CHAIN ON DEMAND: DEFERRED EVIDENCE ORGANIZATION FOR LONG-TERM CONVERSATIONAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term conversational memory faces a timing mismatch: write-time summaries and graphs organize history before a future query is known, while independent retrieval can allocate a limited evidence budget to redundant or fragmented turns. We introduce Chain on Demand (CoD), a query-time evidence organization framework over an unchanged turn-level memory. Given a broad candidate pool, CoD constructs a temporary evidence chain by allocating a fixed anchor budget according to query relevance, lexical diversity relative to selected anchors, and newly covered query terms, followed by deterministic restoration of each selected anchor's same-session neighborhood. The resulting chain is used only for the current query, while the underlying conversational memory remains unchanged. In a paired comparison over all 1,986 LoCoMo questions, CoD achieves 82.2% accuracy on 1,540 non-adversarial questions and 76.3% on all 1,986 questions. On LongMemEval-S, one fixed configuration that indexes both user and assistant turns obtains 86.6% across all 500 questions. These results show that long-term memory can preserve the original conversation while deferring evidence organization until the information need is known. Our code is available at `https://anonymous.4open.science/r/acso3n3A4C4/`.

1 INTRODUCTION

Conversational agents need to retain information across interactions that span days, weeks, or months. A later interaction may depend on a fact mentioned much earlier, a preference established in another session, or several pieces of information spread across the conversation. As history grows, the information needed for a later question becomes harder to recover from the full interaction record. Since the complete history rarely fits within the effective context available to the answer model, long-term memory systems must determine which parts of the history to present for the current query.

Long-term memory involves two separable decisions: what representation persists across interactions, and what evidence is presented for a particular query. Many systems construct summaries, segments, links, graphs, or hierarchical memory units during or shortly after interaction (Zhong et al., 2023; Pan et al., 2025; Xu et al., 2025; Rasmussen et al., 2025; Fang et al., 2026; Xu et al., 2026a;b). These representations can make long histories easier to search and manage. However, they commit to an organization before the later query reveals which details matter. Retaining the original turns avoids rewriting the conversational record, but does not determine how a limited query-time evidence budget should be allocated. Persistent storage and query-time evidence construction therefore serve different objectives.

Independent query-time retrieval addresses this timing problem only partly. Standard top-K retrieval scores candidates independently (Gao et al., 2024). Under a fixed budget, several similar turns may occupy the available slots while a less similar turn supplies a missing relation, temporal detail, or intermediate step. The retrieved turns can therefore be individually relevant but collectively insufficient. Query-conditioned compression and set-wise retrieval account for query requirements more explicitly (Xu et al., 2024; Lee et al., 2025), but conversational evidence adds another dependency: a selected turn may require nearby dialogue and temporal order to be interpreted. Simply

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.

Table 1: Query-time evidence construction in representative systems. "No generated semantic memory" excludes query-independent summaries, cues, and events; sparse or dense indexes do not count. "Fixed query procedure" excludes multi-round agent search. "Set-aware allocation" evaluates each candidate against the evolving selected set. "Source-context restoration" adds original neighboring turns after selection. The columns describe mechanisms rather than an ordinal ranking. † denotes document-level RAG.

Design	No generated semantic memory	Fixed query procedure	Set-aware allocation	Source-context restoration
LoCoMo dialog RAG (Maharana et al., 2024)	✓	✓	✗	✗
SeCom (Pan et al., 2025)	✗	✓	✗	✗
SetR† (Lee et al., 2025)	–	✓	✓	✗
CueMem (Wang et al., 2026)	✗	✓	✗	✓
ReFind (Li et al., 2026a)	✓	✗	✗	✓
Chronos (Sen et al., 2026)	✗	✗	✗	✓
CoD (ours)	✓	✓	✓	✓

retrieving more text does not necessarily resolve this issue, as models can fail to use relevant information embedded in long inputs (Liu et al., 2024). Recent conversational memory systems recover source context through temporal–semantic expansion, archive-level context recovery, or reranking and tool use (Wang et al., 2026; Li et al., 2026a; Sen et al., 2026). These approaches establish the value of source-context recovery, but leave open a central question: given a fixed candidate pool and anchor budget, how should conversation locations be selected jointly before their local context is reconstructed?

We introduce Chain on Demand (CoD), a query-time evidence organization framework that constructs a temporary evidence chain over an unchanged turn record. Given a query, CoD first retrieves a broad candidate pool, then allocates a fixed anchor budget using a set-aware relevance–diversity–coverage rule, and finally restores the original same-session neighborhood around the selected locations. The selected anchors therefore specify where evidence should be organized rather than constituting the final evidence context themselves. The resulting chain is constructed only for the current information need and discarded after answering, while the underlying conversational record remains unchanged. This separates persistent memory from query-specific evidence construction while making the allocation decision explicit and directly testable. An adaptive acquisition variant further allows unresolved evidence gaps to trigger additional query-time selection without modifying the persistent memory.

Table 1 positions CoD relative to representative query-time evidence construction designs. The comparison highlights three properties that jointly define our framework: preserving the original conversational record, using a fixed query procedure with explicit set-aware allocation, and restoring source context after selection.

