinates attention-biased selective encoding, outcome-triggered feedback admission, evidence-linked procedural consolidation, and deterministic retention control. Recorded outcomes can update the state used for later retrieval.
2. We introduce CMLM-Assist, an author-constructed synthetic diagnostic benchmark for cross-session task continuity, user-profile retention, procedural reuse, and memory-grounded response generation.
3. We evaluate Melo and six baselines on LoCoMo and CMLM-Assist using three backbones. Under the declared system configurations, Melo attains the highest F1, BLEU-1, ROUGE-L, and METEOR scores in every evaluated backbone block. Exact match is mixed, and the main comparison is not retrieval-budget controlled.

2 Related Work

2.1 Long-Term Memory Systems and Closed-Loop Updating

Prior work connects experience to later behavior in two main ways. Generative Agents interleaves observations with reflection and planning, Reflexion records verbal feedback from failed attempts, and Voyager turns successful interaction into reusable skills (Park et al., 2023; Shinn et al., 2023; Wang et al., 2024a). Cross-session memory systems maintain an explicit store: MemGPT manages virtual context, MemoryBank maintains profiles and summaries, Mem0 extracts and consolidates conversational information, MemoryOS coordinates memory services, and A-Mem represents experience as linked notes (Packer et al., 2023; Zhong et al., 2024; Chhikara et al., 2025; Kang et al., 2025; Xu et al., 2025). Together, these systems show that experience can be retained, transformed, and revised. They differ in when post-interaction evidence is admitted and whether that evidence changes the persistent state before a later task.

Recent systems focus on particular control points. TiMem organizes memory temporally, APEX-MEM represents evolving and conflicting facts, D-Mem gates expensive retrieval, and MGRetrieval

reflects over historical structure (Li et al., 2026a; Banerjee et al., 2026; You et al., 2026; Wang & Dong, 2026).

No individual memory operation in Melo is presented as novel by itself. The distinction lies in an explicit and auditable return path from post-interaction evidence to the persistent state used by a later query. A-Mem evolves linked representations when new memories are inserted, whereas Melo separately admits declared corrections, tool evidence, and task outcomes after memory-supported behavior. D-Mem controls query-time access between vector retrieval and deeper deliberation, whereas Melo changes persistent state before a later query arrives. Qualified procedure events are consolidated while retaining their supporting identifiers. These distinctions concern trigger, timing, and provenance, not a claim that prior systems lack updating. Melo also does not replace specialized temporal reasoning, conflict resolution, or retrieval pruning. Its retention rule depends on write score and elapsed time rather than directly consuming feedback.

Cognitive theories provide three organizing constraints for this coordination. Working memory maintains task-relevant information over a limited interval (Atkinson & Shiffrin, 1968; Baddeley, 1992). Episodic, semantic, and procedural memory distinguish dated events, stable knowledge, and reusable skills (Tulving, 1972; Anderson et al., 2004). Selective attention determines which competing signals receive priority (Desimone & Duncan, 1995). Melo uses these distinctions as functional design constraints rather than claiming that the taxonomy itself is novel. Additional comparison details are retained in Appendix A.

2.2 Evaluation of Long-Term Agent Memory

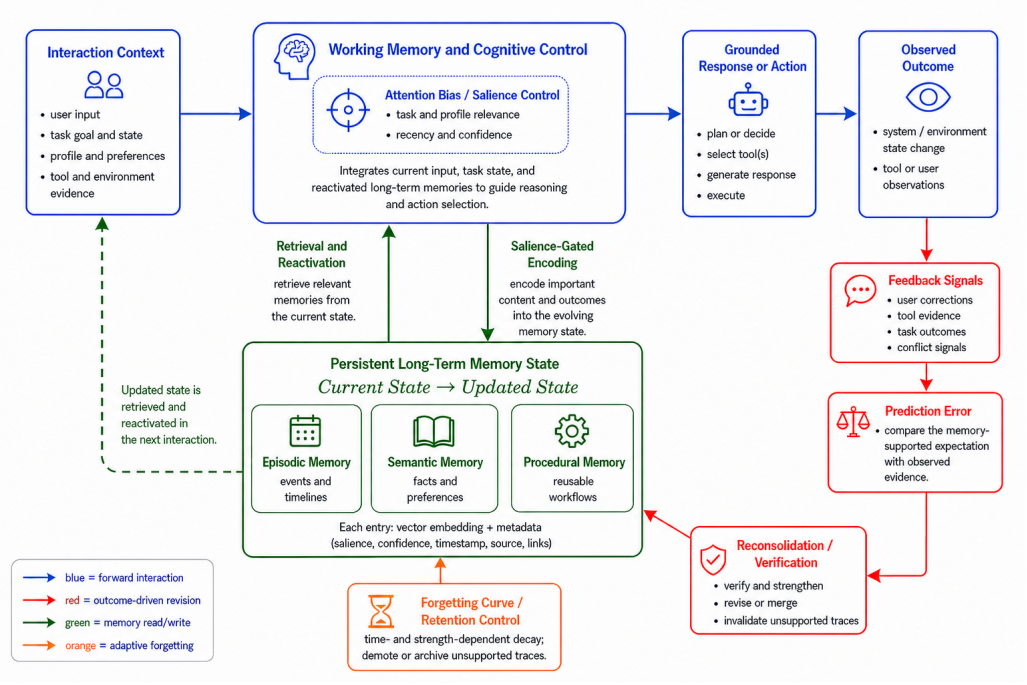
Existing benchmarks evaluate complementary aspects of long-term agent memory. LoCoMo assesses factual, temporal, and multi-hop reasoning over long multi-session conversations. MemBench (Tan et al., 2025) extends evaluation across memory levels, effectiveness, efficiency, and capacity.

Recent benchmarks examine user-specific behavior and downstream action. LoCoMo-Plus (Li et al., 2026b) tests whether agents apply implicit user states, goals, and constraints across long conversations. Mem2ActBench (Shen et al., 2026) tests whether memory guides tool selection and parameter grounding. CMLM-Assist complements them by evaluating profile retention, cross-session task continuity, reusable workflows, and memory-grounded responses in personal-assistant settings.

3 Method

Cognitive design mapping. Melo maps attentional bias to competition at admission and retrieval: records that match the current goal or profile receive bounded priority, while low-confidence input is suppressed. Prediction error motivates the return path from an observed correction or outcome to the memory that will support later behavior. An agent cannot observe a human internal expectation. The implementation therefore treats declared corrections, task outcomes, tool evidence, and conflict signals as observable discrepancy cues, rather than estimating a scalar prediction-error signal. Reconsolidation motivates evidence-linked revision of persistent state, while forgetting motivates time-dependent retention control. The mapping yields a system-level prediction: blocking the feedback-to-memory path should impair later behavior that depends on corrected or consolidated evidence. The matched ablation tests this operational prediction, not a hypothesis about human cognition.

Key insight and design challenges. A correction affects future behavior only if it enters the persistent state consulted by a later query. Implementing this return path raises three challenges. *Feedback identification* must not use evaluation answers, so Melo recognizes host-supplied user, tool, and task events from source and semantic tags and preserves their provenance. *Asynchronous arrival* must cover both immediate tool outcomes and corrections observed in later turns, so events update memory only when they arrive. *State propagation* must make the revision available to later queries, so admitted feedback participates in routing and procedural consolidation before the next retrieval. This interface does not infer arbitrary contradiction links; unresolved conflicts remain visible for clarification.



1 Retrieve → 2 Respond/act → 3 Observe feedback → 4 Update state → 5 Reuse updated state

Figure 1: Melo's static components and closed-loop workflow. Boxes are components or states; colored arrows are directed edges. Blue carries the forward interaction, green reads or writes memory, red returns observed evidence, and orange applies retention control. The prediction-error and reconsolidation boxes express conceptual responsibilities. In the evaluated implementation, source and semantic tags trigger feedback admission; no learned prediction-error magnitude or general record-merge solver is used. The numbered strip gives execution order.

3.1 Static Architecture

Figure 1 separates five responsibilities. The interaction adapter supplies the current query, task state, profile, and observable evidence. Working memory and cognitive control rank current and retrieved information. The persistent state stores episodic events, semantic facts or preferences, and procedural workflows with provenance. The backbone interface produces a grounded response or action. Finally, feedback admission, procedure consolidation, and retention control transform the current persistent state into the state available to later interactions.

