ONE CUE, THE RIGHT EPISODE: EVIDENCE RECOVERY FOR DIALOGUE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

With ever-longer interaction histories, LLM-based conversational systems face a central challenge: how can they retrieve the most discriminative evidence for a response within a limited reader-context budget? The human brain tackles this via hippocampal separation, completion from partial cues, and selective readout. Inspired by this tripartite organization, we propose **Sparse Event Addressing Memory (SEA-Mem)**. SEA-Mem first encodes each turn into immutable traces, source atoms, and sparse bindings. When a query arrives, a planner groups its information needs, and then a simple three-step mechanism resolves the memory access: **DG-Address** pinpoints the episode via sparse addressing, **CA3-Complete** recovers evidence through binding-following, and **CA1-Select** selects a budgeted set of turns that jointly supports the query. Together, they let associations guide access, while the selected original turns—not the associations—remain the evidence presented to the answerer. With lightweight association construction and compact readout, SEA-Mem achieves strong answer quality on LongMemEval and LoCoMo under a fixed source budget. On a matched LongMemEval subset, accuracy is largely preserved from Oracle evidence to full histories. Our experiments further show that SEA-Mem is more resistant to semantically confusable distractors.

1 INTRODUCTION

Long-term memory enables conversational agents to maintain continuity and personalization across sessions (Bae et al., 2022). As histories grow, answer-relevant evidence becomes distributed across turns, while recurring events and changing attributes create semantically confusable memories (Maharana et al., 2024; Su et al., 2026). A larger context window exposes more history but does not determine which sources should support an answer. Under a limited readout budget, a memory system must distinguish relevant evidence from similar alternatives and preserve complementary information spread across sources. This raises a central question: *How can a memory system retain the evidence needed for answering under interference and a limited readout budget?*

Early approaches organize long-term dialogue memory into compact or structured representations and connect related evidence for retrieval (Chhikara et al., 2025; Xu et al., 2025; Gutiérrez et al., 2025; Banerjee et al., 2026; Shu et al., 2026; Zhou & Han, 2025). Because related memories may also describe confusable events, more recent methods emphasize contextual compatibility, scene separation, and evidence-gap-driven retrieval (Yang et al., 2026; Sun et al., 2026; Zhang et al., 2025; Li et al., 2026). We address the subsequent readout bottleneck: efficiently retaining the complementary sources needed for an answer when semantically similar memories compete for a limited budget.

The hippocampus offers a functional analogy: pattern separation distinguishes overlapping experiences, while associative completion recovers related information from partial cues (Leutgeb et al., 2007; Yassa & Stark, 2011; Treves & Rolls, 1994). Dialogue memory faces both demands—identifying the right event amid similar ones and recovering evidence beyond the initial cue. This motivates our three-stage design: source addressing, associative completion, and selective readout.

We introduce **Sparse Event Addressing Memory (SEA-Mem)**, a hippocampal-inspired framework for interference-aware long-term dialogue memory. A single-pass writer preserves original

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.

turns and constructs source-grounded atoms with deterministic cross-turn associations. At retrieval time, a planner specifies the evidence needed to answer the query. **DG-Address** first distinguishes relevant source entrances from confusable memories. Because an entrance may contain only part of the answer, **CA3-Complete** follows its stored associations to recover complementary candidates, with conditional Dense retrieval providing additional raw-source candidates. The resulting pool may exceed the reader budget; **CA1-Select** therefore chooses at most k original turns that jointly support the answer. Associations guide access, while the selected original turns remain the evidence presented to the answerer. Across the evaluated benchmarks, SEA-Mem achieves the best or competitive answer accuracy while maintaining a compact Reader context, and controlled interference experiments further demonstrate stronger robustness to semantically confusable distractors than the baselines. Our contributions can be summarized as follows:

- We study long-term memory access as complementary evidence reconstruction under interference and a bounded source readout, distinguishing evidence available in the candidate pool from evidence ultimately retained for answering.
- We develop SEA-Mem, which coordinates query-guided source addressing, one-hop associative completion, and joint source selection. The pipeline restricts associative expansion to accepted entrances while preserving original dialogue turns as the evidence basis for readout.
- We evaluate SEA-Mem on LongMemEval, with supplementary results on LoCoMo. Paired Oracle–S comparisons, token-matched random and semantic history additions, and component analyses examine answer quality and evidence retention under interference. Results show better accuracy preservation than Dense retrieval under the tested history expansion and moderate event competition, while stronger event ambiguity reveals limits to this robustness.

2 Related Work

Structured long-term memory. Long-term conversational memory systems differ in compression, organization, and retrieval. Mem0 (Chhikara et al., 2025) stores compact memory items, while HEMA (Ahn, 2025) combines vector retrieval with a persistent summary. Structured methods organize memory via entities, events, or relations: A-MEM (Xu et al., 2025) links generated memory notes, HippoRAG (Gutiérrez et al., 2024; 2025) performs graph-based retrieval, REMem (Shu et al., 2026) reasons over episodic structures, and APEX-MEM (Banerjee et al., 2026) adds temporal reasoning. EMem (Zhou & Han, 2025) represents history as event-level units with dense retrieval, LLM filtering, and optional graph propagation. These methods improve organization and evidence accessibility, but structured representation alone does not determine which evidence should survive when related memories compete for limited context.

Associative and interference-aware memory access. Recent work targets failures of similarity-based retrieval in long, semantically dense histories. AssoMem (Zhang et al., 2025) builds an associative graph over dialogue utterances and extracted clues, ranking memories by relevance, importance, and time. STITCH (Yang et al., 2026) indexes memories with contextual intent to suppress semantically similar but context-incompatible records. CreaMem (Sun et al., 2026) partitions memory by life scene to reduce cross-scene interference, maintaining complementary representations within each scene. EviMem (Li et al., 2026) instead treats incomplete retrieval as an evidence-gap problem, iteratively refining retrieval toward missing support. T-Mem (Guo et al., 2026) further explores associative reachability beyond direct query–memory similarity via write-time retrieval triggers.

SEA-Mem is complementary but targets a different bottleneck: under a fixed final source-readout budget, individually relevant memories do not guarantee that the evidence needed to answer survives final selection. SEA-Mem thus separates source addressing, local associative completion, and joint source readout, while presenting the original dialogue turns as evidence to the answerer. This lets us distinguish evidence recovered into the candidate pool from evidence retained under interference and budget competition.

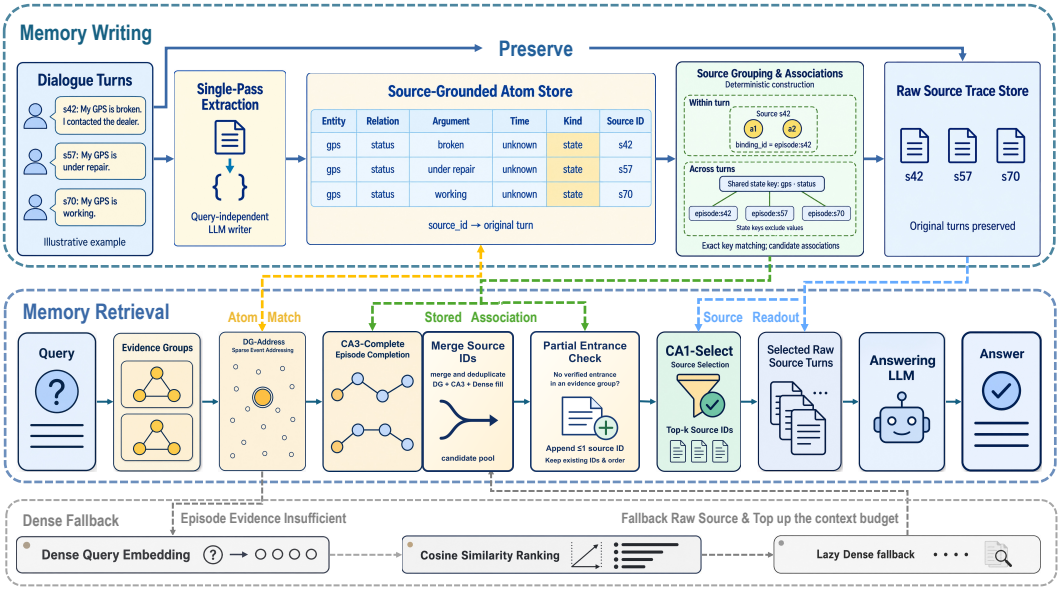


Figure 1: **Overview of SEA-Mem.** During query-independent memory writing, SEA-Mem preserves original dialogue turns and extracts source-grounded atoms in a single pass. Atoms from the same turn are grouped by provenance, and deterministic keys create candidate cross-turn associations. At retrieval time, the query is decomposed into evidence groups: DG-Address identifies atom entrances, CA3-Complete expands their stored associations, and the resulting source IDs are merged and deduplicated with conditional Dense retrieval. A partial-entrance check may append at most one extra source if a group lacks a verified entrance. Finally, CA1-Select chooses up to k original turns for the answering LLM.