Our central evidence is a paired LoCoMo comparison that changes only the allocation rule. Under the same memory, candidate pool, anchor budget, temporal reconstruction, answer model, and judge, CoD raises accuracy from 74.5% to 76.3%, a 1.86-point gain with a conversation-cluster 95% confidence interval of [0.59, 3.15]. Development diagnostics further isolate the contributions of evidence volume, temporal reconstruction, and set-aware allocation. On LongMemEval-S, a fixed configuration that indexes both user and assistant turns obtains 433/500 (86.6%) across all questions, supporting applicability on a second long-term memory benchmark.

Our contributions are threefold:

- We formulate long-term conversational memory around deferred evidence organization: the persistent conversational record is preserved, while query-specific evidence is constructed only after the information need is known.
- We introduce CoD, a query-time evidence organization framework that constructs a temporary evidence chain through broad candidate retrieval, set-aware fixed-budget anchor allocation, and same-session context reconstruction. An adaptive acquisition variant further allows unresolved evidence gaps to trigger additional selection.

- We isolate the effect of set-aware allocation in a paired comparison over all 1,986 LoCoMo questions: changing only the allocation rule improves accuracy from 74.5% to 76.3% (+1.86 points; conversation-cluster 95% CI [0.59, 3.15]). Development analyses characterize reconstruction and allocation, and LongMemEval-S evaluates the complete pipeline on a second benchmark.

2 Related Work

Persistent Memory Construction. Long-term conversational memory systems differ in what they preserve across interactions. MemoryBank summarizes events and user traits (Zhong et al., 2023); SeCom builds topical segments and compressed retrieval units (Pan et al., 2025); and A-MEM augments stored interactions with attributes, links, and evolving notes (Xu et al., 2025). Other systems use temporal graphs, structured summaries, or multiple levels of granularity (Rasmussen et al., 2025; Xu et al., 2026a; Fang et al., 2026; Kang et al., 2025; Xu et al., 2026b). These designs primarily determine what information persists and how it is indexed. Even when they route or filter memory according to a query, the underlying textual units or relations are constructed before that query is known. CoD instead keeps the original turn record unchanged and defers the allocation of query-specific evidence until the information need is available.

Query-Time Evidence Allocation and Reconstruction. Standard retrieval-augmented generation independently ranks candidates (Gao et al., 2024). MMR introduced relevance–diversity reranking (Carbonell & Goldstein, 1998); RECOMP performs query-conditioned compression (Xu et al., 2024); and SetR explicitly selects passages as a set (Lee et al., 2025). Multi-stage and agentic systems make retrieval or memory operations adaptive through deliberative search, learned policies, or program evolution (Cao et al., 2026; You et al., 2026; Packer et al., 2023; Yan et al., 2026; Yu et al., 2026; Liu et al., 2026). Evidence organization also appears outside conversational memory; for example, ZARA uses structured evidence selection for training-free activity reasoning (Li et al., 2026b).

In conversational memory, the closest systems already restore source context. CueMem independently retrieves generated cues, maps them to source-turn anchors, and expands over a precomputed temporal–semantic graph (Wang et al., 2026). ReFind combines an unmodified chat archive with multi-round agent-controlled lexical search and local context expansion (Li et al., 2026a). Chronos combines raw turn and extracted event calendars with reranking, same-session expansion, and an iterative tool loop (Sen et al., 2026). These lines of work provide query conditioning, raw-record access, set selection, and local expansion in different combinations. CoD combines these components into a query-time pipeline in which selected units are conversation locations for subsequent source reconstruction rather than final passages. Its allocator is fixed and non-agentic, allowing its contribution to be measured while holding the candidate pool, anchor budget, reconstruction, answer model, and judge fixed.

3 Methodology

Given a query q and an ordered memory of original conversation turns, Chain on Demand (CoD) constructs query-specific evidence at query time. The pipeline separates three decisions: which turns are available, which conversation locations receive a fixed anchor budget, and which original context is restored around those locations. Its core allocation mechanism explicitly accounts for the evolving selected set: a greedy rule re-scores candidates as anchors are added. Candidate access defines the feasible locations, while deterministic reconstruction maps selected locations back to source context without making another set-dependent allocation decision.

Let $\mathcal{M} = (m_1, \ldots, m_N)$ denote the persistent memory, where each turn $m_i = (x_i, r_i, s_i, \tau_i)$ contains text x_i, speaker role r_i, session identifier s_i, and session-local position τ_i. CoD never rewrites x_i, and its temporary evidence is not written back to memory. Figure 1 distinguishes the fixed-budget core pipeline from the adaptive gap-directed acquisition variant.