3.2 Dynamic Workflow

At interaction t, Melo executes the numbered path in Figure 1: **(1)** retrieve a bounded evidence context from the current persistent state; **(2)** let the host backbone produce a response or action; **(3)** collect user corrections, tool evidence, task outcomes, or conflict signals when they become available; **(4)** admit eligible events, route them by memory role, consolidate qualified repeated procedures, and apply retention control; and **(5)** expose the updated state to later retrieval. An immediate tool result enters Step 3 in the current transition. A delayed user correction enters Step 3 in the later turn in which it is observed; the method does not retroactively use evidence before arrival.

For example, suppose the stored deadline "Monday" supports a reminder and the user later says "the deadline moved to Friday." The correction is admitted with its timestamp and source, so later retrieval can access the revised evidence. Melo does not automatically prove that two arbitrary records contradict each other; if both remain competitive, the grounding policy requests clarification. By contrast, when at least three eligible source-linked procedure records are available for a user, Step 4 can replace them with one procedural summary that retains all supporting identifiers.

3.3 Problem Definition

The input before interaction t is a cross-session history

$$\mathcal{H}_{<t} = ((x_i, y_i, o_i))_{i=1}^{t-1}, \tag{1}$$

where x_i, y_i, and o_i are respectively the user input, agent response, and observed tool or task state at interaction i. At interaction t, the method receives the current query q_t, persistent memory $\mathcal{M}_t$, profile u_t, and the set of feedback events that arrive during the transition,

$$\mathcal{B}_t = \{e_j : \text{arrive}(e_j) = t\}. \tag{2}$$

Its outputs are a response or action $\hat{a}_t$ and the next state $(\mathcal{M}_{t+1}, u_{t+1})$; the backbone parameters remain fixed. The evidence context and response are

$$\begin{aligned} C_t &= \text{Retrieve}(q_t, u_t, \mathcal{M}_t, B), \\ \hat{a}_t &= f_\theta(q_t, C_t, u_t, \pi_{\text{ground}}), \end{aligned} \tag{3}$$

where B is the retrieval budget and π_{ground} requires evidence-bounded generation and uncertainty when support is insufficient. Evidence is stored content with source and time metadata, not an evaluation answer. State transition is

$$(\mathcal{M}_{t+1}, u_{t+1}) = \text{Update}(\mathcal{M}_t, u_t, q_t, \hat{a}_t, \mathcal{B}_t). \tag{4}$$

The next interaction explicitly retrieves from the updated state:

$$C_{t+1} = \text{Retrieve}(q_{t+1}, u_{t+1}, \mathcal{M}_{t+1}, B). \tag{5}$$

3.4 Implemented Memory Operators

The host adapter first determines whether an arriving event belongs to the feedback return path:

$$h_j = \mathbb{1}[\text{source}(e_j) \in \mathcal{S}_{\text{fb}} \ \vee \ \text{tags}(e_j) \cap \mathcal{G}_{\text{fb}} \neq \varnothing], \tag{6}$$

where the declared source and tag vocabularies cover user feedback, corrections, task outcomes, task successes, and conflicts. This binary trigger uses no reference answer and is not a learned prediction-error magnitude. Selective encoding scores a candidate from supplied importance, confidence, profile overlap, and noise tags. Typed routing assigns an admitted record to an episodic, semantic, procedural, or working role. Three or more successful, source-linked procedure events may form one traceable workflow summary. Retention removes only unprotected records whose time- and write-score-dependent value falls below threshold; it does not consume feedback directly. Retrieval then ranks the remaining records and assembles a bounded context. All operators are deterministic for fixed records and configuration. Appendix B gives the complete rules and coefficients.

4 CMLM-Assist Benchmark Construction

4.1 Design Objectives

The assistant-memory benchmark tests profile preservation, current task-state maintenance, procedure reuse, and response grounding across repeated interactions. Concretely, a profile question may ask for a response preference after unrelated intervening events; a task-continuity question may ask for the latest milestone after an earlier state is revised; a procedure question asks how to repeat a workflow after several recorded successes; and an unsupported-query case should trigger abstention because no supporting record exists. These cases expose whether the memory system retains the current state rather than merely matching a nearby sentence.

Evidence-bearing questions are paired with supporting-memory identifiers; negative-abstention questions intentionally have none. Section C.2 defines the task-specific scores used during evaluation.

4.2 Scenario and Sample Construction

CMLM-Assist is programmatically constructed under a fixed seed from four task-oriented scenarios: research planning, writing assistance, course advising, and project management. Each of the 120 user

profiles is assigned one scenario, two long-term goals, two response preferences, and two domain interests. For every profile, we construct 34 timestamped memory facts. These include five scenario observations, two preference statements, two goal records, four task milestones, three tool-produced artifacts, four successful workflow repetitions, and 14 low-confidence distractors. Each fact retains its source, time index, entities or tags, confidence, and importance.

Construction proceeds from profiles to timestamped facts, questions, reference answers, and supporting identifiers. Deriving these fields from one scenario specification makes links checkable: preference questions target stored preferences, procedure questions target workflow repetitions, and abstention questions receive no supporting fact. The cases remain synthetic and template based.

Each user contributes 16 questions, with two questions for each of eight task types. Answers and supporting-memory identifiers are generated from the corresponding profile records, producing 1,920 questions in total. Table 4 summarizes the diagnostic target and size of each type.

4.3 Dataset Split and Quality Controls

We use a fixed 20/80 user-disjoint split: 24 users and 384 questions form the development partition, while 96 users and 1,536 questions form the held-out test partition. CMLM-Assist is not used to train model parameters; the smaller development set is reserved for adapter development and fixed-setting selection, and the larger test set provides the formal comparison. Reported results use one published test manifest rather than a user-chosen resplit. Both partitions share construction templates, so the split tests held-out users, not unseen scenario templates.

Answer fields are excluded from memory construction, retrieval, and generation; this answer isolation is the control that prevents target-answer leakage. Fixed inputs and shared scripts instead support comparability and auditability. The runner checks split membership, identifiers, evidence links, and row coverage, and excludes incomplete or duplicate predictions. These checks establish structural consistency, not semantic validity or real-world representativeness.

Evaluation dependence. The same team developed Melo and CMLM-Assist, so its templates and supplied metadata may favor Melo even with answer isolation. Supplied confidence and noise tags are also easier to use than trustworthiness inferred from raw conversation. We therefore treat CMLM-Assist as controlled diagnostic evidence and interpret it together with public LoCoMo, matched ablations, and failure cases; independent testing on natural histories and new templates remains necessary.

The complete task composition and task-specific metric definitions are retained in Appendix C.

5 Experiments

The evaluation asks three questions. First, does Melo improve end-to-end conversational QA on the independent public LoCoMo benchmark? Second, how does it behave on controlled profile, state-update, procedure, and grounding tasks in CMLM-Assist? Third, what changes when individual return-path functions are blocked under a matched protocol? The failure analysis separates observed gains from unsupported claims of indefinite online evolution or general conflict resolution.

5.1 Evaluation Protocol

5.1.1 Datasets and Objectives

LoCoMo contains 1,542 questions covering factual, temporal, single-hop, multi-hop, and open-domain reasoning over extended multi-session conversations. We use it to assess general long-term conversational memory, including factual retention, evidence composition, and resistance to long-history distraction.

CMLM-Assist is described in Section 4. Its fixed 1,536-question test set evaluates profiles, evolving goals and task states, reusable procedures, and evidence-bounded responses. Adding labels to LoCoMo could support some diagnostics, but the required repeated workflow successes, controlled state revisions, unsupported-query cases, and evidence identifiers are not consistently present in its

selected histories; adding them would require altering the conversations rather than only relabeling them. We therefore keep LoCoMo unchanged as an independent public comparison and use CMLM-Assist only for controlled diagnostics.

The evaluation compares Melo with six baselines, namely Vanilla RAG, Mem0, A-Mem, MemoryOS, MemoryBank, and MemGPT/Letta, across both benchmarks and three language-model backbones: DeepSeek-V4-Flash, Qwen3-14B, and GPT-4o-mini. Each backbone generates answers from the context supplied by the memory system. CMLM-Assist diagnostics examine personalized-memory behavior, while controlled ablations and error analysis identify component contributions and remaining limitations. Both benchmarks use fixed question sets and offline scoring, so these experiments do not establish indefinite online self-evolution.

5.1.2 BASELINES

All six baselines are evaluated on both datasets and all three backbones. Vanilla RAG provides the retrieval reference, while the other five represent long-term memory systems. Full Context remains a separate long-context reference and is excluded from the main comparison because it does not construct or retrieve explicit memories.