Hippocampal inspiration for memory systems. Computational accounts associate hippocampal pattern separation with distinguishing overlapping experiences and pattern completion with recovering associated information from partial cues (Marr, 1971; Treves & Rolls, 1994; Rolls, 2013). These principles have inspired systems such as HippoRAG (Gutiérrez et al., 2024) and HEMA (Ahn, 2025). SEA-Mem uses the analogy only to motivate its functional decomposition into addressing, associative completion, and selective source readout, rather than as a claim of neural implementation.

3 Method

3.1 Overview

SEA-Mem enables evidence-efficient access to long dialogue histories under a limited reader budget. It balances retaining complete records for faithful generation with activating only a compact evidence subset per query. It stores persistent raw turns and sparse event-level indices; at retrieval, only a bounded subset of recovered sources is passed to the LLM (see Figure 1). A worked example is provided in Appendix A.1.

3.2 Memory Writing

SEA-Mem constructs query-independent access paths by extracting source-grounded atoms, grouping them by provenance, and indexing cross-turn associations, while preserving the original dialogue for subsequent readout.

Atom Extraction. We extract compact, source-grounded atoms to make individual facts addressable without replacing the original dialogue. PropBank represents propositions through predicates and their semantic arguments (Palmer et al., 2005), while TimeML represents event expressions and their temporal anchoring (Pustejovsky et al., 2003). Drawing on these representations, SEA-Mem

decomposes each dialogue turn into atoms of the form

$$a = (e, r, u, \tau, c; s), \tag{1}$$

where e, r, and u denote the entity, relation, and argument; τ records the explicitly stated time or is marked unknown when unavailable; $c \in \{\text{event}, \text{state}\}$ distinguishes events from states; and s is the identifier of the original source turn.

The semantic fields are extracted in a single, query-independent extraction stage, with the source identifier attached deterministically. For state atoms, the relation denotes an attribute and the argument its stated value. Original turns are retained, including those without extracted atoms.

Source Grouping. Atoms extracted from the same dialogue turn form a *source group* and share a pointer to that turn. Group membership is determined by provenance rather than semantic similarity: distinct turns remain separate even when they describe related events or states.

Cross-Turn Associations. Prior work examines structured context in cross-document event identity (Pratapa et al., 2021). Motivated by this use of structured descriptions, we derive typed keys from atom fields to index candidate associations between source groups:

$$\begin{aligned} \kappa_{\text{evt}}(a) &= \nu(e, r, u), \\ \kappa_{\text{attr}}(a) &= \nu(e, r) \qquad \text{if } c = \text{state}, \end{aligned} \tag{2}$$

where ν denotes deterministic text normalization. Keys are formed only when their required components are available. The event key applies to both event and state atoms; the state-attribute key provides an additional association for states. Distinct source groups sharing an exact normalized key of the same type are linked bidirectionally, with duplicate links removed. Link construction requires no additional LLM pass.

The state-attribute key excludes u, allowing reports of different values of the same attribute to remain connected; neither key includes τ or s, although both are retained in the atoms. Shared keys therefore indicate candidate evidence associations, not verified episode identity or temporal succession. Source mappings and associations are reused across queries: the write-time index provides candidate access paths, while query-time retrieval determines which sources initiate associative expansion and which candidate turns are selected for readout. Atoms guide access, whereas original sources retain contextual details that may be omitted during extraction.

3.3 Memory Retrieval

Long dialogue histories require two coupled retrieval abilities: *event discrimination*, distinguishing query-relevant events from confusable alternatives, and *evidence recovery*, assembling complementary evidence scattered across turns. Neither alone suffices; for example, answering where a user stayed during a particular trip and why they changed hotels requires both distinguishing that trip from similar visits and combining information from multiple turns. This decomposition parallels two complementary computations associated with hippocampal memory: pattern separation reduces interference among similar experiences, whereas associative completion recovers related information from partial cues (Leutgeb et al., 2007; Yassa & Stark, 2011; Treves & Rolls, 1994). This analogy motivated the corresponding components of our retrieval architecture.

Following this principle, SEA-Mem organizes memory access into source addressing, associative completion, and budgeted selection. A planner first organizes the query into evidence groups. DG-Address then uses each group's cues to admit source-grounded entrances; when fewer than k distinct entrance sources are available, full-query Dense retrieval supplements them to form a base pool. CA3-Complete adds sources reached through stored associations from the DG entrances without replacing base-pool sources, after which at most one unused partial DG-Address source is appended for the first eligible evidence group lacking a verified source. Finally, CA1-Select evaluates the candidate sources jointly against the query and its evidence plan, selecting at most k original turns for answer generation.

The hippocampal analogy guides functional organization: associative completion uses stored links rather than recurrent attractor dynamics. Together with source-grounded writing, this design recovers complementary evidence amid interference while keeping the final readout compact and traceable to the original dialogue.

Query Evidence Grouping. Prior work has shown that question decomposition helps structure complex information needs and support multi-hop reasoning (Wolfson et al., 2020; Fu et al., 2021). Motivated by this, our planner organizes a query into a declarative retrieval plan in two steps.

First, it decomposes the information needed to answer a query q into an ordered sequence of evidence groups, $\mathcal{G}(q) = (g_1, \ldots, g_m)$. Each group is represented as $g_i = (E_i, P_i, T_i, L_i)$, where the four components are sets of entity, predicate, event-type, and discriminative lexical cues, respectively. The index $i \in \{1, \ldots, m\}$ follows the planner's group order. DG-Address uses these cues to locate candidate stored atoms and their original source turns.

Second, the planner records query-level requirements: the expected number of answers and any selection, comparison, or ordering intent. These requirements guide need-aware entrance expansion and CA1-Select's final choice of source turns. Thus, the evidence groups specify what to retrieve, while the query-level requirements specify how the retrieved evidence contributes to the answer.

Throughout retrieval, s denotes an original source-turn identifier. Entrance, candidate, and selected-source sets contain source identifiers, not atoms or turn text. Their associated atoms and original turns are accessed through the stored source mappings. For a fixed query q, we leave the query dependence of group-level quantities implicit.

DG-Address: Need-Aware Source Addressing. DG-Address uses the planner's evidence groups to locate source entrances for associative completion. For each group, it first evaluates cue support from stored sources, then admits atom-backed entrances from the resulting support tiers, and when needed supplements them with semantic entrances.

Cue-based source verification. Let $b_i \in \{0, 1\}$ indicate whether E_i specifies a concrete entity, excluding generic entity terms. In the main configuration, the binary variables $e_{is}, p_{is}, \ell_{is}$ indicate whether source s matches an entity, predicate, or lexical cue, respectively. We define $n_{is} = p_{is} + \ell_{is}$ as the number of matched non-entity cue types and assign each source the following support tier:

$$\mathrm{tier}(g_i, s) = \begin{cases} \text{verified}, & (b_i = 1 \wedge e_{is} = 1 \wedge n_{is} \geq 1) \ \vee \ (b_i = 0 \wedge n_{is} \geq 2), \\ \text{partial}, & e_{is} + n_{is} \geq 1, \\ \text{unmatched}, & \text{otherwise.} \end{cases} \tag{3}$$

The cases are evaluated in order, so a verified source is not also classified as partial. For each evidence group, DG-Address admits distinct verified sources as entrances when available; otherwise, it admits the highest-ranked partial source. Unmatched sources are not admitted by the native cue-based stage. Sources included only through Dense retrieval remain readout candidates, not associative entrances.

Need-aware semantic supplementation. Cue matching can miss relevant sources when stored atoms express a relation or argument differently from the query. For eligible requests requiring evidence from multiple memories, DG-Address therefore considers additional atom-backed sources using semantic similarity and checks their compatibility with the group's constraints. Admitted sources form a supplemental entrance set S_i^+.

The continuation decision depends on the query plan and the native entrances already found. For eligible synthesis requests with unspecified answer cardinality, semantic supplementation proceeds when fewer than two distinct native sources have been admitted and stops otherwise. Other eligible set-valued requests follow their multi-entrance policy. Denoting the native entrance set by N_i, DG-Address returns

$$D_i = N_i \cup S_i^+, \tag{4}$$

where $S_i^+ = \varnothing$ when supplementation is not activated. Semantic supplementation is limited by the configured entrance budget without removing native entrances. For each evidence group, D_i provides the source-grounded entrances from which CA3-Complete follows stored associations to recover additional candidate sources.

CA3-Complete: Associative Evidence Completion. DG-Address identifies sources that directly support the query's evidence requirements. CA3-Complete extends these entrances to source turns connected by the associations established at writing time. For each evidence group, it traverses only direct neighbors of its admitted entrances and maps the reached atoms back to their original source

turns. One-hop traversal limits the propagation of indirect matches while allowing the number of recovered candidates to vary with the stored associations.

Let $D_q = \bigcup_{i=1}^{m} D_i$ denote the set of source identifiers admitted as DG-Address entrances. We first build a base pool from these entrances, using full-query Dense retrieval to supplement the pool toward k distinct sources when needed. Let $F_k(q)$ be the set of identifiers of the highest-ranked original turns not already in D_q, selected until the base pool reaches k sources or no further source is available. We set $F_k(q) = \varnothing$ when $|D_q| \geq k$. The base pool is

$$B_q = D_q \cup F_k(q). \tag{5}$$

Dense fill preserves access to raw evidence when structured addressing supplies few sources; its sources are readout candidates, not entrances for associative traversal.