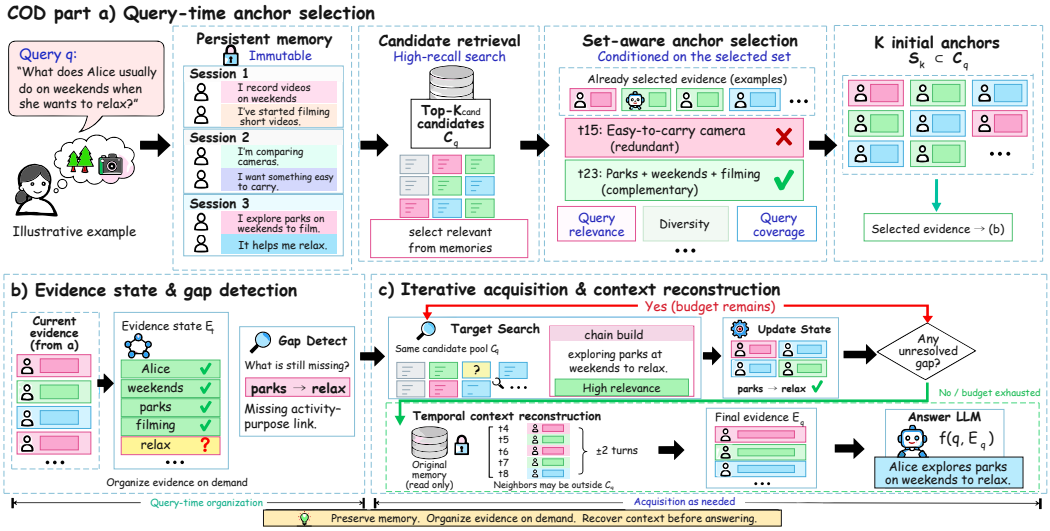

Figure 1: **Chain on Demand.** (a) Query-time set-aware allocation selects anchor locations from a fixed candidate pool, followed by deterministic same-session reconstruction. (b–c) An adaptive gap-directed acquisition variant iteratively updates a temporary evidence state and acquires additional anchors within the same candidate pool. The end-to-end results use the fixed-budget pipeline; the adaptive variant is evaluated separately with selection-level diagnostics.

3.1 Candidate Access

Candidate retrieval determines which turns are available to the allocator; it does not determine the final evidence. Each stored turn has a dense embedding, and the query retrieves

$$\mathcal{C}_q = \mathrm{TopK}_{m_i \in \mathcal{M}} \left(R(m_i, q); K_{\mathrm{cand}} \right), \qquad R(m_i, q) = s_{\mathrm{dense}}(m_i, q). \tag{1}$$

The resulting pool $\mathcal{C}_q$ defines the fixed search space for anchor allocation. Candidate access therefore determines evidence availability, while the subsequent allocator determines how the anchor budget is distributed.

3.2 Fixed-Budget Set-Aware Anchor Allocation

Independent top-K retrieval assigns each turn a fixed score. Once one turn is selected, those scores do not reflect what the selected set already contains. CoD instead recomputes a candidate's score after every selection. Starting from $S_0 = \emptyset$, it applies the greedy update

$$\begin{aligned} u_t(m; q) &= R(m, q) + \lambda D(m, S_t) + \beta C(m, q, S_t), \\ m^*_{t+1} &= \arg \max_{m \in \mathcal{C}_q \setminus S_t} u_t(m; q), \qquad S_{t+1} = S_t \cup \{m^*_{t+1}\}. \end{aligned} \tag{2}$$

Here, R is dense query relevance. The lexical diversity term

$$D(m, S_t) = 1 - \max_{s \in S_t} \mathrm{Jaccard}\big(\mathrm{tok}(m), \mathrm{tok}(s)\big) \tag{3}$$

penalizes overlap with selected anchors, with $D(m, \emptyset) = 1$. The coverage term $C(m, q, S_t)$ is the proportion of currently uncovered query terms matched by m; it is zero when no query terms remain uncovered.

The update repeats until $|S_t| = K_{\mathrm{anchor}}$. Because D and C depend on S_t, the score of an unselected candidate can change after every choice. This is the defining set-aware operation in CoD. The resulting turns are *anchors*: they identify conversation locations from which the query-specific evidence chain is subsequently constructed rather than serving directly as the final evidence context.

3.3 Deterministic Same-Session Reconstruction

An anchor can identify the relevant location while omitting dialogue needed to interpret it. For each selected anchor m_i, CoD restores a fixed neighborhood from the original memory:

$$\mathcal{N}_w(m_i) = \{m_j \in \mathcal{M} \mid s_j = s_i,\ |\tau_j - \tau_i| \leq w\}. \tag{4}$$

The final evidence $\mathcal{E}_q$ is obtained by taking the union of the neighborhoods of all selected anchors, removing duplicate turns, and preserving their original chronological order. The neighborhoods are read from $\mathcal{M}$ rather than $\mathcal{C}_q$, so a restored neighbor need not belong to the candidate pool.

Conditioned on the anchors and window width, reconstruction performs only same-session indexing, deduplication, and chronological ordering. It adds no semantic retrieval, generated memory, or further allocation decision. The anchors therefore define the locations from which the temporary query-specific evidence chain is constructed. The answer model receives q and $\mathcal{E}_q$, while the persistent memory remains unchanged.

Adaptive Gap-Directed Acquisition. Figure 1 also illustrates an adaptive variant that updates a temporary evidence state while reallocating anchors within the same fixed candidate pool. The procedure can acquire additional evidence when the current state leaves unresolved gaps. We evaluate this variant separately with selection-level diagnostics; it is not included in the end-to-end results of the fixed-budget pipeline. Appendix B.6 reports its configuration and diagnostic results.

4 Experiments

We evaluate CoD on LoCoMo (Maharana et al., 2024) and LongMemEval-S (Wu et al., 2025). Our main comparison isolates the effect of set-aware allocation by fixing candidate access, anchor count, reconstruction, answer model, and judge. We further evaluate performance across question types, analyze the roles of evidence volume and temporal reconstruction on a 100-question LoCoMo development subset, and measure the query-time cost of the evidence-construction pipeline.