5.1.3 EVALUATION METRICS

On CMLM-Assist, accuracy is assessed through answer quality and evidence access. Consistency is assessed through goal alignment, temporal updating, and conflict resolution. Retention is assessed through cross-session profile recall and procedural reuse within the recorded histories, rather than indefinite online operation.

We evaluate answer quality with EM, token-level F1, BLEU-1, ROUGE-L, and METEOR (Papineni et al., 2002; Lin, 2004; Banerjee & Lavie, 2005). EM is a strict surface-form diagnostic: after lowercasing and token normalization, it assigns credit only when the complete prediction sequence equals the reference. Added qualifiers, different date orders, or reordered lists therefore receive zero EM credit despite retaining correct content. F1 is the primary cross-dataset metric, while the remaining metrics provide complementary lexical and sequence views. Recall@k and MRR measure evidence access (Manning et al., 2008); they are reported only for compatible identifier spaces and retain each system's actual cutoff.

CMLM-Assist reports task-specific diagnostics defined fully in Appendix C.2. Goal Alignment is mean token-level F1 over the 192 held-out goal–task questions. Personalization is mean token-level F1 over 576 profile, preference, and multi-hop personalization questions. Memory Hit@k, Procedural Formation, and Storage Size describe evidence access and internal memory behavior. These scores are deterministic summaries, not independent learned judges, and do not establish causal use of a retrieved item.

Relative improvements in the result text are computed against the named baseline. For example, the LoCoMo F1 increase from 43.68% to 49.86% corresponds to a relative improvement of 14.1%. The tables retain the original metric scores so that each comparison can be verified directly.

5.1.4 COMPARISON PROTOCOL AND RESULT ADMISSION

Across both datasets and all three backbones, Melo is evaluated against the six baselines using the same evaluator, decoding parameters, and result-admission criteria within each reported block. Melo uses top-$k = 20$ and Vanilla RAG uses top-$k = 5$; external systems retain their exposed native settings. On CMLM-Assist, Melo's low-similarity gate can expand the returned budget to at most 60 records. The main tables therefore compare complete systems under declared configurations, not equal-k or equal-token-budget retrieval. LoCoMo's Qwen block uses a common 1,529-question subset; other reported rows retain their traceable coverage. Retrieval diagnostics are interpreted only within compatible identifier spaces.

Raw conversations, user facts, and QA files are locked before method adaptation. Retrieval, reranking, and post-processing cannot access target-answer fields. External-system rows are included only when complete test-question coverage and provenance checks pass. Smoke tests, partial runs, failed outputs, and substitute predictions are excluded from the main tables.

Table 1: F1 on LoCoMo and CMLM-Assist under declared, non-equal retrieval configurations. DS, Qwen, and GPT denote DeepSeek-V4-Flash, Qwen3-14B, and GPT-4o-mini. The LoCoMo Qwen block uses a common 1,529-question subset. Full metrics and coverage notes are in Appendix D.

	LoCoMo			CMLM-Assist		
Method	DS	Qwen	GPT	DS	Qwen	GPT
Vanilla RAG	43.68	31.20	28.91	28.63	15.52	18.02
Mem0	36.26	21.26	28.54	5.52	18.06	20.58
A-Mem	34.53	34.26	28.41	18.77	17.46	14.34
MemoryOS	42.02	38.00	28.27	15.79	12.99	10.40
MemoryBank	22.20	18.24	20.48	1.34	2.24	11.84
MemGPT/Letta	12.12	10.38	9.53	17.02	9.90	9.33
Melo	**49.86**	**45.08**	**31.28**	**38.94**	**28.13**	**46.59**

Table 2: Matched ablation on the 1,536-question CMLM-Assist test set. ΔF1 is relative to the matched Melo control; intervals are paired 95% bootstrap confidence intervals.

Variant	F1	ΔF1 [95% CI]	Direct diagnostic: control $\rightarrow$ variant
Melo (matched control)	77.00	–	–
w/o Feedback-to-Memory Update	66.44	-10.56 [-12.08, -9.04]	Procedures formed: 96 $\rightarrow$ 0
w/o Layered Orchestration	72.14	-4.86 [-6.59, -3.15]	MRR: 34.81 $\rightarrow$ 29.64
w/o Consolidation	75.22	-1.78 [-3.19, -0.39]	Procedural F1: 86.74 $\rightarrow$ 50.77
w/o User Profile	56.28	-20.72 [-22.93, -18.56]	Recall@20: 93.75 $\rightarrow$ 50.00

Appendix E documents protocol and leakage controls, prompt boundaries, and supplementary reproducibility artifacts. Appendix Tables 8, 9, and 10 specify the auditable intermediate states, result-admission controls, and permitted prompt inputs and outputs.

5.2 LoCoMo Results

Melo ranks above all six comparison methods in F1, BLEU-1, ROUGE-L, and METEOR under every backbone (Appendix Table 5). Its F1 scores are 49.86%, 45.08%, and 31.28% with DeepSeek-V4-Flash, Qwen3-14B, and GPT-4o-mini, giving relative margins of 14.1%, 18.6%, and 8.2% over the strongest non-Melo F1 row. These are end-to-end system comparisons under declared native configurations, not equal-budget retrieval comparisons.

5.3 CMLM-Assist Results

Every CMLM-Assist result covers the same 1,536 test questions and passes row-coverage and provenance checks. Melo ranks first in F1, BLEU-1, ROUGE-L, METEOR, Goal Alignment, and Personalization under all three backbones (Appendix Table 6). Its F1 scores are 38.94%, 28.13%, and 46.59%, compared with the nearest non-Melo scores of 28.63%, 18.06%, and 20.58%. Because the authors designed this benchmark, these results are controlled diagnostic evidence rather than independent validation.

5.4 Core Component Ablation

Table 2 reports a separate matched GPT-4o-mini ablation. All rows use the same TF–IDF backend, top-$k = 20$, concise-answer prompt, decoding settings, and evaluator, with quality gating disabled. Its matched-control F1 of 77.00 is therefore not interchangeable with the 46.59 main-comparison score. "w/o Feedback-to-Memory Update" prevents source- or tag-identified feedback events from entering persistent memory and thereby disables feedback-conditioned procedure formation; it does *not* remove ordinary writing or retrieval, and test predictions are never written back. The other variants remove typed routing and retrieval, procedure consolidation, or the persistent user profile.

All four removals lower F1 with confidence intervals below zero. Blocking feedback-to-memory update reduces F1 by 10.56 points. The accompanying change from 96 formed procedures to zero

verifies the operational dependency encoded by this variant; it is not an independent causal test because procedure formation requires admitted feedback records by definition. Consolidation has a smaller aggregate effect, but procedural-question F1 falls by 35.97 points, showing that its benefit is concentrated rather than uniform.

5.5 Non-obvious Findings and Failure Cases

The experiments expose four qualifications that are not visible from the headline F1 table. First, EM does not always track overlap metrics: on GPT-4o-mini LoCoMo, Melo raises F1 from 28.91% to 31.28% but has lower EM than Vanilla RAG (2.20% versus 3.24%); its longer outputs are consistent with this divergence but do not exclude content errors. Second, the user-profile removal causes the largest ablation drop, but this ordering may reflect the profile-heavy constructed tasks rather than a universal ranking of memory mechanisms. Third, consolidation's effect is concentrated on procedural questions, as Table 2 shows. Finally, Melo still performs much worse on LoCoMo multi-hop than single-hop questions and remains below 8% conflict-resolution F1 on every CMLM-Assist backbone, so evidence access does not yet guarantee correct composition or contradiction resolution.

6 Limitations

The main comparison covers two benchmarks, three backbones, and six baselines, but it is an end-to-end system comparison rather than a retrieval-budget-controlled study. External systems retain native memory construction and retrieval settings, while Melo and Vanilla RAG use different top-k values and record lengths; Melo can also expand retrieval on low-similarity CMLM-Assist queries. The evidence therefore supports comparative performance under declared configurations, not equal-context or isolated retrieval superiority. The present study reports no equal-budget experiment.

CMLM-Assist replays a fixed interaction history and cannot establish indefinite online memory evolution. The controlled ablation is limited to GPT-4o-mini, uses a separate matched protocol, and does not isolate selective encoding, forgetting, score gating, the question router, or component interactions. Goal Alignment and Personalization measure task behavior but do not prove that a particular record caused the response. We also do not report a length-normalized or precision-focused analysis, so the observed F1–EM divergence cannot be attributed solely to answer length. The low conflict-resolution scores show that tagged feedback is not a general contradiction-resolution algorithm.