We next add evidence reached from the DG entrances. Let R_q be the set of distinct source identifiers recovered through their stored one-hop associations. The association-augmented pool is

$$A_q = B_q \cup R_q. \tag{6}$$

This union is non-replacing: associated sources can enlarge the pool but cannot displace sources already in B_q. Traversal starts only from D_q, not from the Dense sources in $F_k(q)$.

Finally, we consider whether a group without verified support can contribute one unused partial source at readout. For each group g_i, let V_i and U_i be the sets of source identifiers classified as verified and partial, respectively, by equation 3. The indices of eligible groups are

$$\mathcal{H}_q = \{\, i \in \{1, \ldots, m\} : V_i = \varnothing,\ U_i \setminus A_q \neq \varnothing \,\}. \tag{7}$$

The first condition identifies a group without a verified source; the second requires a partial source not already in A_q. When $\mathcal{H}_q \neq \varnothing$, let $i_q^\star = \min \mathcal{H}_q$ be the first eligible group index in planner order, and let $s_q^\star$ be the highest-ranked unused partial source in $U_{i_q^\star} \setminus A_q$. We admit at most this one additional source:

$$P_q = \begin{cases} \{s_q^\star\}, & \mathcal{H}_q \neq \varnothing, \\ \varnothing, & \mathcal{H}_q = \varnothing, \end{cases} \qquad C_q = A_q \cup P_q. \tag{8}$$

The source in P_q does not initiate further traversal. All unions deduplicate by original source ID while retaining distinct turns as separate candidates. We use C_q for the resulting source-ID set and $\mathbf{C}_q$ for its ordered, deduplicated candidate list. Set unions specify membership; retrieval order is retained separately in $\mathbf{C}_q$. CA1-Select then chooses at most k original source identifiers from C_q, using $\mathbf{C}_q$ as its ordered input.

CA1-Select: Budgeted Complementary Source Readout. DG-Address, associative completion, and Dense fill favor evidence coverage, so the candidate pool may exceed the answerer's budget and contain redundant or merely topical sources. CA1-Select resolves this coverage–budget tension by selecting a compact combination of original turns that jointly supports the query's evidence requirements. It receives the query, its evidence groups and intent, and the ordered candidate list $\mathbf{C}_q$ paired with the original turn text for each source. Source text is presented without summarization and truncated to a configured limit; in the current setting, each view contains at most the first 1,600 characters of its turn. An LLM evaluates the sources jointly and proposes the smallest sufficient combination, favoring direct support and complementary facts over redundant or merely topical material. Its task is to select source IDs, not to compose the answer.

The proposed IDs are then checked against the candidate pool, deduplicated, and bounded by k. For queries that combine independent evidence or require comparison or temporal ordering, a coverage rule retains designated DG/CA3 core sources when the budget permits. If the proposal contains no valid ID, the order in $\mathbf{C}_q$ provides a deterministic recovery selection. Denoting the resulting selection procedure by Select_k, we obtain

$$I_q = \mathrm{Select}_k(q, \mathbf{C}_q), \qquad I_q \subseteq C_q, \qquad |I_q| \leq k. \tag{9}$$

The selected IDs are dereferenced to original turns for answer generation. Thus, k bounds the answerer's final source readout and also sets the Dense-fill target for the base pool. It does not cap the association-augmented candidate pool: CA1 may inspect more than k sources and incurs its own selection cost.

4 EXPERIMENTS

4.1 DATASET AND EVALUATION

We evaluate on the paired LongMemEval-Oracle (LME-O) and full-history LongMemEval-S (LME-S) settings (Wu et al., 2024) to measure robustness to history expansion while holding questions and target evidence fixed. We additionally evaluate on the 1,540 answerable questions from LoCoMo Categories 1–4 (Maharana et al., 2024) to assess generalization to long, multi-session dialogues. We use DeepSeek-V4-Flash (DeepSeek-AI et al., 2026) as the primary memory and answering backbone, with DeepSeek-V4-Pro as the independent Judge across all methods. We additionally repeat the evaluation with Qwen3.7-Plus (Yang et al., 2025) to assess backbone generalization, while keeping the Judge fixed.

We compare SEA-Mem with Dense retrieval as an embedding-based baseline, Mem0 as a fact-centric memory system, A-MEM as an agentic associative memory, HippoRAG-2 as a graph-based retrieval system, REMem as an episodic memory baseline, and AssoMem as an interference-aware associative memory system. Each method retains its native memory pipeline. We evaluate with DeepSeek-V4-Flash and Qwen3.7-Plus, using judged accuracy J and selected-source evidence recall $\mathrm{ER}_{\mathrm{sel}}$ as the primary metrics. Full protocols and additional results are provided in Appendix B.1.

SEA-Mem achieves a higher average Judge correctness on both LoCoMo and LongMemEval, demonstrating robust transfer to longer multi-session dialogues.

4.2 ROBUSTNESS TO MEMORY INTERFERENCE

Memory Interference. The gap between LongMemEval-O and LongMemEval-S suggests retrieval interference from expanded history but cannot isolate its cause. To isolate retrieval interference, we construct four history conditions for 150 LongMemEval-S questions: evidence-only, evidence plus random distractors, evidence plus semantically similar distractors, and full history. The similar condition adds roughly 50% of the available non-evidence tokens, while the random condition matches its session count and token volume. We freeze Writer outputs and rerun retrieval, source selection, and answer generation with $k = 5$. We report judged accuracy J, selected-source evidence recall $\mathrm{ER}_{\mathrm{sel}}$, and distractor occupancy (DO), the proportion of selected sources drawn from added non-evidence sessions. See Appendix B.2.

Table 2 reports the controlled results. Under Full-S, SEA-Mem attains higher judged accuracy and a smaller drop from Evidence-only than the baselines. Under Semantic-50, its aggregate accuracy remains unchanged, with comparable numbers of questions flipping in each direction. SEA-Mem also has lower distractor occupancy. These results suggest that SEA-Mem maintains a cleaner bounded Reader context.

Readout-Budget Sensitivity. As shown in Figure 2(a), we vary $k \in \{2, 5, 10\}$ to test robustness under interference. We focus on Semantic-50 because it adds query-relevant non-evidence sessions while retaining all evidence and matching Random-50 in history volume, thereby creating direct competition for the limited readout budget. For SEA-Mem, k caps the raw source turns that CA1-Select may pass to the Reader (it may select fewer); for Dense-native and AssoMem-native, we truncate their rankings to the top k. Thus all methods share the same maximum final readout budget, though they may select fewer sources or use different intermediate candidates. Results for all history conditions are in Appendix B.2.

4.3 EFFICIENCY EVALUATION

We evaluate efficiency on the same fixed 60-question LongMemEval-S cohort. Each method uses its native memory writing and retrieval procedure, while questions, histories, maximum readout budget ($k = 5$), and Reader configuration are held fixed. For each question q, we measure the cold-start lifecycle workload as

$$T_{\mathrm{life}}(q) = T_{\mathrm{write}}(q) + T_{\mathrm{online}}(q), \tag{10}$$

where T_{write} covers memory construction and T_{online} covers retrieval, source selection, and answer generation. Both include LLM input/output tokens and embedding input tokens, measured with a shared tokenizer.

Table 1: **Robustness to history expansion and cross-dataset generalization.** Panel (a) reports aggregate judged accuracy J, selected-source evidence recall $\mathrm{ER}_{\mathrm{sel}}$, and LoCoMo official F1. For LongMemEval, $\Delta J = J_{\mathrm{S}} - J_{\mathrm{O}}$, with values closer to zero indicating better preservation under history expansion. Panel (b) reports category-level LongMemEval accuracy as paired $J_{\mathrm{O}}/J_{\mathrm{S}}$ results. $\mathrm{ER}_{\mathrm{sel}}$ is omitted for REMem because its gist and fact nodes lack exact turn-level provenance, and for A-MEM because its native expansion is not constrained to a source-matched $k = 5$ readout. Deltas and paired results are reported only for matched protocols. All values are percentages.