4.1 Experimental Setup

For LoCoMo, we report results on both the 1,540 non-adversarial questions and the full set of 1,986 questions. For LongMemEval-S, we evaluate all 500 questions. GPT-4o-mini serves as both the answer model and evaluator in our runs; answer generation uses temperature 0 and at most 512 tokens.

We retain each original turn with its speaker, timestamp, session, and dense embedding. Using `text-embedding-3-small` and Qdrant, CoD retrieves 200 candidates, allocates 60 anchors with $\lambda = 0.06$ and $\beta = 0.10$, and restores two turns on each side within the same session. LongMemEval-S indexes both user and assistant turns.

The primary comparison changes only the allocator; memory, questions, candidate pool, anchor budget, reconstruction, answer model, and judge remain fixed. Mechanism tests use a fixed 100-question LoCoMo development subset that is contained in the 1,986-question set, so these results are diagnostics rather than independent held-out estimates. Published values are reproduced from MemPro's comparison table (Liu et al., 2026), which combines reproduced held-out runs with values taken from prior papers. MemPro reserves 154/1,540 LoCoMo and 50/500 LongMemEval questions for framework evolution and evaluates on the remaining 1,386 and 450; our rows report all 1,540 and 500. These values therefore provide context rather than matched comparisons. Appendix A gives the full protocol.

4.2 Main Results

The matched comparison changes only the allocation rule while keeping the memory, candidate pool, 60-anchor budget, reconstruction, answer model, and judge fixed. On all 1,986 LoCoMo questions, CoD improves accuracy from 74.5% with relevance-only allocation to 76.3% (+1.86 points), with a conversation-cluster 95% CI of [0.59, 3.15]. The paired question-level comparison gives 135 wins and 98 losses for CoD (Appendix A.9). This comparison directly measures the effect of set-aware allocation under the same evidence-construction pipeline.

Table 2: Cross-protocol LoCoMo context with GPT-4o-mini (%). CoD is evaluated on all 1,540 non-adversarial questions. Multi, Open, Single, and Temp denote multi-hop, open-domain, single-hop, and temporal reasoning.

Method	Overall ↑	Multi	Open	Single	Temp
Full Text	73.83	68.79	56.25	86.56	50.16
RAG	63.64	55.32	47.92	70.99	56.39
LangMem	55.45	52.10	41.65	62.80	43.25
Mem0	36.49	30.85	34.38	38.41	37.07
A-MEM	64.16	56.03	31.25	72.06	60.44
MemoryOS	58.25	56.74	45.83	67.06	40.19
LightMem	70.26	62.06	42.71	74.67	74.14
SimpleMem	72.08	64.50	44.90	76.80	74.50
CoD (ours)	82.20	71.30	58.30	88.70	81.90

Table 3: Cross-protocol LongMemEval-S context with GPT-4o-mini (%). The CoD row comes from one fixed configuration that indexes all turns and is applied to all 500 questions. Temp, Multi, Know, User, Asst., and Pref. denote temporal reasoning, multi-session reasoning, knowledge update, single-user recall, single-assistant recall, and preference.

Method	Temp	Multi	Know	User	Asst.	Pref.	Overall ↑
Full Text	31.58	45.45	76.92	87.14	89.29	36.67	56.89
RAG	39.85	48.48	67.95	90.00	98.21	53.33	60.90
LangMem	15.79	20.30	66.67	60.00	46.43	60.00	37.20
Mem0	40.15	46.21	70.12	81.43	41.07	60.00	53.51
A-MEM	47.36	48.87	64.11	92.86	96.43	46.67	62.20
MemoryOS	32.33	31.06	48.72	80.00	64.29	30.00	44.66
LightMem	67.18	71.74	83.12	87.14	32.14	68.18	69.81
SimpleMem	69.17	60.90	78.21	85.71	75.00	73.33	71.60
CoD (ours)	91.70	84.20	91.00	98.60	78.60	50.00	86.60

On the non-adversarial LoCoMo split, CoD achieves 82.2% accuracy (1,266/1,540), and it achieves 76.3% (1,516/1,986) on the complete benchmark. Across question types, the accuracy is 71.3% for multi-hop, 58.3% for open-domain, 88.7% for single-hop, and 81.9% for temporal questions. Table 2 places these results alongside selected published results. Since those rows are not protocol-matched, the paired comparison above provides the evidence for the effect of the allocation rule.

On LongMemEval-S, one fixed CoD configuration that indexes both user and assistant turns obtains 86.6% accuracy (433/500). The corresponding accuracies are 91.7% for temporal reasoning, 84.2% for multi-session reasoning, 91.0% for knowledge updates, 98.6% for single-user recall, 78.6% for single-assistant recall, and 50.0% for preferences. The two benchmarks provide complementary evidence: the paired LoCoMo comparison isolates the allocator's effect, while the unified LongMemEval-S run evaluates one fixed evidence-construction configuration across different question types. Since LongMemEval-S does not include a matched relevance-only allocator ablation, these results provide cross-paper context rather than evidence of an allocator improvement.

On the 56 assistant-memory questions, dual-role indexing raises accuracy from 11/56 (19.6%) with user-only indexing to 44/56 (78.6%). This diagnostic indicates that access to assistant turns matters for these questions in the evaluated pipeline. It does not test the set-aware allocator and is not used to construct the unified 500-question result.