7 Conclusion

This paper presented Melo, a cognitive science–inspired framework for coordinating long-term memory operations in personalized AI agents. Its closed-loop mechanism connects memory-supported interaction to later, explicitly recorded evidence and then to the persistent state used for subsequent retrieval. The evaluated implementation combines selective admission, typed routing, evidence-linked retrieval, procedural consolidation, and deterministic retention without exposing evaluation answers to the update rule. CMLM-Assist complements public LoCoMo with controlled diagnostics of task continuity, profile retention, procedure reuse, and memory-grounded response generation.

Across six baselines, Melo obtains the highest F1, BLEU-1, ROUGE-L, and METEOR scores in every evaluated LoCoMo and CMLM-Assist backbone block under the declared configurations, while EM remains mixed. It also ranks first in Goal Alignment and Personalization among the evaluated CMLM-Assist systems. In the separate matched GPT-4o-mini ablation, blocking feedback-to-memory update lowers F1 by 10.56 points. Removing layered orchestration, procedural consolidation, or the user profile also lowers F1. This pattern is consistent with the system-level prediction derived from the cognitive mapping, but it is not evidence of biological equivalence. The findings do not establish equal-budget superiority, independent validation of the author-constructed benchmark, indefinite online evolution, or general conflict resolution.

AI USE STATEMENT

Generative AI tools were used to provide feedback on descriptions of methodology and experiments, assist in software implementation and code editing, support manuscript restructuring and drafting, perform language refinement and LaTeX formatting, and format references. These tools were not used to fabricate experimental observations or manually alter reported results. All AI-assisted text and code were reviewed by the authors. All reported metrics were cross-checked against stored experimental artifacts. The authors retain full responsibility for all claims, references, software, and submitted materials.

ETHICS AND BROADER-IMPACT STATEMENT

Personal assistants that retain outdated preferences or lose established workflows may force users to repeat information and correct recurring mistakes. Linking stored information to later feedback may improve consistency across sessions. Melo keeps provenance for retained records and consolidated procedures, which supports inspection of how a persistent state was formed. CMLM-Assist provides controlled tasks for examining profile retention, task continuity, and procedure reuse alongside public conversational-memory evaluation. Persistent personalization nevertheless creates risks of retaining sensitive information or applying outdated preferences. The present results do not establish privacy protection, reliable resolution of arbitrary contradictions, or sustained performance in live deployment. Such uses require independent evaluation and explicit user control over retained information.

Because the authors designed both Melo and CMLM-Assist, the paper treats CMLM-Assist as controlled, author-constructed evidence rather than independent validation. The public LoCoMo evaluation, answer isolation, provenance checks, and matched ablation reduce but do not remove this evaluation dependence.

REPRODUCIBILITY STATEMENT

The method section defines the implemented memory transition and its evidence boundary. The appendices preserve the original detailed operators, benchmark definitions, complete result tables, implementation settings, audit artifacts, and prompt boundaries. The anonymous supplement contains fixed identifiers and de-identified records, configurations, evaluation scripts, aggregate results, and provenance summaries; it excludes credentials, private conversations, caches, and incomplete runs.

REFERENCES

John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060, 2004. doi: 10.1037/0033-295X.111.4.1036.

Richard C. Atkinson and Richard M. Shiffrin. Human memory: A proposed system and its control processes. In *Psychology of Learning and Motivation*, volume 2, pp. 89–195. Academic Press, 1968. doi: 10.1016/S0079-7421(08)60422-3.

Alan Baddeley. Working memory. *Science*, 255(5044):556–559, 1992. doi: 10.1126/science.1736359.

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. APEX-MEM: Agentic semi-structured memory with temporal reasoning for long-term conversational AI. *arXiv preprint arXiv:2604.14362*, 2026. doi: 10.48550/arXiv.2604.14362. URL https://arxiv.org/abs/2604.14362.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2005.

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.

Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222, 1995. doi: 10.1146/annurev.ne.18.030195.001205.

Joaquín M. Fuster. Upper processing stages of the perception–action cycle. *Trends in Cognitive Sciences*, 8(4):143–145, 2004. doi: 10.1016/j.tics.2004.02.004.

Joaquín M. Fuster and Steven L. Bressler. Past makes future: Role of pFC in prediction. *Journal of Cognitive Neuroscience*, 27(4):639–654, 2015. doi: 10.1162/jocn_a_00746.

Yuyang Hu et al. Memory in the age of AI agents. *arXiv preprint arXiv:2512.13564*, 2025. doi: 10.48550/arXiv.2512.13564. URL https://arxiv.org/abs/2512.13564.

Jiarui Kang, Mingyan Ji, Ziyuan Zhao, and Tianle Bai. Memory OS of AI agent. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, 2025. doi: 10.18653/v1/2025.emnlp-main.1318.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

Jonathan L. C. Lee. Memory reconsolidation mediates the strengthening of memories by additional learning. *Nature Neuroscience*, 11:1264–1266, 2008. doi: 10.1038/nn.2205.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Kai Li, Xuanqing Yu, Ziyi Ni, Yi Zeng, Yao Xu, Zheqing Zhang, Xin Li, Jitao Sang, Xiaogang Duan, Xuelei Wang, Chengbao Liu, and Jie Tan. TiMem: Temporal-hierarchical memory consolidation for long-horizon conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21700–21720, 2026a. doi: 10.18653/v1/2026.findings-acl.1091.

Yifei Li, Weidong Guo, Lingling Zhang, Rongman Xu, Muye Huang, Hui Liu, Lijiao Xu, Yu Xu, and Jun Liu. LoCoMo-Plus: Beyond-factual cognitive memory evaluation framework for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 25085–25100, 2026b. doi: 10.18653/v1/2026.acl-long.1150.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, pp. 74–81, 2004.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638.

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*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Karim Nader, Glenn E. Schafe, and Joseph E. LeDoux. Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. *Nature*, 406:722–726, 2000. doi: 10.1038/35021052.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002. doi: 10.3115/1073083.1073135.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. doi: 10.1145/3586183.3606763.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016. doi: 10.18653/v1/D16-1264.

Daniela Schiller, Marie-H. Monfils, Candace M. Raio, David C. Johnson, Joseph E. LeDoux, and Elizabeth A. Phelps. Preventing the return of fear in humans using reconsolidation update mechanisms. *Nature*, 463:49–53, 2010. doi: 10.1038/nature08637.

Dieuwke Sevenster, Tom Beckers, and Merel Kindt. Prediction error governs pharmacologically induced amnesia for learned fear. *Science*, 339(6121):830–833, 2013. doi: 10.1126/science.1231357.

Y. Shen, K. Li, W. Zhou, and S. Hu. Mem2ActBench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous agents. *arXiv preprint arXiv:2601.19935*, 2026. doi: 10.48550/arXiv.2601.19935. URL https://arxiv.org/abs/2601.19935.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hongjin Tan, Zeyang Zhang, Chao Ma, Xiliang Chen, Qian Dai, and Zenglin Dong. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19336–19352, 2025. doi: 10.18653/v1/2025.findings-acl.989.

Endel Tulving. Episodic and semantic memory. In Endel Tulving and Wayne Donaldson (eds.), *Organization of Memory*, pp. 381–403. Academic Press, 1972.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b. doi: 10.1007/s11704-024-40231-1.

Tan Wang and Yunwei Dong. MGRetrieval: Memory-guided reflective retrieval for long-term dialogue agents. *arXiv preprint arXiv:2605.27437*, 2026. doi: 10.48550/arXiv.2605.27437. URL https://arxiv.org/abs/2605.27437.

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*, volume 38, 2025.

Ke Yang, Zixi Chen, Xuan He, Jize Jiang, Michel Galley, Chenglong Wang, Jianfeng Gao, Jiawei Han, and ChengXiang Zhai. PlugMem: A task-agnostic plugin memory module for LLM agents. *arXiv preprint arXiv:2603.03296*, 2026. doi: 10.48550/arXiv.2603.03296. URL https://arxiv.org/abs/2603.03296.

Zhixing You, Jiachen Yuan, and Jason Cai. D-Mem: A dual-process memory system for LLM agents. *arXiv preprint arXiv:2603.18631*, 2026. URL https://arxiv.org/abs/2603.18631.