(a) Aggregate results

Method	Answer Accuracy			Evidence Recall			LoCoMo		
	J_{O}	J_{S}	ΔJ	ER_{O}	ER_{S}	$\Delta\mathrm{ER}$	J	F1	$\mathrm{ER}_{\mathrm{sel}}$
DeepSeek-V4-Flash backbone									
SEA-Mem	**79.4**	**78.0**	**-1.4**	80.8	**81.2**	**+0.4**	**58.5**	**49.3**	**73.1**
AssoMem	76.0	69.0	-7.0	**82.3**	76.2	-6.1	23.0	21.2	22.8
REMem	74.4	69.2	-5.2	–	–	–	48.1	47.4	–
HippoRAG-2	71.0	68.2	-2.8	76.8	74.5	-2.3	47.8	40.0	58.8
A-MEM	69.0	66.0	-3.0	–	–	–	47.2	39.0	–
Mem0	69.2	67.6	-1.6	78.2	76.1	-2.1	51.9	42.0	62.9
Dense	70.0	64.0	-6.0	76.0	69.1	-6.9	46.2	39.0	55.6
Qwen3.7-Plus backbone									
SEA-Mem	**78.2**	**77.0**	**-1.2**	79.2	78.9	**-0.3**	**56.1**	46.9	**72.1**
AssoMem	71.2	66.8	-4.4	80.1	74.3	-5.8	20.9	21.3	22.5
REMem	73.0	68.6	-4.4	–	–	–	48.8	**48.8**	–
HippoRAG-2	70.4	67.2	-3.2	78.2	74.5	-3.7	42.1	38.6	62.4
A-MEM	67.8	63.6	-4.2	–	–	–	45.6	38.4	–
Mem0	68.6	63.2	-5.4	**86.9**	**81.1**	-5.8	50.9	31.9	69.2
Dense	68.4	62.0	-6.4	76.3	69.4	-6.9	50.0	27.8	55.6

(b) Category-level robustness on LongMemEval

Method	Judged Accuracy: LongMemEval-O / LongMemEval-S					
	Knowledge Update	Single-session User	Single-session Preference	Single-session Assistant	Temporal Reasoning	Multi-session
DeepSeek-V4-Flash backbone						
SEA-Mem	86.4 / **91.3**	**97.1** / 90.5	**83.3** / 70.0	89.3 / 94.1	**82.0** / **75.0**	**68.4** / **60.0**
AssoMem	80.8 / 74.4	94.3 / 84.3	**83.3** / 70.0	**94.6** / **96.4**	75.9 / 63.2	54.1 / 51.9
REMem	**89.7** / 78.3	92.9 / **91.4**	80.0 / **96.7**	58.9 / 76.5	75.9 / 57.5	59.4 / 55.0
HippoRAG-2	74.4 / 78.2	82.9 / 82.9	73.3 / 73.3	75.0 / 94.6	69.9 / 65.4	50.4 / 45.1
A-MEM	75.6 / 70.5	94.3 / 85.7	70.0 / 40.0	91.1 / 96.4	52.6 / 63.9	58.6 / 48.9
Mem0	83.3 / 85.6	78.6 / **91.4**	76.7 / 70.0	64.3 / 69.6	77.4 / 61.2	63.9 / 46.6
Dense	73.1 / 67.9	94.3 / 90.0	76.7 / 86.7	89.3 / **96.4**	66.2 / 53.4	49.6 / 39.8
Qwen3.7-Plus backbone						
SEA-Mem	80.8 / 84.6	92.9 / **90.0**	66.7 / 60.0	**92.9** / 83.9	76.7 / 73.7	**65.4** / **67.7**
AssoMem	76.9 / 82.6	91.4 / 76.2	**80.0** / **88.9**	**92.9** / **96.4**	74.4 / 57.5	54.9 / 45.0
REMem	**88.5** / 79.5	92.9 / **90.0**	76.7 / **93.9**	55.4 / 78.6	73.7 / 55.6	59.4 / 54.1
HippoRAG-2	75.6 / 79.5	94.3 / 81.4	70.0 / 60.0	89.3 / 91.0	**77.4** / **81.4**	63.2 / 48.1
A-MEM	71.2 / 69.6	95.7 / 81.0	70.0 / 33.3	91.1 / 96.4	52.6 / 62.4	58.6 / 45.1
Mem0	82.6 / **91.3**	81.0 / 81.0	77.8 / 66.7	64.7 / 58.8	72.5 / 57.5	50.0 / 45.0
Dense	71.8 / 70.5	**95.7** / **90.0**	66.7 / 66.7	91.1 / 92.9	63.2 / 48.1	48.1 / 42.1

Table 2: **Controlled history interference at $k = 5$.** All methods use the same Reader and Judge. J and distractor occupancy (DO) use 150 questions; ΔJ is relative to Evidence-only for each method. Values are percentages. Best values within each history are bold.

	J			ER			DO			ΔJ		
History	SEA	Asso	Dense	SEA	Asso	Dense	SEA	Asso	Dense	SEA	Asso	Dense
Evidence-only	**76.0**	69.3	64.0	81.4	**82.9**	78.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Random-50	**72.7**	64.0	59.3	**79.5**	79.1	71.9	**4.0**	13.1	14.7	**-3.3**	-5.3	-4.7
Semantic-50	**76.0**	60.7	58.0	78.0	**78.6**	70.2	**5.1**	17.3	19.9	**0.0**	-8.7	-6.0
Full-S	**73.3**	62.0	58.0	**79.0**	78.2	70.2	**5.1**	17.5	19.9	**-2.7**	-7.3	-6.0

Figure 2(b) shows that SEA-Mem achieves the smallest Reader context among the evaluated methods, yielding the most compact evidence readout. Its lifecycle workload is also lower than several structured memory baselines, including HippoRAG-2, REMem, Mem0, and A-MEM. Thus, SEA-

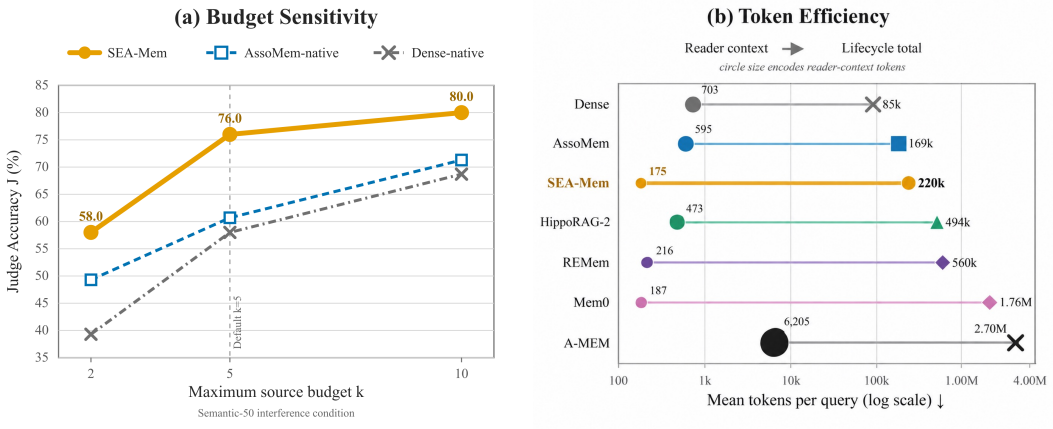


Figure 2: **Readout-budget sensitivity and token efficiency. (a)** Judge accuracy on Semantic-50 for source budgets $k \in \{2, 5, 10\}$; dashed line: $k = 5$. **(b)** Token efficiency on LongMemEval-S60 at $k = 5$. Each row links Reader-context tokens (left circle) to lifecycle workload (right marker) on a log scale; circle size is log-scaled by Reader-context tokens. Lifecycle includes LLM/embedding tokens from cold writing and online querying, excluding judging. Lower is better.

Mem's efficiency advantage lies primarily in compact evidence readout rather than in minimizing memory-construction cost.

4.4 Component Ablation

We isolate the contributions of DG-Address, CA3-Complete, and CA1-Select on 150 LongMemEval-S questions. All variants reuse the same frozen Writer memories and query plans and use the same reasoning-enabled Reader, independent Judge, and maximum readout budget of $k = 5$. Each variant reconstructs its complete access path rather than modifying only the final candidate list. Further protocol details are provided in Appendix B.4.

Table 3 shows that removing DG causes the largest degradation, reflecting the loss of both verified entrance matching and DG-dependent partial-source recovery. Removing CA3-Complete reduces accuracy by 4.7 points, showing that CA3-Complete associations contribute beyond initial addressing. Removing CA1-Select reduces both accuracy and evidence recall, indicating that selecting a compact sufficient evidence set is more effective than passing the first five candidates directly to the Reader.

Table 3: **Component ablation on LongMemEval-S.** All variants use the same 150 questions. Scores are percentages.

Variant	J	$\mathrm{ER}_{\mathrm{sel}}$	ΔJ
Full	**78.7**	**80.3**	0.0
w/o DG	54.0	57.5	-24.7
w/o CA3	74.0	77.4	-4.7
w/o CA1	71.3	76.6	-7.3

5 Conclusions and Limitations

SEA-Mem combines source-grounded memory atoms, structured addressing, and bounded source selection to preserve answer-relevant evidence under expanding dialogue histories. Experiments on LongMemEval and LoCoMo demonstrate improved robustness to interference and generalization to long multi-session dialogues, particularly under limited readout budgets. Its advantage becomes less pronounced as the Reader budget increases, suggesting that SEA-Mem primarily benefits settings where evidence capacity is limited. Future work could integrate SEA-Mem with agent planning, tool use, and reflection as a grounded long-term memory layer for continuously acting agents.

REFERENCES

Kwangseob Ahn. HEMA: A hippocampus-inspired extended memory architecture for long-context AI conversations, 2025.

Sanghwan Bae, Donghyun Kwak, Soyoung Kang, Min Young Lee, Sungdong Kim, Yuin Jeong, Hyeri Kim, Sang-Woo Lee, Woomyoung Park, and Nako Sung. Keep me updated! memory management in long-term conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022.

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. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16470–16489, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.749. URL https://aclanthology.org/2026.acl-long.749.pdf.