In our measured implementation on the same 100 queries, caching token sets and pairwise overlaps reduces selection time from 7,046 to 115 ms/query and total memory-construction time from 7,183 to 262 ms/query, without changing the resulting evidence statistics. Dense-60 retrieval takes 21 ms/query, so the optimized evidence-construction pipeline adds 241 ms/query. CoD uses 13,218 prompt tokens per query versus 4,206 for Dense-60. This comparison reports the footprint of the

Table 4: Evidence construction on the LoCoMo diagnostic (%, $N = 100$).

Method	Acc. ↑	Turns	Words
Dense-60	51.0	60.0	1,268
Dense-200	57.0	200.0	4,587
Dense + Temporal	73.0	202.87	4,566
CoD	**77.0**	202.82	4,575

full pipeline rather than allocator-only overhead. Detailed latency and token-usage measurements appear in Appendices A.7 and A.8.

4.3 Controlled Mechanism Tests

We next examine three factors on the fixed 100-question LoCoMo development diagnostic: evidence volume, local reconstruction, and set-aware allocation. These results are intended to explain the mechanism and are not independent benchmark estimates.

Increasing independent dense retrieval from 60 to 200 turns raises accuracy from 51.0% to 57.0%. At nearly the same final volume, restoring local same-session context around relevance-selected anchors reaches 73.0% (202.87 turns and 4,566 words, compared with 200 turns and 4,587 words). Thus, on this diagnostic, increasing independent retrieval to a similar evidence volume does not reproduce the effect of local reconstruction.

Dense plus temporal and CoD use the same candidate pool, 60-anchor budget, and reconstruction procedure, and produce nearly identical context sizes: 202.87 versus 202.82 turns and 4,566 versus 4,575 words. Replacing relevance-only anchors with set-aware allocation increases accuracy from 73.0% to 77.0% at essentially unchanged evidence volume. The resulting comparisons separate evidence quantity from organization: Dense-60 versus Dense-200 gives a 6-point increase from broader independent retrieval, Dense-200 versus Dense plus Temporal gives a 16-point increase from local reconstruction at similar volume, and Dense plus Temporal versus CoD gives a 4-point increase from changing the anchor allocation. These development results are not an additive decomposition of benchmark-wide performance. Section 5 further examines diversity, coverage, candidate availability, and alternative allocation rules.

5 Mechanism Analysis

Section 4 provides a matched result over all 1,986 LoCoMo questions showing that set-aware allocation improves answer accuracy. We now use the fixed 100-question LoCoMo development diagnostic to examine where query-time evidence construction creates value. We consider local source reconstruction, the components of set-aware allocation, candidate availability, and the adaptive acquisition variant. These analyses explain the mechanism; they are not independent held-out estimates.

5.1 Local Source Context Helps Beyond Evidence Volume

Figure 2(a) compares independent retrieval, random expansion, and temporal reconstruction starting from the same 60 relevance-only anchors. Top-300 raw retrieval is a separate independent-retrieval control. Random expansion samples non-anchor turns from the full conversation; its token-matched version samples to approximately match the reconstructed evidence volume and reports the mean over five seeds (42–46).

Accuracy is 63.0% for Top-300 raw retrieval, 67.0% for random expansion, 68.4% for token-matched random expansion, and 73.0% for temporal reconstruction. The 4.6-point gap to the approximately token-matched control shows that matching evidence volume alone does not reproduce the result. Instead, restoring the local same-session context around relevance-selected anchors provides additional value. This comparison supports temporal reconstruction as a pipeline component but does not test the allocator.

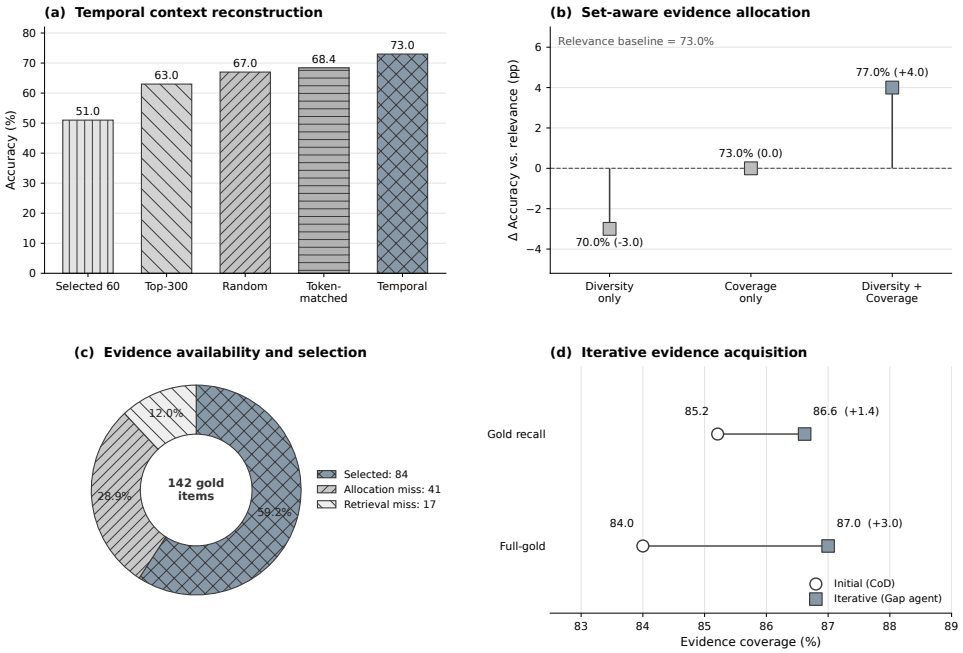

Figure 2: Analysis of query-time evidence construction on the 100-question diagnostic subset. (a) Temporal reconstruction improves over independent retrieval and random expansion at similar evidence volume. (b) Combining diversity and coverage with relevance improves set-aware allocation over the relevance-only baseline. (c) Gold evidence is missed both during candidate retrieval and anchor allocation. (d) Iterative evidence acquisition improves evidence coverage in the selection-only diagnostic.