Zeyang Zhang, Qian Dai, Xinyi Bo, Chao Ma, Rui Li, Xiliang Chen, Jieming Zhu, Zenglin Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025. doi: 10.1145/3748302.

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. doi: 10.1609/aaai.v38i17.29946.

A Additional Related Work

A.1 Long-Term Memory and Context Access for LLM Agents

Recent memory systems for language-model agents can be grouped by the long-term behavior they emphasize. Generative Agents links observations with reflection and planning, while Reflexion stores feedback from failed attempts as self-reflection. Voyager (Wang et al., 2024a) converts exploration experience into an executable skill library. These systems establish several forms of experience reuse, although their update events and persistent-state interfaces differ from Melo's explicit post-interaction return path.

Persistent memory-management systems treat memory as an external resource that must be organized and updated across sessions. MemGPT provides virtual context management, while MemoryBank maintains profiles, summaries, and forgetting. Mem0 extracts and consolidates conversational information, and MemoryOS organizes storage, updating, retrieval, and generation as a memory service. A-Mem (Xu et al., 2025) represents memories as dynamically linked notes. PlugMem (Yang et al., 2026) converts interaction experience into reusable knowledge through a task-agnostic plugin. The relevant distinction is therefore not whether these systems update memory, but which events trigger revision, when the revision occurs, and whether its provenance remains inspectable.

Temporal and adaptive-retrieval systems address different access problems. TiMem (Li et al., 2026a) uses a temporal hierarchy and complexity-aware recall; APEX-MEM (Banerjee et al., 2026) represents evolving and conflicting facts. D-Mem (You et al., 2026) gates expensive retrieval, while MGRetrieval (Wang & Dong, 2026) reflects over historical structure. These methods primarily alter how stored history is organized or accessed; Melo instead focuses on the auditable return of post-interaction evidence to persistent state.

Alongside explicit memory systems, retrieval-augmented generation (RAG) and long-context prompting supply language models with information beyond their parameters. RAG (Lewis et al., 2020) retrieves query-relevant passages from an external collection and places them in the generation context. Dense passage retrieval (Karpukhin et al., 2020) identifies relevant evidence in large collections. Conventional RAG assumes that the collection already exists. It does not decide how interaction-derived information should be retained, updated, consolidated, or removed, nor does it maintain evolving user states or reusable procedures.

Long-context prompting instead places much or all of the interaction history in the model context. This approach preserves the original conversation and avoids explicit memory construction, but its cost grows with history length. The analysis in Lost in the Middle (Liu et al., 2024) shows that models may miss relevant evidence when it appears among redundant or distant context.

RAG and long-context prompting provide useful alternatives for historical information access, but neither explicitly manages how interaction-derived information is formed, updated, and reused across sessions. We treat them as retrieval and context-access approaches rather than complete long-term memory systems.

B Detailed Method Operators

B.1 Selective Encoding

Melo avoids indiscriminate storage. Importance and confidence are input-adapter metadata, not quantities estimated from reference answers. Text tokens come from the candidate record, and profile tokens from stored user attributes. For a candidate memory unit m_i, let $a_i \in [0,1]$ be its supplied importance, $c_i \in [0,1]$ its source confidence, $\mathcal{T}_i$ its normalized text-token set, and $\mathcal{U}_i$ the token set derived from the user's stored goals, preferences, and domain attributes. The implemented write score is

$$w_i = \mathrm{clip}(0.52a_i + 0.30c_i + b_i - n_i, 0, 1)\,, \tag{7}$$

where the profile bonus is

$$b_i = \min(0.18, 0.025|\mathcal{T}_i \cap \mathcal{U}_i|)\,, \tag{8}$$

Table 3: Four memory layers in Melo.

Layer	Stored content	Temporal scope	Main operations
Working	Current task state, structured plan state, recent tool results	Very short term	Read, update, clear, promote
Episodic	Timestamped events, user actions, task outcomes	Medium term	Event writing, entity/time retrieval, source tracing
Semantic	Stable facts, preferences, concepts, verified summaries	Long term	Store, summarize, retrieve
Procedural	Reusable workflows, tool-use routines, task policies	Long term	Consolidate, retrieve, invoke

and the noise penalty is

$$n_i = \begin{cases} 0.28, & m_i \text{ is tagged as noise or } c_i < 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The profile-overlap bonus operationalizes attentional priority at the memory-management level, while the noise penalty suppresses low-confidence fragments. Neither operation changes the internal attention weights of the backbone model or claims to reproduce human selective attention. Section D.1.1 states how the fixed coefficients were selected and what the current evidence does not validate.

The write decision is

$$z_i = \begin{cases} 1, & w_i \geq \tau_w, \\ 0, & w_i < \tau_w, \end{cases} \tag{10}$$

where τ_w denotes the write threshold for admitting a candidate into persistent memory. Candidates below τ_w remain available in working memory for the current task but are not promoted automatically. A higher τ_w makes memory admission more conservative, whereas a lower value admits more candidate memories.

B.2 Memory Forms and Routing

Melo uses four memory types for different information roles and time scales, as summarized in Table 3. Working memory stores current task states and recent tool results. Episodic memory stores timestamped events. Semantic memory stores stable facts, preferences, and concepts. Procedural memory stores reusable workflows and task strategies. These labels organize the implemented store; they do not imply a neural memory architecture.

Melo routes records using semantic tags supplied by the input adapter. Tags are explicit adapter inputs: Melo does not infer them from task labels or reference answers. The reported CMLM-Assist adapter generates them from the scenario specification, while legacy-source adapters map available source fields into the same vocabulary. The deterministic rule makes routing reproducible and keeps each assignment inspectable:

$$\ell_i = \begin{cases} P, & \mathcal{G}_i \cap \{\text{procedure, workflow}\} \neq \varnothing, \\ S, & \mathcal{G}_i \cap \{\text{profile, preference, goal, university}\} \neq \varnothing, \\ E, & \mathcal{G}_i \cap \mathcal{G}_{\text{event}} \neq \varnothing, \\ W, & \text{otherwise,} \end{cases} \tag{11}$$

where $\mathcal{G}_i$ is the tag set of m_i and $\mathcal{G}_{\text{event}}$ contains dialogue, timeline, project-memory, and tool-memory tags. The symbols W, E, S, and P denote working, episodic, semantic, and procedural roles. This rule keeps temporary states, dated events, stable profile knowledge, and reusable procedures distinct without implying that the four-way taxonomy itself is novel.

B.3 Consolidation and Memory Revision

B.3.1 Consolidation and Conflict Revision

The implemented consolidation rule targets repeated successful procedures. For user u, define

$$\mathcal{P}_u = \{m_i : \text{user}(m_i) = u,\ h_i = 1, \\ \mathcal{G}_i \cap \{\text{procedure, workflow}\} \neq \varnothing\}, \tag{12}$$

where h_i is one when the source or tags of m_i satisfy the feedback-return test defined above, and zero otherwise. If $|\mathcal{P}_u| \geq 3$, Melo replaces these raw procedure events with one procedural summary. The summary retains the latest event day, all supporting identifiers, and the workflow phrase most consistently observed in the group. Its importance and confidence are fixed at 0.88 and 0.90 rather than estimated as probabilities; Section D.1.1 explains their role and evidential limit. Every formed procedure remains traceable to the records it replaces.

The update interface also accepts correction and conflict signals, but Melo does not define a general learned merge/split operation over arbitrary contradictory records. Such records remain identifiable through their source and tags; conflict-oriented retrieval gives additional priority to auditable tool, project, and verification evidence. If the evidence remains inconsistent, the generator is instructed to express uncertainty or request clarification. The low conflict-resolution results reported later therefore do not establish a complete conflict solver.

B.3.2 Time-Dependent Retention Control

Without retention control, low-value and outdated records accumulate over time. Melo uses a deterministic retention score

$$\text{Ret}_i = w_i \exp(-0.04 d_i), \tag{13}$$

where w_i is the write score and d_i is the elapsed days between the record timestamp and the current memory-update time. A non-protected record is retained only when $\text{Ret}_i \geq 0.30$. Procedure, workflow, profile, preference, goal, and university records are protected because infrequency alone does not make them unimportant.