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

DeepSeek-AI et al. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. Decomposing complex questions makes multi-hop qa easier and more interpretable. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021. doi: 10.48550/arXiv.2110.13472.

Weidong Guo, Dakai Wang, Zixuan Wang, Hui Liu, and Yu Xu. T-mem: Memory that anticipates, not archives. *arXiv preprint arXiv:2606.15405*, 2026.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.48550/arXiv.2405.14831.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. doi: 10.48550/arXiv.2502.14802.

Jill K. Leutgeb, Stefan Leutgeb, May-Britt Moser, and Edvard I. Moser. Pattern separation in the dentate gyrus and ca3 of the hippocampus. *Science*, 315(5814):961–966, February 2007. doi: 10.1126/science.1135801. PMID: 17303747.

Yuyang Li, Yime He, Zeyu Zhang, and Dong Gong. EviMem: Evidence-gap-driven iterative retrieval for long-term conversational memory. *arXiv preprint arXiv:2604.27695*, 2026. doi: 10.48550/arXiv.2604.27695. URL https://arxiv.org/abs/2604.27695.

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 (Volume 1: Long Papers)*, pp. 13851–13870, 2024. doi: 10.18653/v1/2024.acl-long.747.

David Marr. Simple memory: a theory for archicortex. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 262(841):23–81, 1971.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005. doi: 10.1162/0891201053630264.

Adithya Pratapa, Zhengzhong Liu, Kimihiro Hasegawa, Linwei Li, Yukari Yamakawa, Shikun Zhang, and Teruko Mitamura. Cross-document event identity via dense annotation. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 496–517, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.39. URL https://aclanthology.org/2021.conll-1.39/.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. Timeml: Robust specification of event and temporal expressions in text. In *AAAI Technical Report SS-03-07*. AAAI Press, 2003.

Edmund T Rolls. The mechanisms for pattern completion and pattern separation in the hippocampus. *Frontiers in Systems Neuroscience*, 7:74, 2013.

Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. Remem: Reasoning with episodic memory in language agent. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Miao Su, Yucan Guo, Zhongni Hou, Long Bai, Zixuan Li, Yufei Zhang, Guojun Yin, Wei Lin, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. Beyond dialogue time: Temporal semantic memory for personalized LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 29935–29951, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1496. URL https://aclanthology.org/2026.findings-acl.1496.pdf.

Qixuan Sun, Yue Que, Bowei He, Jin Guo, Dihang Yang, Wenchang Situ, and Chen Ma. CreaMem: A scene-aware memory architecture for personalized agents. *arXiv preprint arXiv:2609.08550*, 2026. URL https://arxiv.org/abs/2609.08550.

Alessandro Treves and Edmund T Rolls. Computational analysis of the role of the hippocampus in memory. *Hippocampus*, 4(3):374–391, 1994.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark, 2020.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.48550/arXiv.2502.12110.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ruozhen Yang, Yucheng Jiang, Yueqi Jiang, Priyanka Kargupta, Yunyi Zhang, and Jiawei Han. Grounding agent memory in contextual intent. *arXiv preprint arXiv:2601.10702*, 2026. doi: 10.48550/arXiv.2601.10702. URL https://arxiv.org/abs/2601.10702.

Michael A. Yassa and Craig E. L. Stark. Pattern separation in the hippocampus. *Trends in Neurosciences*, 34(10):515–525, 2011. doi: 10.1016/j.tins.2011.06.006.

Kai Zhang, Xinyuan Zhang, Ejaz Ahmed, Hongda Jiang, Caleb Kumar, Kai Sun, Zhaojiang Lin, Sanat Sharma, Shereen Oraby, Aaron Colak, Ahmed Aly, Anuj Kumar, Xiaozhong Liu, and Xin Luna Dong. Assomem: Scalable memory qa with multi-signal associative retrieval. *arXiv preprint arXiv:2510.10397*, 2025.

Sizhe Zhou and Jiawei Han. A simple yet strong baseline for long-term conversational memory of LLM agents. *arXiv preprint arXiv:2511.17208*, 2025. doi: 10.48550/arXiv.2511.17208. URL https://arxiv.org/abs/2511.17208.

APPENDIX

A METHOD DETAILS

A.1 WORKED EXAMPLE OF MEMORY RETRIEVAL

Example. LongMemEval-S `dfde3500`, with source budget $k = 5$. Source IDs omit the example prefix, source text is excerpted, and only representative atoms are shown.

Query and evidence plan.

> *What day of the week did I meet with my previous language exchange tutor Juan?*

The query planner produces one evidence group:

- Entity cue: `juan`.
- Relation cues: `language_exchange_tutor` and `met_with`.
- Event type: `meeting`.
- Lexical terms: `previous`, `language_exchange`, and `tutor`.
- Intent: `select`; selection: `last`; return role: `event_value`.
- Temporal constraint: `temporal_order(previous)`.

Stored sources and atoms. The following excerpts illustrate the source-grounded memory used in this retrieval trace. Each atom retains a pointer to its original source.

- **`s13:t0`**
  Source: "Can you remind me what day I have my language exchange class with Juan?"
  Representative atom:

  ```
  (user, has_language_exchange_class_with, Juan).
  ```

- **`s13:t8`**
  Source: "I had a language exchange class at a local language school, where I'm paired with a Colombian tutor named Juan. We meet every Wednesday evening ..."
  Representative atoms:

  ```
  (user, is_paired_with, Juan),
  (juan, is_tutor, Colombian),
  (user, meets_with, Juan),
  (user, meets_every_wednesday_evening, Juan).
  ```

- **`s13:t6`**
  Source: "Juan and I always meet in the evening, and I usually have dinner plans with friends on Thursdays. I'm pretty sure it's not Thursday."
  Representative atoms:

  ```
  (user, meets_with, Juan),
  (user, has_dinner_plans_with_friends_on, Thursday).
  ```

Retrieval trace. The atoms remain unchanged during retrieval; only the set of source candidates evolves. Native DG-Address matching identifies two entrances:

$$N_1 = \{\texttt{s13:t0}, \texttt{s13:t8}\}.$$

Because the query is not set-valued, conditional semantic supplementation is not activated:

$$S_1^+ = \varnothing, \qquad D_q = N_1.$$

Full-query Dense fill adds three raw-source candidates:

$$F_k(q) = \{\texttt{s13:t1}, \texttt{s13:t5}, \texttt{s13:t2}\}.$$

The stored association

$$\texttt{s13:t8} \leftrightarrow \texttt{s13:t6}$$

allows CA3-Complete to recover one additional source:

$$R_q = \{\texttt{s13:t6}\}.$$

Under the non-replacing union policy, Dense fill completes the five-source base pool and the CA3-recovered source is appended without displacing an existing candidate. After deduplication by source ID, the CA1 candidate pool is

$$\begin{aligned} C_q &= D_q \cup F_k(q) \cup R_q \\ &= \{\texttt{s13:t0}, \texttt{s13:t8}, \texttt{s13:t1}, \texttt{s13:t5}, \texttt{s13:t2}, \texttt{s13:t6}\}, \qquad |C_q| = 6. \end{aligned}$$

Selected evidence and answer. CA1-Select returns

$$I_q = \{\texttt{s13:t8}, \texttt{s13:t0}\}, \qquad |I_q| = 2 \le k.$$

The LLM selector chooses `s13:t8`, which explicitly states that the meeting occurred on Wednesday evening. The core-coverage check additionally retains the native entrance `s13:t0`. The CA3-recovered source `s13:t6` remains in the candidate pool but is not selected for final readout.

The Reader produces

> *Wednesday evening.*

The reference answer is *Wednesday*, and the Judge marks the response as correct.

The benchmark annotates `s13:t8` and `s23:t2` as gold sources. Only the former appears in the selected set, giving selected-source evidence recall 0.5. Thus, this example illustrates that the retrieved source can directly support a correct answer even when the complete annotated evidence set is not recovered.

A.2 Memory Writing Implementation Details

This section describes the single-pass Writer, its deterministic post-processing, and the construction of source groups and cross-turn associations.

Atom Extraction. The Writer processes batches of dialogue turns without access to downstream queries or reference answers. Each input turn includes a source identifier, speaker, and original text. The LLM extracts atoms in one stage; *single-pass* does not mean that an entire history fits in one API call. Each atom has the five semantic fields defined in equation 1. The source identifier is supplied with the input and retained as the atom's provenance.

The JSON fields `entity`, `relation`, `argument`, `time`, and `kind` correspond to e, r, u, τ, c, respectively. The field `turn_id` supplies the original source-turn identifier s, giving the internal atom representation $a = (e, r, u, \tau, c; s)$.