5.2 Set-Aware Allocation

Figure 2(b) isolates the components of the allocation rule. With the same candidate pool, anchor budget, reconstruction, answer model, and judge, relevance-only selection reaches 73.0%. Adding diversity alone ($R + D$) reaches 70.0%, adding coverage alone ($R + C$) reaches 73.0%, while the combined $R + D + C$ rule reaches 77.0%. The diversity and coverage terms are recomputed against the evolving selected set rather than applied as fixed per-turn scores.

We further test whether the combined rule needs to control the full anchor budget. The protected-coverage and protected-$D + C$ variants reach 72.0% and 73.0%, respectively, compared with 73.0% for relevance-only selection. Neither reproduces the 77.0% accuracy of CoD, where relevance, diversity, and coverage jointly influence all 60 anchor decisions. A gold-informed allocator reaches 78.0%, one point above CoD, but it directly uses gold evidence identities and is not deployable; we therefore use it only as a reference.

Together, these controls indicate that the diagnostic improvement is not reproduced by applying diversity or coverage only after a relevance-ranked prefix has been fixed. The full set-aware rule allows the three signals to interact throughout anchor allocation. The corresponding matched comparison over all 1,986 questions gives a +1.86-point gain with a conversation-cluster 95% CI of [0.59, 3.15] (Section 4).

5.3 Candidate Availability Leaves an Allocation Opportunity

Figure 2(c) tracks the 142 dataset-provided gold evidence items before reconstruction. The candidate pool contains 125 items (88.0%); 84 become anchors, 41 are available but not selected, and 17 are absent from the pool. Among the 58 items not selected as anchors, 41 (70.7%) are therefore already visible to the allocator.

This leaves a substantial allocation opportunity: most gold items missed at the anchor stage are present in the candidate pool rather than unavailable to retrieval. At the same time, 17 gold items remain absent from the pool. These are anchor-stage counts, not final-context misses or answer errors, since reconstruction can still restore an unselected turn as a neighbor of another anchor.

5.4 Adaptive Evidence Acquisition

Figure 2(d) evaluates the adaptive gap-directed acquisition variant using selection-level evidence coverage. Starting from the same initial configuration, iterative acquisition improves Gold Recall@60 from 85.2% to 86.6% and Full-Gold Coverage from 84.0% to 87.0%. The procedure maintains a temporary evidence state and acquires additional anchors when unresolved evidence gaps remain, while keeping the candidate pool fixed.

These results show that additional query-time acquisition can improve evidence coverage without modifying the persistent memory. They are reported only as selection-level diagnostics: the end-to-end accuracy results in Section 4 use the fixed-budget pipeline without the iterative loop. Appendix B.6 gives the full configuration and diagnostic protocol.

6 Conclusion

We introduced Chain on Demand (CoD), a query-time evidence construction framework that preserves the original conversational record and organizes evidence only after the information need is known. Given a broad candidate pool, CoD allocates a fixed anchor budget with set-aware greedy scoring and then reconstructs the original same-session context around the selected locations, forming a temporary evidence chain without modifying persistent memory.

On LoCoMo, changing only the allocation rule improves accuracy from 74.5% to 76.3% over all 1,986 questions (+1.86 points; conversation-cluster 95% CI [0.59, 3.15]). The development analyses further show the complementary roles of set-aware allocation and source-context reconstruction, while the LongMemEval-S results demonstrate that the same query-time evidence construction framework transfers to a second long-term memory benchmark. Together, these results support deferring evidence organization until the query is known, rather than committing the organization of conversational history in advance.

7 Limitations

Our evaluation has several limitations. First, the 100-question LoCoMo development subset is contained in the full 1,986-question evaluation, so the component analyses are development diagnostics rather than independent held-out estimates. Second, the answer model and judge are fixed to GPT-4o-mini in the reported runs; other models and evaluators remain untested. Third, LongMemEval-S does not include a matched relevance-only allocator ablation, so its results support applicability across benchmarks rather than an independent estimate of the allocator's effect.

The current configuration also fixes the candidate pool at 200 turns, the anchor budget at 60, and the reconstruction window at two turns on each side. We do not systematically study how these choices scale with longer histories or different evidence budgets. Query-time construction additionally incurs higher prompt-token and construction costs than Dense-60 in our measurements. Finally, reconstruction is limited to local same-session context and does not explicitly model long-range temporal dependencies or cross-session links.