B.4 Retrieval and Scheduling

For query q_t, Melo first obtains embedding-similarity candidates and then applies a deterministic query bonus:

$$\text{score}(q_t, m_i) = \text{sim}(q_t, m_i) + B_{\text{intent}}(q_t, m_i) \\ + B_{\text{lex}}(q_t, m_i) - B_{\text{noise}}(m_i). \tag{14}$$

The intent term uses only query text. Its principal bonuses are 0.45 for a matching profile or preference record, 0.45 for a matching goal, 0.40 for a direct procedure record, and 0.55 for a consolidated profile summary on a multi-evidence query. The lexical term is capped at 0.18 and adds 0.025 per shared normalized token; a noise tag subtracts 0.35. The rule also contains smaller, fixed bonuses for tool, project, temporal, and audit evidence. No task label or reference answer enters this score.

Before context construction, an implementation-level cutoff removes negligible candidates. Let $s_i = \text{score}(q_t, m_i)$, $s_{\max} = \max_i s_i$, and

$$\tau_r = \max(0.06, \min(0.20, 0.20 s_{\max})) . \tag{15}$$

The retained candidate set is

$$\mathcal{R}(q_t) = \{m_i : s_i \geq \tau_r\}. \tag{16}$$

A higher τ_r restricts the context to stronger matches, whereas a lower value preserves candidate coverage. Noise-tagged records are removed. If fewer than two records survive, the two highest-scoring records are retained to avoid an empty or singleton context. Dataset-specific exceptions are listed with the evaluation configuration.

The scheduler forms a candidate pool before selecting the final evidence context. Its pool size, fallback rule, and optional deeper-retrieval trigger are configuration choices specified in the evaluation section.

The grounding module constrains final generation. Factual answers must align with retrieved evidence. Preference-related answers should prioritize explicitly confirmed profile fields. Procedural tasks

Table 4: Composition of the CMLM-Assist benchmark.

Question type	Diagnostic target	QA
Profile recall	Retain stable user and scenario information	240
Preference recall	Apply persistent user preferences and constraints	240
Goal–task alignment	Use long-term goals to guide the current task	240
Procedural memory	Reuse a validated multi-step workflow	240
Temporal update	Prefer the current state over an outdated record	240
Conflict resolution	Resolve contradictory memories using source and time	240
Multi-hop personalization	Combine profile cues and multiple memory facts	240
Negative abstention	Abstain when no supporting memory is available	240

should use validated procedural memory instead of reconstructing the workflow from scratch. If evidence is insufficient, the agent should request clarification, abstain, or express uncertainty.

B.5 Cognitive-to-Computational Mapping

The forward phase maps the selected memory state to observable agent behavior. Retrieved evidence constrains factual content, the current profile supplies user-specific goals and preferences, the task state identifies the active objective, and procedural memory provides reusable action sequences. Melo exposes these elements as structured context and explicit policies rather than modifying or storing the backbone's hidden reasoning process.

The user profile u_t stores relatively stable information, including long-term goals, interests, roles, response preferences, and common workflows. Admitted profile evidence retains its source, day, confidence, and tags. Melo also constructs one evidence-derived profile summary per user when goal or preference records are available.

During encoding, the overlap bonus b_i raises the write score of records that match stored goals, preferences, or domain attributes. Tag routing sends profile, preference, and goal records to semantic memory. During retrieval, the question-derived intent bonus raises matching profile or goal records, subject to the fixed caps defined above. These bonuses cannot introduce a fact absent from the recorded history, but the present evaluation does not test all forms of irrelevant or excessive personalization.

Given query q_t, retrieved memory $M_t^\star$, user profile u_t, and task state h_t, the profile-aware context builder constructs

$$C_t^\star = g(q_t, M_t^\star, u_t, h_t; B)\,, \tag{17}$$

where g is deterministic for fixed inputs. For profile-, goal-, multi-evidence-, or conflict-oriented questions, it places one evidence-derived profile summary before the selected direct evidence when available. For procedural questions, it places one consolidated workflow before the direct evidence. The final output is

$$\hat{a}_t = f_\theta(q_t, C_t^\star, \pi_{\text{ground}})\,. \tag{18}$$

Here, π_{ground} denotes the grounding policy. This design lets memory affect evidence selection, goal constraints, procedure reuse, and answer organization. The arriving feedback set $\mathcal{B}_t$ completes the path from memory use to later memory revision.

C Additional CMLM-Assist Details

C.1 Complete Task Composition

C.2 Task-Specific Evaluation Metrics

CMLM-Assist derives its diagnostics from predefined task types, references, and supporting-memory annotations. For question q, let $\hat{y}_q$ and y_q denote the generated and reference answers, $\mathcal{R}_k(q)$ the top-k retrieved identifiers, and $\mathcal{E}_q$ the supporting identifiers. We use standard token-level $F_1(\hat{y}_q, y_q) \in [0, 1]$ after lowercase token normalization (Rajpurkar et al., 2016). All annotations are available only during post-generation evaluation.

C.2.1 GOAL ALIGNMENT

Goal Alignment measures whether a response applies a persistent user goal to the current task. The set $\mathcal{Q}_{\text{goal}}$ contains 192 goal–task-alignment questions whose references are derived from user goal records. We compute

$$\text{GoalAlign} = \frac{100}{|\mathcal{Q}_{\text{goal}}|} \sum_{q \in \mathcal{Q}_{\text{goal}}} F_1(\hat{y}_q, y_q). \tag{19}$$

A higher score indicates better agreement with persistent goals on task-sensitive questions; it does not independently prove causal use of the stored goal.

C.2.2 PERSONALIZATION

Personalization measures whether responses recover and combine user-specific information. We define $\mathcal{Q}_{\text{pers}} = \mathcal{Q}_{\text{profile}} \cup \mathcal{Q}_{\text{preference}} \cup \mathcal{Q}_{\text{multi}}$, covering profile recall, preference recall, and multi-hop personalization. Each set contains 192 questions, giving $|\mathcal{Q}_{\text{pers}}| = 576$. References come from stable profile fields, persistent preferences, or multiple linked profile and memory records. The score is

$$\text{Personalization} = \frac{100}{|\mathcal{Q}_{\text{pers}}|} \sum_{q \in \mathcal{Q}_{\text{pers}}} F_1(\hat{y}_q, y_q). \tag{20}$$

A higher value indicates more accurate use of retained attributes, preferences, and jointly required memories. Negative-abstention questions are excluded because they assess evidence-bounded abstention rather than positive personalization.

C.2.3 MEMORY AND STORAGE DIAGNOSTICS

Let $\mathcal{Q}_E$ contain questions with annotated supporting memories. Memory Hit@k records whether retrieval exposes any supporting item:

$$\text{MH@}k = \frac{100}{|\mathcal{Q}_E|} \sum_{q \in \mathcal{Q}_E} \mathbb{1}[\mathcal{R}_k(q) \cap \mathcal{E}_q \neq \varnothing]\,. \tag{21}$$

Each result retains the system's declared cutoff and is reported only when retrieved and supporting identifiers share a compatible space.

Procedural Formation is an operational consolidation diagnostic. Let $\mathcal{W}_u$ contain feedback-return records tagged as procedural or workflow evidence for user u. One workflow is formed when at least three such records are available:

$$\text{PF} = \sum_{u \in \mathcal{U}} \mathbb{1}[|\mathcal{W}_u| \geq 3]\,. \tag{22}$$

Storage Size is $|\mathcal{M}_{\text{persist}}|$, the records remaining after encoding, consolidation, and forgetting. Both diagnostics require compatible write traces and describe internal memory behavior rather than answer quality, latency, bytes, or monetary cost.

D ADDITIONAL EXPERIMENTAL DETAILS

D.1 IMPLEMENTATION DETAILS

Offline replay and adapter configuration. The experiments process recorded histories in temporal order and admit recorded feedback during memory construction. Test predictions are not written back, and reference answers and task labels are excluded from memory operations. This evaluates replay of the update rule, not a live system learning indefinitely from its own outputs. The benchmark is an input source for this test harness, not a component required by Melo in deployment.

The adapters supply record text, timestamps, importance, confidence, and semantic tags. CMLM-Assist supplies these fields programmatically; a host application would need to supply equivalent metadata. Melo does not learn their calibration. Legacy LoCoMo source tags are mapped to episodic records and protected from deletion. These dataset-specific settings limit claims about forgetting and transfer to other input schemas.

For valid CMLM-Assist records with confidence at least 0.50, Melo preserves the record and records the following diagnostic penalty for older, non-profile items:

$$p_i = \min\big(0.14, 0.22(1 - e^{-0.04d_i})(1 - \min(1, w_i))\big) . \tag{23}$$

The penalty is zero for new or high-write-score records and grows with age and weakness. In the current code it is stored for audit and is not subtracted again by the retrieval scorer. This is the implemented retention boundary; the current experiments do not estimate a learned memory-strength update rate from test outcomes.