The Writer returns:

```
{
  "turns": [
    {
      "turn_id": "<source identifier>",
      "facts": [
        {
          "entity": "<subject>",
          "relation": "<action or attribute>",
          "argument": "<object or value>",
          "time": "<explicit time or unknown>",
          "kind": "<event or state>"
```

```
        }
      ]
    }
  ]
}
```

The system prompt for this configuration is shown below, with line wrapping for readability:

```
Extract durable, source-grounded memories from EACH supplied
dialogue turn. Return JSON only as
{turns:[{turn_id,facts:[
{entity,relation,argument,time,kind}]}]}.

Return every supplied turn_id exactly once and use facts=[]
when a turn has no durable fact. Do not impose a numeric
fact limit.

Each fact has exactly five semantic fields. entity is the
concrete grammatical subject. relation is a concise
canonical action for an event, or a persistent attribute
for a state; it must not contain the current state value.
argument is the explicitly stated participant, object,
destination, preference value, or state value. time is an
explicit event/state time in ISO-8601 form only when the
source states it, otherwise unknown. kind is event for an
occurrence/action and state for a persistent property whose
argument may change across turns.

For state facts, choose relation narrowly enough to
distinguish unrelated attributes but consistently enough
that updates to the same property share the same entity and
relation. For user turns, retain explicit autobiographical
events, states, preferences, plans, constraints, and
updates. For assistant turns, retain only minimal direct
answers, named recommendation results, explicit commitments,
assigned tasks, or table entries. A named list may yield one
fact per item, but do not atomize rationales, descriptions,
steps, caveats, definitions, or background. Do not infer
facts, dates, entities, or relations that are not supported
by the source.
```

The user message supplies the batch in the following form:

```
{
  "turns": [
    {
      "turn_id": "<source identifier>",
      "speaker": "<user or assistant>",
      "text": "<original dialogue turn>"
    }
  ]
}
```

Validation and Source Preservation. The response must cover the supplied turn identifiers. Facts without a valid entity or relation, or with a kind other than `event` or `state`, are discarded. Unavailable arguments are normalized to `unknown`; exact duplicates within a turn are removed using the normalized tuple (e, r, u, τ, c). The implementation normalizes the Writer's time field but does not independently verify that it appears in the source text.

The five extracted fields are mapped deterministically to the internal atom representation; this adds no new facts or LLM pass. Original turns remain available as raw source records, including turns with `facts: []`. If a failed batch is split and extraction still fails for an individual turn, the enabled raw-source fallback preserves that turn without claiming an extracted atom.

Source Groups and Associations. Atoms from the same turn share a source binding, implemented as `episode:` followed by the original source-turn identifier s. Group membership therefore follows provenance, not semantic similarity. The prefixed binding ID is an implementation handle; the source-ID sets in the mathematical description contain the original source identifiers s.

Cross-turn associations use the keys defined in equation 2: the event key $\kappa_{\text{evt}}(a) = \nu(e, r, u)$ for atoms with all required fields, and the state-attribute key $\kappa_{\text{attr}}(a) = \nu(e, r)$ when $c = \text{state}$. The latter omits u so that different values of the same attribute can be associated. Neither key includes τ or s. Distinct source bindings sharing an exact normalized key of the same type are linked bidirectionally; self-links and duplicate links are removed. Key construction and linking require no additional LLM call.

The resulting links are candidate access paths, not assertions that two turns describe the same episode or a temporal succession. At retrieval time, atoms and links guide source access, while the preserved original turns provide the evidence for answering.

A.3 Memory Retrieval

This section specifies the query-plan format and planner prompt used in Section 3.3.

Query Evidence Grouping. The planner takes the question as input and returns evidence-group cues and query-level intent in one JSON response. Its output follows this schema (fields within `intent` are shown for illustration and need not all be populated for every query):

```
{
  "evidence_groups": [{
    "entities": ["<entity>"],
    "predicates": ["<relation>"],
    "event_types": ["<category>"],
    "terms": ["<discriminative term>"]
  }],
  "intent": {
    "kind": "<operation>",
    "selection": "<selection rule>",
    "index": 0,
    "comparison": "<comparison rule>",
    "return_role": "<requested information>",
    "anchor_group": 0,
    "target_group": 1,
    "temporal_relation": "<before or after>",
    "distinct": false,
    "expected_count": 0,
    "order_direction": "<asc or desc>"
  }
}
```

Unused cue lists are empty. The prompt uses `expected_count=0` when the question does not specify a count; group and selection indices are zero-based. Anchor and target indices apply to relative temporal selection. The following abridged prompt summarizes the planning instructions:

```
Compile the question into the smallest set of independently
required evidence groups. Use separate groups for the two
sides of a comparison or temporal relation; use one group
for a single preference or current-state question.

For each group, provide entities, stable predicates,
event-type categories, and discriminative literal terms.
Record the query-level operation, selection rule, requested
answer role, and any explicit count or ordering requirement.
For relative temporal selection, identify the anchor and
target groups.

Return JSON using the specified schema. Do not answer the
question or invent facts.
```

After the response, cue lists are normalized and empty groups are removed. If no usable group remains, the implementation creates one group from query terms. Explicit temporal constraints may

also be derived deterministically from the question and added to the intent. These operations do not make another planner LLM call.

We denote the ordered evidence groups used by retrieval after this processing by $\mathcal{G}(q) = (g_1, \ldots, g_m)$, with $g_i = (E_i, P_i, T_i, L_i)$. The fields `entities`, `predicates`, `event_types`, and `terms` correspond to E_i, P_i, T_i, L_i, respectively. Mathematical group indices are one-based, whereas JSON group indices are zero-based: for the same ordered group list, JSON group index $i - 1$ refers to g_i. The query operation in `intent.kind` is distinct from the atom kind c.

DG-Address Implementation. DG-Address processes each evidence group in two stages: cue-based entrance admission and, when needed, semantic supplementation.

Cue-based entrance admission. For each evidence group g_i and original source turn with identifier s, DG-Address evaluates the atoms associated with s. In the main configuration with `dg_complete_atom_view` enabled, entity cues E_i are matched against stored entities and aliases, predicate cues P_i against atom relations r, and normalized lexical terms L_i against a source-bound text view containing atom fields, the proposition, and the original turn. The `minimal` atoms have no independent event-type value in this configuration; T_i provides no separate direct-match support and is not matched to c. Matches from atoms belonging to the same source are combined; lexical terms that merely repeat a structured cue provide no additional support. The source-level cue-match indicators are $e_{is}, p_{is}, \ell_{is}$, with $n_{is} = p_{is} + \ell_{is}$, and support tiers follow equation 3. With `lexical_soft_cue` enabled, Lazy Dense triggering additionally uses a separate `direct_verified` check that excludes lexical support without changing these tiers. Sources are ranked by support tier and cue coverage. DG-Address admits all distinct verified sources for the group, or its highest-ranked partial source if no verified source exists. Native admission requires an atom-backed match. We denote the resulting native source-ID entrance set by N_i.

Need-aware semantic supplementation. For eligible multi-memory requests, DG-Address may supplement these entrances using embedding similarity. The query combines the group's predicate cues P_i, event-type cues T_i, and discriminative lexical terms L_i; the atom-side representation combines its relation r, argument u, and source-grounded proposition. The constraint-aligned policy excludes candidates that conflict with the query's available subject, temporal, or polarity constraints. For synthesis requests with unspecified answer cardinality, supplementation stops once native matching has already admitted two distinct sources; other eligible set-valued requests follow the configured multi-entrance policy. Supplemental entrances form the source-ID set S_i^+ and are added without removing native entrances. The resulting source-grounded entrance set passed to CA3-Complete is $D_i = N_i \cup S_i^+$, as in equation 4; $S_i^+ = \varnothing$ when supplementation is not activated.

CA3-Complete Implementation. *Association traversal.* DG-Address supplies source-grounded atom bindings as entrances. Their original source identifiers form $D_q = \bigcup_{i=1}^{m} D_i$. For each entrance, CA3-Complete reads its linked binding IDs and retrieves atoms from those directly linked bindings; recovered atoms are not used for further traversal, and no associations are created at query time. Evidence from an admitted DG entrance is distinguished from evidence recovered through associations. Associations operate over bindings; readout operates over original source turns. We denote the set of original source IDs recovered through these one-hop associations by R_q.

Base-pool construction and union. All entrance source IDs in D_q are retained. When $|D_q| < k$, full-query Dense retrieval ranks original turns and adds previously unused source IDs until the base pool reaches k or no further source is available. These added source IDs form $F_k(q)$; $F_k(q) = \varnothing$ when $|D_q| \geq k$. The base pool is $B_q = D_q \cup F_k(q)$, as in equation 5. Dense-only turns never become traversal seeds. CA3 then adds sources in R_q without removing or reordering base sources, giving $A_q = B_q \cup R_q$, as in equation 6. Recovered atoms follow their stored temporal order within each chain, and chains are interleaved across evidence groups. Each new source ID contributes at most one representative atom. CA3 applies no additional relevance score or post-union cap; CA1-Select enforces the final k-source limit.

Conditional partial-source admission. After constructing A_q, evidence groups are checked in the order of $\mathcal{G}(q)$. Let V_i and U_i be the source-ID sets classified as verified and partial for g_i, respectively. A group is eligible only if $V_i = \varnothing$ and $U_i \setminus A_q \neq \varnothing$. The eligible group indices form $\mathcal{H}_q$, defined in equation 7. When $\mathcal{H}_q \neq \varnothing$, the first eligible group has index $i_q^\star = \min \mathcal{H}_q$. Its unused partial sources in $U_{i_q^\star} \setminus A_q$ are ranked by descending cue coverage, then ascending Dense

rank and source ID. Let $s_q^\star$ denote the highest-ranked such source ID. We set $P_q = \{s_q^\star\}$ when an eligible group exists and $P_q = \varnothing$ otherwise, giving $C_q = A_q \cup P_q$, as in equation 8. At most one source is appended per query; existing candidates and their order are preserved, no further traversal is initiated, and a single CA1-Select call follows.