The Use of Large Language Models (LLMs)

We used AI-assisted tools to revise wording, improve readability, and assist with LaTeX formatting. The authors reviewed and revised the resulting content and take responsibility for the final manuscript. Language models used in the experiments are described in the methodology and experimental setup. GPT-4o-mini is used as the answer model and evaluation judge: it generates answers from the evidence constructed by each memory system and performs the LLM-as-a-Judge evaluation.

Ethical Considerations

Our experiments use publicly available benchmark datasets and do not collect private user data. Long-term conversational memory may contain sensitive personal information in real-world deployments. Such systems should therefore incorporate appropriate privacy and data-governance mechanisms, including user consent, access control, and mechanisms for users to inspect, correct, or delete stored memories. The memory system has no access to or ability to modify the benchmark gold answers.

Reproducibility Statement

We provide the implementation and experimental resources needed to reproduce the reported results. The main paper describes the Chain on Demand (CoD) pipeline, including candidate retrieval, set-aware anchor allocation, and same-session reconstruction. Appendix A specifies the experimental protocols, hyperparameters, evaluation settings, and dataset processing procedures, while Appendix A.9 describes the statistical analysis used for the matched comparison. The development diagnostics and adaptive acquisition variant are described in Appendix B.6. Our anonymous source code and supporting files are available at `https://anonymous.4open.science/r/acso3n3A4C4/`.

References

Shuqi Cao, Jingyi He, and Fei Tan. HiGMem: A hierarchical and LLM-guided memory system for long-term conversational agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33853–33862, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1690. URL https://aclanthology.org/2026.findings-acl.1690/.

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR98, pp. 335–336. ACM, August 1998. doi: 10.1145/290941.291025. URL http://dx.doi.org/10.1145/290941.291025.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 98706–98729, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a05b72653ec5b473732129829ae04195-Paper-Conference.pdf.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17606–17619, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.861. URL https://aclanthology.org/2025.acl-long.861/.

Ruizhe Li, Licheng Zhang, Benfeng Xu, Mingxuan Du, Zheren Fu, and Weidong Chen. When your agent opens the chat app: Agent-controlled search over raw chat logs rivals structured memory, 2026a. URL https://arxiv.org/abs/2608.12888.

Zechen Li, Baiyu Chen, Hao Xue, and Flora D. Salim. ZARA: Training-free motion time-series reasoning via evidence-grounded LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14986–15008, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.684. URL https://aclanthology.org/2026.acl-long.684/.

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. URL https://aclanthology.org/2024.tacl-1.9/.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

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.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 91851–91885, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e56f394bbd4f0ec81393d767caa5a31b-Paper-Conference.pdf.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Sahil Sen, Elias Lumer, Anmol Gulati, and Vamse Kumar Subbiah. Chronos: Temporal-aware conversational agents with structured event retrieval for long-term memory, 2026. URL https://arxiv.org/abs/2603.16862.

Changjian Wang, Rongzhen Li, Weili Guan, Shuming Shi, Quan Lu, and Ning Jiang. Cuemem: Cue-guided context reconstruction for long-term conversational memory, 2026. URL https://arxiv.org/abs/2609.12354.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2025. URL https://arxiv.org/abs/2410.10813.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. StructMem: Structured memory for long-horizon behavior in LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-391-3. doi: 10.18653/v1/2026.acl-short.12. URL https://aclanthology.org/2026.acl-short.12/.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/93e444b4a1a64815631c3c74977ee200-Paper-Conference.pdf.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 43478–43502, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/bda88ed2892f5e61c9a9bf215c566913-Paper-Conference.pdf.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

Zhixing You, Jiachen Yuan, and Jason Cai. D-mem: A dual-process memory system for llm agents, 2026. URL https://arxiv.org/abs/2603.18631.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory, 2023. URL https://arxiv.org/abs/2305.10250.

A Additional Experimental Details

This appendix documents the evaluation scope, core CoD configuration, controlled protocols, efficiency, statistical analysis, and additional diagnostics. The evidence hierarchy follows the main paper: the matched 1,986-question LoCoMo comparison supports the allocator claim; the fixed 100-question subset provides development diagnostics; LongMemEval-S evaluates the complete pipeline on a second benchmark; and published systems provide cross-protocol context. Unless stated otherwise, end-to-end answer-accuracy experiments use the same answer-generation and judging pipeline.

A.1 Experimental Protocol

Datasets. We evaluate CoD on LoCoMo and LongMemEval-S. For LoCoMo, we report results on both the 1,540 non-adversarial questions and the full set of 1,986 questions. For LongMemEval-S, the main experiment uses all 500 questions.

Answer generation and evaluation. For each question, the memory system first constructs the evidence supplied to the answer model. The answer model then generates a response using the question and the constructed evidence. For our runs, we use GPT-4o-mini with temperature 0 and

Evaluation set	N	Role in the paper
LoCoMo diagnostic	100	Configuration and mechanism analyses
LoCoMo non-adversarial	1,540	Aggregate benchmark result
LoCoMo all questions	1,986	Matched allocator and aggregate result
LongMemEval-S	500	One fixed user–assistant indexing configuration

Table 5: Evaluation scope. The LoCoMo benchmark reports overlap with the development diagnostic subset; they are not disjoint held-out estimates.

a maximum of 512 generated tokens as both the answer model and the LLM-as-a-Judge evaluator. The same generation and judging pipeline is used when comparing CoD with matched evidence-construction variants.