The scheduler targets $\max(3k, k + 8)$ candidates before truncation, capped by the number of available records. For CMLM-Assist, a highest raw similarity below 0.20 marks the query for deeper retrieval. The returned budget can then expand from k to at most $3k$. LoCoMo searches the available conversation records and does not use this deliberation trigger.

Melo uses top-$k = 20$ to cover multi-evidence questions within bounded context. Seed 7 is fixed solely for reproducibility and has no modeled meaning. Temperature 0 with top-$p = 1$ removes sampling, and the 128-token limit matches the concise QA format.

For goal- and profile-sensitive questions, a deterministic router prioritizes memories tagged with goals, profile attributes, or response preferences. Routing is computed from query phrases only; the QA identifier is used solely to align input and output rows. Task labels, evidence identifiers, and reference answers are excluded from selection. Under GPT-4o-mini, 768 questions follow this path and 768 follow the default path. A post-routing audit may summarize benchmark types, but these labels do not change the selected route. Both branches use identical prompts and decoding settings. Because no router ablation is reported, later score differences are not attributed causally to this rule.

D.1.1 Hyperparameter Rationale and Limits

The numerical coefficients are fixed implementation heuristics, not learned cognitive parameters. The write rule places more weight on supplied importance than confidence, caps the profile bonus, and penalizes low-confidence input. The consolidation threshold of three requires repeated evidence, while summary values 0.88 and 0.90 assign high retrieval priority. Neither these coefficients nor their optimality are established by a sensitivity study. They should be treated as a reproducible configuration rather than a validated optimum.

D.2 COMPLETE LOCOMO RESULTS

Table 5: Answer quality comparison on LoCoMo. Six baseline methods are included under all three backbones. The Qwen3-14B block is reported on a common subset of 1,529 questions; traceable rows under the other backbones retain the coverage documented by their provenance records.

Backbone	Method	F1	EM	BLEU-1	ROUGE-L	METEOR
DeepSeek-V4-Flash	Vanilla RAG	43.68	13.23	38.98	42.25	31.65
	Mem0	36.26	13.28	30.77	34.23	29.56
	A-Mem	34.53	14.06	29.54	33.23	24.18
	MemoryOS	42.02	**16.87**	36.33	40.34	29.50
	MemoryBank	22.20	6.61	19.18	21.24	14.75
	MemGPT/Letta	12.12	4.09	10.78	11.84	8.61
	Melo	**49.86**	14.46	**44.32**	**48.10**	**35.92**
Qwen3-14B	Vanilla RAG	31.20	3.47	25.25	29.62	24.17
	Mem0	21.26	1.31	14.97	19.37	20.64
	A-Mem	34.26	6.02	29.04	33.12	21.82
	MemoryOS	38.00	12.30	32.64	36.38	27.00
	MemoryBank	18.24	3.01	14.58	17.26	13.26
	MemGPT/Letta	10.38	1.83	8.63	9.99	7.89
	Melo	**45.08**	**14.00**	**39.47**	**43.71**	**32.20**
GPT-4o-mini	Vanilla RAG	28.91	**3.24**	21.27	27.23	27.46
	Mem0	28.54	2.98	20.96	26.87	27.14
	A-Mem	28.41	2.79	20.79	26.75	27.09
	MemoryOS	28.27	2.72	20.57	26.59	27.12
	MemoryBank	20.48	2.59	15.21	19.36	18.85
	MemGPT/Letta	9.53	1.88	7.25	9.10	8.68
	Melo	**31.28**	2.20	**22.29**	**29.30**	**31.24**

D.3 ADDITIONAL LOCOMO INTERPRETATION

The consistent gains across four complementary overlap metrics indicate that Melo preserves more answer-bearing content across different lexical criteria. This pattern is consistent with typed memory organization and grounded context construction, which rank relevant episodic, semantic, and procedural evidence before generation. Because LoCoMo lacks a matched component ablation, these results support a framework-level association rather than assigning the gain to one module.

EM varied more across backbones. Melo led Qwen3-14B at 14.00%, but trailed MemoryOS under DeepSeek-V4-Flash and Vanilla RAG under GPT-4o-mini. In the GPT-4o-mini audit, Melo predictions averaged 12.43 normalized tokens against 4.95 reference tokens, and 307 non-exact answers still achieved F1 ≥ 0.5. Added qualifiers, alternate date orders, and reordered lists can thus preserve answer-bearing content while receiving zero EM credit. We therefore interpret EM jointly with the overlap metrics and retain F1 as the primary cross-dataset measure.

D.4 COMPLETE CMLM-ASSIST RESULTS

Table 6: Results on CMLM-Assist. The six baselines are evaluated for all three backbones on the fixed set of 1,536 test questions. Goal Alignment and Personalization are the task-specific diagnostics defined in Section C.2.

Backbone	Method	F1	EM	BLEU-1	ROUGE-L	METEOR	Goal Align.	Personal.
DeepSeek-V4-Flash	Vanilla RAG	28.63	10.35	35.83	35.78	33.50	78.77	17.24
	Mem0	5.52	1.63	13.98	12.55	8.31	2.60	6.04
	A-Mem	18.77	5.47	28.12	25.87	21.23	24.38	18.15
	MemoryOS	15.79	6.18	24.96	22.78	18.29	17.46	18.49
	MemoryBank	1.34	0.78	9.55	8.48	4.66	9.63	0.12
	MemGPT/Letta	17.02	6.84	23.64	24.17	21.84	34.22	12.03
	Melo	**38.94**	**16.86**	**46.24**	**46.07**	**42.34**	**84.55**	**37.05**
Qwen3-14B	Vanilla RAG	15.52	0.85	21.80	22.66	25.16	32.81	8.25
	Mem0	18.06	2.86	24.84	25.14	25.37	24.45	21.93
	A-Mem	17.46	3.71	25.68	24.44	22.69	16.52	18.96
	MemoryOS	12.99	**4.10**	21.24	19.94	17.26	16.81	14.27
	MemoryBank	2.24	0.26	9.81	9.35	7.11	11.48	0.28
	MemGPT/Letta	9.90	1.95	15.60	16.97	18.94	27.01	5.63
	Melo	**28.13**	4.04	**33.59**	**35.26**	**39.42**	**38.49**	**29.66**
GPT-4o-mini	Vanilla RAG	18.02	3.65	21.81	25.14	31.41	52.90	11.39
	Mem0	20.58	0.78	22.87	27.38	36.48	23.50	23.20
	A-Mem	14.34	2.86	20.11	21.37	22.88	19.77	14.06
	MemoryOS	10.40	3.26	17.22	17.45	16.73	16.80	10.86
	MemoryBank	11.84	2.02	17.67	18.91	19.74	17.83	10.57
	MemGPT/Letta	9.33	0.59	14.57	16.33	20.33	18.59	8.50
	Melo	**46.59**	**16.47**	**40.89**	**46.55**	**58.85**	**59.83**	**49.67**

D.5 ADDITIONAL CMLM-ASSIST INTERPRETATION

Melo also leads EM under DeepSeek-V4-Flash and GPT-4o-mini. Under Qwen3-14B, MemoryOS only marginally exceeds Melo, at 4.10% versus 4.04%. Non-exact Qwen3-14B Melo predictions averaged 8.62 normalized tokens against 3.82 reference tokens, and 313 still achieved F1 ≥ 0.5. These observations are consistent with sensitivity to answer length and wording. They do not exclude content errors or establish the cause of the small EM difference.

The joint gains in answer quality, Goal Alignment, and Personalization are consistent with Melo's explicit profile state and evidence-linked memory construction. The question-text router prioritizes goal and profile records without using labels or reference answers, but its independent contribution is not isolated here. The controlled ablation below provides the available component-level evidence under one matched GPT-4o-mini setting. Removing the user profile causes the largest F1 decrease and lowers Recall@20 from 93.75% to 50.00%. Removing layered orchestration lowers MRR, blocking feedback-to-memory update eliminates formed procedures, and removing consolidation reduces procedural F1 from 86.74% to 50.77%.