Here, C_q denotes the candidate source-ID set, whereas $\mathbf{C}_q$ denotes the ordered, deduplicated list of the same source IDs. Membership tests and set unions use C_q; CA1 presentation and order-dependent fallback use $\mathbf{C}_q$.

CA1-Select Implementation. CA1-Select receives the query q, evidence groups $\mathcal{G}(q)$, query-level intent, source budget k, and the ordered candidate list $\mathbf{C}_q$ paired with the original turn text for each source. Source views use the first 1,600 characters of each turn, without summarization. The LLM is called once with the following system prompt:

```
Select the smallest sufficient set of evidence source IDs for a later
    answerer. Return JSON only as {"selected_source_ids":[...]}. Select
    at most the given budget and only from candidate IDs. Evaluate the
    selected sources jointly: maximize direct support for the question
    and complementary query-relevant facts, while avoiding sources that
    only repeat information already covered. Candidate source turns are
    grounded in the original dialogue; do not answer the question or
    infer facts not present in those turns.
```

The user message supplies the query, its evidence groups and intent, the budget k, and the candidate records containing source IDs and text. The returned IDs are checked for membership in C_q, deduplicated, and truncated to the budget. For queries requiring multiple evidence groups, comparison, or temporal ordering, designated DG/CA3 core sources are retained when the budget permits; a missing core is added or replaces a non-core selection. A valid nonempty selection may use fewer than k sources. If no valid ID is returned, the system selects sources deterministically in the order of $\mathbf{C}_q$ up to the budget.

The resulting source-ID set is $I_q = \mathrm{Select}_k(q, \mathbf{C}_q)$, with $I_q \subseteq C_q$ and $|I_q| \leq k$, as in equation 9. These IDs are resolved to original turns for the answerer. CA1-Select does not generate an answer. The parameter k bounds the final source readout and also sets the Dense-fill target for B_q; it does not cap the size of C_q or the selection cost.

B Experiments

B.1 Evaluation Metrics and Protocols

Evaluation Setup and Answer Metrics. We evaluate on LongMemEval-Oracle (LME-O), LongMemEval-S (LME-S), and LoCoMo. LME-O and LME-S contain the same 500 questions: the former provides an evidence-focused context, whereas the latter requires retrieval from the corresponding full interaction history. For LoCoMo, we evaluate the 1,540 answerable questions from Categories 1–4: 841 single-hop, 282 multi-hop, 321 temporal, and 96 open-domain questions. We exclude the 446 Category-5 adversarial questions because they require a separate binary-refusal protocol.

We use released implementations or public packages and retain each baseline's native memory-writing and retrieval procedure. A-MEM uses its released reproduction code, HippoRAG-2 uses the published package and native graph construction, REMem retains its released episodic gist/fact access procedure, and AssoMem follows the public clue-merging, associative-graph, and RIT-ranking pipeline. Our Mem0 baseline uses the open-source Mem0 package with an isolated Chroma store for each dialogue, preserving its native fact extraction, memory update, and top-k search. For LoCoMo, we retain speaker roles, image annotations, and session times during ingestion. Because Mem0 OSS does not expose the platform-only timestamp interface, temporal information is carried in the source text and metadata rather than through Mem0 Platform's native temporal indexing. We therefore refer to this baseline as Mem0. All methods use the same downstream Reader, answer prompt, and Judge.

We report judged answer accuracy

$$J = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} c_q, \tag{11}$$

where $c_q \in \{0, 1\}$ is the Judge's correctness label. Normalized exact matches are accepted deterministically; other predictions are evaluated for semantic equivalence against the reference answer. LongMemEval preference questions use the benchmark-specific personalized-response criterion.

For LoCoMo, we additionally report the official category-aware F1 score following the released evaluation implementation.[1] The scorer lowercases and Porter-stems tokens after removing punctuation and articles. It applies the released comma-separated matching procedure to Category-1 questions, uses the first semicolon-delimited reference for Category 3, and scores Categories 2 and 4 over the complete answer.

Evidence and History-Expansion Metrics. Let $\mathcal{I}_q$ be the deduplicated set of original source identifiers selected for answer generation, and let $\mathcal{G}_q$ be the annotated gold evidence-source set. Selected-source evidence recall for question q is

$$\mathrm{ER}_{\mathrm{sel}}(q) = \frac{|\mathcal{I}_q \cap \mathcal{G}_q|}{|\mathcal{G}_q|}. \tag{12}$$

The dataset-level score is macro-averaged over questions with at least one annotated gold source:

$$\mathrm{ER}_{\mathrm{sel}} = \frac{1}{|\mathcal{Q}_G|} \sum_{q \in \mathcal{Q}_G} \mathrm{ER}_{\mathrm{sel}}(q), \qquad \mathcal{Q}_G = \{q : |\mathcal{G}_q| > 0\}. \tag{13}$$

This metric measures whether the final Reader context covers the annotated evidence, rather than whether the generated answer is correct. We report it only when retrieved items retain exact turn-level provenance and follow the shared source-budget protocol. It is therefore unavailable for REMem, whose retrieved gist and fact nodes do not retain exact mappings to original dialogue turns, and is omitted for A-MEM because its native associative expansion is not constrained to a strict five-source Reader context.

Because LME-O and LME-S contain paired questions, we measure accuracy preservation under history expansion as

$$\Delta J = J_{\mathrm{S}} - J_{\mathrm{O}}. \tag{14}$$

A value closer to zero indicates less degradation when moving from the evidence-focused Oracle setting to the corresponding full interaction history. We report ΔJ only when the Oracle and full-history results use matched Reader and evaluation protocols.

Evaluation Controls. The final Reader budget is at most $k = 5$ original source turns. Candidate generation may produce additional sources, but the source-selection stage reduces the final context to at most five deduplicated turns. Source-matched comparisons use the same Reader, answer prompt, Judge, and maximum final source budget. Each baseline otherwise retains its native memory-writing and retrieval procedure. In particular, REMem retrieves up to five gist nodes and five fact nodes and is treated as a native-budget episodic baseline rather than a strictly source-matched $k = 5$ baseline.

When changing the LLM backbone, we rerun the complete memory pipeline with that backbone, from memory writing through retrieval-time LLM operations and answer generation. The embedding and Judge configurations are held fixed where the method permits; methods requiring a native retriever retain their original retrieval configuration. All metrics are macro-averaged over evaluation questions and reported as percentages. Every expected example must have exactly one result, and failed examples are not silently removed. A metric is reported as "–" when it cannot be computed faithfully rather than being assigned a value of zero.

B.2 Controlled Interference Evaluation

Setup and protocol. We use 150 stratified LongMemEval-S questions covering all six types. Evidence sessions are kept in all conditions, while remaining sessions form a non-evidence pool.

[1] https://github.com/snap-research/locomo

Table 4: **Category-level results on LoCoMo.** We report official category-aware F1 and judged answer accuracy J on the 1,540 answerable questions in Categories 1–4, together with overall selected-source evidence recall $\mathrm{ER}_{\mathrm{sel}}$. All methods use the DeepSeek-V4-Flash Reader with reasoning enabled and the same DeepSeek-V4-Pro Judge. Values are percentages; the best result in each column is bold, and "–" denotes an unavailable metric.

Method	Overall			Single-hop		Multi-hop		Temporal		Open-domain	
	F1	J	$\mathrm{ER}_{\mathrm{sel}}$	F1	J	F1	J	F1	J	F1	J
SEA-Mem	**49.3**	**58.5**	**73.1**	**59.1**	**63.5**	**38.4**	**34.0**	39.3	**72.3**	28.9	40.6
AssoMem	21.2	23.0	22.8	28.9	27.6	13.4	7.1	7.9	23.7	21.3	27.1
REMem	47.4	48.1	–	52.7	52.4	26.6	19.1	**57.7**	66.6	27.4	32.3
HippoRAG-2	40.0	47.8	58.8	52.8	56.1	29.2	23.4	13.3	50.8	26.9	36.5
A-MEM	39.0	47.2	–	50.5	55.2	37.1	29.8	14.9	45.2	24.2	35.4
Mem0	42.0	51.9	62.9	56.8	61.7	32.9	27.7	14.0	50.2	**32.3**	**43.8**
Dense	39.0	46.2	55.6	48.5	51.5	27.9	20.9	29.3	58.3	21.0	34.4

Table 5: **Detailed controlled-interference results at $k = 5$.** J denotes judged answer accuracy, $\mathrm{ER}_{\mathrm{sel}}$ selected-source evidence recall, and DO distractor occupancy. ΔJ is measured relative to Evidence-only for each method. Best values within each history condition are bold. All values are percentages.