D.6 ADMISSION-DECISION BOUNDARY ON CMLM-ASSIST

The 1,680 CMLM-Assist noise records score 0–0.083, whereas 2,400 valid records score 0.586–0.878. Setting $\tau_w = 0.34$ near the gap midpoint rejects all noise and retains all valid records. Every threshold from 0.20 to 0.58 produces the same decisions on these supplied records. This is a score-separation check, not a downstream performance-sensitivity study. The separation reflects synthetic metadata and does not establish robust filtering of natural noisy conversations.

Retrieval uses $\tau_r = \max(0.06, \min(0.20, 0.20 s_{\max}))$, where $s_{\max}$ is the highest augmented retrieval score. The relative term adapts to each query, the 0.06 floor removes negligible matches, and the 0.20 cap prevents excessive pruning. This gate applies only to CMLM-Assist; LoCoMo retains its long-dialogue context. All values are fixed without using reference answers or final-test F1.

D.7 Detailed Matched-Ablation Controls

We evaluate these components on the complete held-out CMLM-Assist set of 1,536 questions using GPT-4o-mini. Holding the backbone fixed separates component removal from generator variation. All rows share the question identifiers, TF–IDF backend, top-$k = 20$ budget, decoding settings, concise-answer prompt, and evaluation scripts. Quality gating is inactive throughout to keep retrieval control fixed. The 20,000 paired bootstrap resamples stabilize the confidence-interval estimates without additional model calls. Table 7 supports within-protocol comparisons; its absolute F1 values are not interchangeable with the multi-baseline values in Table 6.

D.8 Complete Matched Ablation

Table 7: Controlled core-component ablation on CMLM-Assist with GPT-4o-mini. ΔF1 is the variant score minus the matched-control score; paired confidence intervals use 20,000 bootstrap samples. The final columns report the diagnostic most directly affected by each removal.

Variant	F1	ΔF1	Paired 95% CI	Target diagnostic	Control → variant
Melo (matched control)	77.00	–	–	–	–
Melo w/o Feedback-to-Memory Update	66.44	-10.56	[-12.08, -9.04]	Procedures formed	96 → 0
Melo w/o Layered Orchestration	72.14	-4.86	[-6.59, -3.15]	MRR	34.81 → 29.64
Melo w/o Consolidation	75.22	-1.78	[-3.19, -0.39]	Procedural F1	86.74 → 50.77
Melo w/o User Profile	56.28	-20.72	[-22.93, -18.56]	Recall@20	93.75 → 50.00

D.9 Additional Ablation and Cross-Setting Analysis

Removing layered orchestration lowers F1 to 72.14% and MRR from 34.81% to 29.64%, which is consistent with less effective ranking of supporting memories. Consolidation has a smaller aggregate effect, reducing F1 to 75.22%, but its impact is concentrated on the 192 procedural-memory questions, where F1 decreases from 86.74% to 50.77%. These paired changes provide component-level explanations within the controlled ablation protocol; they do not imply that one component alone explains every difference in the cross-system tables.

D.10 Cross-Setting and Memory-Behavior Analysis

Melo leads F1, BLEU-1, ROUGE-L, and METEOR under all three backbones on both benchmarks, but EM does not follow the same pattern. On LoCoMo, its F1 margins over the strongest baseline are 6.18, 7.08, and 2.37 percentage points for DeepSeek, Qwen, and GPT-4o-mini. GPT-4o-mini EM instead falls from 3.24 for Vanilla RAG to 2.20. Longer answers are consistent with this disagreement, but do not exclude omissions or incorrect qualifiers.

For runs with compatible identifiers, Melo's Memory Hit rates are 87.50%, 87.50%, and 84.38%, versus 69.21%, 43.75%, and 69.21% for Vanilla RAG. Its MRR is also higher and its store is 38.2%–46.7% smaller. These diagnostics apply only to the matched evidence protocol.

Systems without compatible evidence identifiers are excluded from this diagnostic comparison. Goal Alignment and Personalization vary by backbone and do not isolate the generator.

D.11 Failure Analysis and Additional Limitations

The matched ablation provides a narrower insight. Removing the profile lowers F1 by 20.72 points, versus 10.56 for feedback admission and 1.78 for consolidation. This ordering may reflect the profile-heavy constructed tasks and is not a universal ranking of memory mechanisms. On LoCoMo, multi-hop F1 remains 30.79%, 22.54%, and 23.35%, well below single-hop scores of 63.36%, 58.98%, and 38.30%; evidence access therefore does not ensure correct composition.

Conflict-resolution F1 remains below 8% for every backbone. Melo detects conflict tags but has no general record-level merge/split algorithm; failures arise when recency, confidence, and user confirmation favor different records or the generator fails to request clarification.

These errors bound the positive results. The ablation supports within-protocol comparisons of four components, while selective encoding, forgetting, score gating, the question router, and component interactions remain unisolated.

Profile-aware retrieval can over-personalize an answer when inferred or outdated preferences receive excessive weight. Melo caps profile bonuses and retains source and confidence metadata, but the present evaluation does not test adversarial profiles, irrelevant personalization, privacy leakage, or user-controlled deletion. Deployment efficiency is also unverified because hardware, caching, latency, cost, and index size are not normalized. These issues require longitudinal, equal-budget, and privacy-aware evaluation rather than stronger claims from the current tables.

E Implementation and Reproducibility Details

The appendix lists the intermediate states that Melo preserves for audit, reproduction, and debugging across writing, routing, consolidation, forgetting, retrieval, and generation. Table 8 summarizes these artifacts and the review risks they address.

Table 8: Memory operations and auditable artifacts in Melo.

Stage	Intermediate artifact	Audit purpose
Selective encoding	Candidate fragments, write scores, source confidence	Checks uncontrolled writing, noise, and low-confidence memory formation.
Memory-type routing	Working, episodic, semantic, and procedural labels	Checks whether similar information is routed consistently.
Consolidation	Feedback groups, evidence IDs, procedural summaries	Traces each formed workflow to the records it replaces.
Retention control	Write score, age, retention value, protection tags	Checks which records are retained or removed and why.
Retrieval and generation	Similarity, bonuses, cutoff, retrieved evidence	Separates candidate scoring, context selection, and generation errors.

A. CMLM-Assist Protocol and Leakage Controls

CMLM-Assist tests long-term goals, user preferences, profile cues, and procedural memory in a personal-assistant setting. To avoid implicit answer extraction, we use the controls in Table 9.

Table 9: CMLM-Assist protocol controls and result-admission rules.

Control	Rule
Input isolation	Each sample uses an independent memory space; memory entries, retrieval indices, and intermediate states are not shared across samples.
Answer isolation	The memory-construction stage cannot access the final question, gold answer, or diagnostic label.
Fixed test set	Formal results must cover all 1,536 QA items, with no missing, duplicated, or extra qa_id.
Empty responses	Empty responses from external services are scored as produced; manual completion, smoke outputs, and shared substitute predictions are not used in main tables.
Provenance	Each number must trace to a prediction file, metric file, and run manifest.
Key protection	Traces record provider, base URL, model name, output path, and prompt version, but not API keys.

B. Prompt Templates and Diagnostic Outputs

The anonymous review supplement contains the prompt-construction code and versioned execution paths. The memory-construction template receives only historical sessions and allowed metadata,

and extracts user-confirmed facts, preferences, corrections, goals, and workflows. The final-answer template receives only retrieved memory context and the final question, and reports unknown or insufficient evidence when support is lacking. Table 10 makes these input and output boundaries explicit.

Table 10: Prompt-template inputs, outputs, and leakage boundaries.

Template	Input	Output
Memory construction	Historical sessions, user facts, allowed metadata	Memory entries, layer labels, source evidence, confidence.
Memory retrieval	Current question, user profile, memory index	Retrieved evidence, ranking, hit status, MRR-related information.
Final answer	Final question, retrieved memory context	Answer, evidence excerpt, current memory state, brief rationale.
Evaluator	Prediction file, gold answer, task labels	F1, EM, Recall, Goal Alignment, Personalization, Storage Size, and related metrics.

C. Data, Code, and Reproducibility Artifacts

LoCoMo is reused from the public source cited in this article. The anonymous review supplement contains fixed CMLM-Assist identifiers and de-identified records, configurations, aggregate result and provenance summaries, and evaluation scripts; it excludes credentials and private conversation text. A versioned public archive with a persistent identifier will replace this package after acceptance. The static test set cannot establish the causal effect of time-dependent retention control because it neither introduces outdated memories nor advances deployment time. That mechanism requires future stress tests with preference reversals, corrected facts, stale tool outputs, retention–deletion curves, current-state accuracy, and false forgetting.