	J			$\mathrm{ER}_{\mathrm{sel}}$			DO			ΔJ		
History	**SEA**	**Asso**	**Dense**	**SEA**	**Asso**	**Dense**	**SEA**	**Asso**	**Dense**	**SEA**	**Asso**	**Dense**
Evidence-only	**76.0**	69.3	64.0	81.4	**82.9**	78.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Random-50	**72.7**	64.0	59.3	**79.5**	79.1	71.9	**4.0**	13.1	14.7	**-3.3**	-5.3	-4.7
Semantic-50	**76.0**	60.7	58.0	78.0	**78.6**	70.2	**5.1**	17.3	19.9	**0.0**	-8.7	-6.0
Full-S	**73.3**	62.0	58.0	**79.0**	78.2	70.2	**5.1**	17.5	19.9	**-2.7**	-7.3	-6.0

SEA = SEA-Mem; Asso = AssoMem-native; Dense = Dense-native.

Evidence-only keeps only evidence; Semantic-50 adds the most query-similar non-evidence sessions up to roughly 50% of non-evidence tokens; Random-50 adds a seeded random subset matched in session count and token volume; Full-S keeps the full history. All conditions are built from Full-S memory snapshots by copying retained atoms/turns unchanged and removing excluded sources/links, after which retrieval, selection, generation, and judging are rerun.

We compare SEA-Mem, Dense-native (top-k embedding ranking), and AssoMem-native (native RIT ranking without CA1-Select). SEA-Mem uses CA1 sufficient-set selection with $k = 5$ as the maximum raw turns passed to the Reader (fewer may be selected). All methods share the same cohort, embedding model, Reader, Judge, and budget. We report judged accuracy J, selected-source evidence recall $\mathrm{ER}_{\mathrm{sel}}$, and distractor occupancy DO. For condition c, with selected sources $\mathcal{I}_q^{(c)}$ and added non-evidence sources $\mathcal{D}_q^{(c)}$,

$$\mathrm{DO}(q, c) = \frac{|\mathcal{I}_q^{(c)} \cap \mathcal{D}_q^{(c)}|}{|\mathcal{I}_q^{(c)}|},$$

macro-averaged over questions. We also report ΔJ vs. Evidence-only and counts of correct-to-wrong and wrong-to-correct transitions.

Results. SEA-Mem is the most robust method under added history. Its judged accuracy remains essentially unchanged under Semantic-50 and drops only slightly under Full-S, whereas both baselines degrade substantially. It also maintains much lower distractor occupancy. Although AssoMem-native attains slightly higher evidence recall under Semantic-50, its lower accuracy and higher DO indicate that SEA-Mem preserves answer quality through a cleaner bounded Reader context rather than by maximizing evidence coverage alone.

Readout-Budget Sensitivity. We evaluate Semantic-50 under maximum Reader budgets $k \in \{2, 5, 10\}$, where k is the maximum final source budget and also determines the Dense-fill target

for the base pool; it does not cap the association-augmented candidate pool. All settings share the same Reader and Judge, and J, DO, and $\mathrm{ER}_{\mathrm{sel}}$ are computed over all 150 questions.

From Table 6, increasing the budget improves all methods, but SEA-Mem achieves the highest judged accuracy at every budget while keeping distractor occupancy low. In contrast, AssoMem-native and Dense-native attain higher evidence recall yet lower accuracy and much higher DO, showing that retrieving more evidence does not necessarily yield a cleaner Reader context. SEA-Mem also uses fewer turns on average due to its sufficient-set policy, so the comparison reflects each method's use of the same maximum readout capacity rather than equal realized context size.

Table 6: **Readout-budget sensitivity under Semantic-50.** J denotes judged answer accuracy, $\mathrm{ER}_{\mathrm{sel}}$ selected-source evidence recall, and DO distractor occupancy. Higher J and $\mathrm{ER}_{\mathrm{sel}}$ are better; lower DO is better. All values are percentages, and the best result within each budget and metric is bold.

	$k=2$			$k=5$			$k=10$		
Method	J	$\mathrm{ER}_{\mathrm{sel}}$	DO	J	$\mathrm{ER}_{\mathrm{sel}}$	DO	J	$\mathrm{ER}_{\mathrm{sel}}$	DO
SEA-Mem	**58.0**	**63.0**	**6.3**	**76.0**	78.0	**5.1**	**80.0**	84.4	**6.0**
AssoMem-native	49.3	60.8	8.7	60.7	**78.6**	17.3	71.3	**89.7**	27.5
Dense-native	39.3	47.6	12.7	58.0	70.2	19.9	68.7	85.1	30.7

B.3 Efficiency Evaluation

We evaluate efficiency on a fixed set of 60 LongMemEval-S questions, with ten questions from each question type. All methods process the same histories with a maximum final readout budget of $k = 5$, while retaining their native memory-writing and retrieval procedures.

We separate online query workload from one-time memory-writing cost. Let

$$T_{\mathrm{online}}(q) = T_{\mathrm{LLM}}(q) + T_{\mathrm{emb}}(q),$$

where T_{LLM} and T_{emb} denote query-time LLM and embedding tokens. Let $T_{\mathrm{write}}(s)$ denote the one-time cost of constructing memory store s. In this benchmark, each store serves one query, so the lifecycle workload is

$$T_{\mathrm{life}}(q) = T_{\mathrm{online}}(q) + T_{\mathrm{write}}(s_q).$$

Reader-context tokens count only the final historical evidence passed to the answering model. Token counts include LLM input, LLM output, and embedding input; judging, latency, billing, and provider-internal reasoning tokens are excluded.

Table 7 reports the stage-level averages underlying Figure 2. SEA-Mem achieves the smallest Reader context among the completed methods, using approximately 175 historical tokens per query, and maintains a lower lifecycle workload than HippoRAG-2, REMem, and A-MEM.

Table 7: **Detailed token-efficiency results on LongMemEval-S60.** Reader context contains only historical evidence passed to the answering model. All values are mean tokens per question.

Method	n	**Reader context**	**Query total**	**Writing total**	**Lifecycle total**
Dense	60	702.8	84,885.5	0.0	84,885.5
AssoMem	60	595.2	770.1	168,099.4	168,869.5
SEA-Mem	60	**175.0**	89,088.9	130,426.3	219,515.2
HippoRAG-2	60	473.1	649.2	492,853.5	493,502.7
REMem	60	216.5	422.9	559,277.0	559,699.9
Mem0	60	186.8	346.8	1,764,586.0	1,764,932.8
A-MEM	60	6,204.5	6,371.0	2,691,350.1	2,697,721.1

B.4 Component Ablation Protocol

We use a stratified set of 150 LongMemEval-S questions and hold the Writer memories, query plans, source budget, answer model, and Judge fixed across all variants. The final Reader context contains at most five deduplicated original source turns. We compare the Full system against three ablations:

- **Full:** verified DG entrances, one-hop CA3 completion, conditional Dense fill, the partial-source rule, and one CA1 selection step.
- **w/o DG access block:** replaces DG entrances with a candidate-count-matched set of Dense entrances while retaining CA3, Dense-fill, and CA1. Because partial entrances are defined by DG, this variant also removes partial-source recovery.
- **w/o CA3-Complete:** retains the same DG entrances, Dense-fill, partial-source rule, and CA1, but disables one-hop association completion and its non-replacing union.
- **w/o CA1-Select:** preserves the complete Full candidate pool, including any appended partial source, but passes the first five candidates in their existing order to the Reader without LLM selection.

Lazy Dense retrieval is shared across these paths and is not treated as a separate component. Candidate evidence recall $\mathrm{ER}_{\mathrm{cand}}$ is computed before final selection, and $\mathrm{ER}_{\mathrm{sel}}$ after. In the paired analysis, $\mathrm{C} \rightarrow \mathrm{W}$ counts questions answered correctly by Full but incorrectly by the ablation, and $\mathrm{W} \rightarrow \mathrm{C}$ counts the reverse.

Table 8: **Detailed component-ablation results.** All variants use the same 150 questions and a maximum readout budget of $k = 5$. Evidence-recall metrics use the 146 questions with source-level annotations. $\#P$ is the number of questions for which partial-source recovery is triggered.

Variant	**Correct**	J	$\mathrm{ER}_{\mathrm{cand}}$	$\mathrm{ER}_{\mathrm{sel}}$	$\#P$	$\mathrm{C} \rightarrow \mathrm{W}$	$\mathrm{W} \rightarrow \mathrm{C}$
Full	**118/150**	**78.7**	**89.9**	**80.3**	65	–	–
w/o DG access block	81/150	54.0	63.1	57.5	0	40	3
w/o CA3-Complete	111/150	74.0	86.6	77.4	65	7	0
w/o CA1-Select	107/150	71.3	**89.9**	76.6	65	19	8

Full and w/o CA1-Select share the same candidate pool, as reflected by their equal $\mathrm{ER}_{\mathrm{cand}}$, so their difference isolates the effect of budgeted source selection. Full and w/o CA3-Complete use the same DG and partial-source rules; their paired 7-to-0 transition pattern isolates the contribution of one-hop stored associations. The w/o DG result has a broader interpretation because removing DG also disables its dependent partial-source recovery.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work to improve writing clarity and grammar, offer suggestions on experimental design, support the analysis and presentation of experimental findings, and assist with coding-related tasks. All AI-generated edits and recommendations were carefully reviewed by the authors before being incorporated into the manuscript. The authors retain full responsibility for the accuracy, integrity, and final content of this work.