Baseline comparisons. The external reference rows are transcribed from MemPro's GPT-4o-mini comparison table (Liu et al., 2026). MemPro reports that this table combines baselines reproduced under its held-out protocol with values taken from the corresponding papers. Its LoCoMo protocol reserves 154 of 1,540 questions for framework evolution and evaluates on the remaining 1,386 question-disjoint examples; its LongMemEval protocol analogously uses 50 of 500 questions for evolution and 450 for held-out evaluation. In contrast, our CoD rows report all 1,540 LoCoMo and all 500 LongMemEval-S questions. The external rows are therefore contextual rather than matched comparisons. Controlled comparisons in this paper use the same memory, question set, answer model, and judging pipeline across evidence-construction variants.

Development and diagnostic subset. We use a fixed 100-question development and diagnostic subset to freeze the evidence-construction configuration and conduct controlled mechanism analyses. The subset is sampled from the full 1,986-question LoCoMo benchmark with stratified quotas of 14 multi-hop, 16 temporal, 5 open-domain, 42 single-hop, and 23 adversarial questions. Within each category, questions are sorted by question identifier and sampled with random seed 42, followed by a final shuffle.

The development subset is used to fix the evidence-construction hyperparameters before the benchmark-wide evaluations. Controlled comparisons on this subset are intended to isolate evidence volume, temporal reconstruction, and set-aware anchor allocation; they are not independent benchmark estimates. The reported 1,540- and 1,986-question LoCoMo results overlap with this development subset and should be read as complete-set reporting, not as disjoint held-out evaluation. The 1,986-question set contains all 100 diagnostic questions; the 1,540-question non-adversarial set contains its non-adversarial portion.

A.2 Evidence Selection Configuration

Candidate retrieval. At query time, CoD first retrieves a broad candidate pool of 200 conversation turns. This pool defines the turns available to the allocator but does not determine the final evidence view. Anchor allocation is handled by the subsequent selection stage.

Anchor allocation. From the candidate pool, CoD greedily selects 60 anchor turns. At each step, the next anchor is selected according to

$$m_{t+1}^{*} = \arg \max_{m \in \mathcal{C}_q \setminus S_t} \left[R(m, q) + \lambda D(m, S_t) + \beta C(m, q, S_t) \right], \tag{5}$$

where $R(m, q) = s_{\text{dense}}(m, q)$ is the dense retrieval score, D measures diversity with respect to the already selected set, and C measures the proportion of currently uncovered query terms that occur in the candidate. The full selector uses $\lambda = 0.06$ and $\beta = 0.10$.

Diversity. For a candidate m and selected set S_t, diversity is implemented as

$$D(m, S_t) = 1 - \max_{s \in S_t} \text{Jaccard}\big(\text{tok}(m), \text{tok}(s)\big), \tag{6}$$

Setting	Value
Dense retrieval	`text-embedding-3-small`; Qdrant
Candidate pool / anchor budget	200 / 60 turns
Allocator weights (λ, β)	$(0.06, 0.10)$
Same-session neighborhood	Two turns per side
Answer model / judge	GPT-4o-mini / GPT-4o-mini
Temperature / maximum output	0 / 512 tokens

Table 6: Configuration used for the main CoD evaluations.

with $D(m, \emptyset) = 1$. The selector tokenizes each turn by lowercasing and extracting tokens using the regular expression `[a-z0-9']+`; $\mathrm{tok}(m)$ denotes the resulting token set. A candidate therefore receives a lower diversity score when it substantially overlaps with an already selected turn.

Query coverage. Query terms are formed from extracted uppercase entities, excluding stop names, together with content words of length at least four. Candidate matching is performed by case-insensitive substring matching. For a given selection state, $C(m, q, S_t)$ measures the proportion of query terms not already covered by S_t that are matched by candidate m. If no uncovered query terms remain, the coverage contribution is zero.

Selector ablations. The clean selector ablations activate the components as follows:

- Relevance-only: $\lambda = 0,\ \beta = 0$;
- $R + D$: $\lambda = 0.06,\ \beta = 0$;
- $R + C$: $\lambda = 0,\ \beta = 0.10$;
- $R + D + C$: $\lambda = 0.06,\ \beta = 0.10$.

All selector variants use the same candidate pool, anchor budget, and temporal reconstruction procedure unless otherwise specified.

Ordering and ties. The greedy allocator uses a strict greater-than comparison, so an exact score tie retains whichever tied candidate is encountered first.

A.3 Same-Session Source-Context Reconstruction

The selected anchors are not directly passed to the answer model. For each anchor m_i, CoD retrieves the w preceding and following turns from the same conversation session:

$$\mathcal{N}_w(m_i) = \{m_j \mid s_j = s_i,\ |\tau_j - \tau_i| \leq w\}, \tag{7}$$

where s_i denotes the session identifier and τ_i denotes the session-local turn position. We use $w = 2$ in all main experiments.

Reconstruction operates on the original stored conversational memory rather than performing another retrieval over the candidate pool. Consequently, a same-session neighbor does not need to appear among the initial candidate turns. The retrieved neighborhoods are merged and deduplicated, and the retained turns are restored to their original chronological order before being passed to the answer model.

This distinction separates anchor allocation from context reconstruction: the anchor set determines which locations in the conversation are considered important, while reconstruction restores nearby dialogue needed to interpret those locations.

A.4 Evidence-Volume Controls

We use several controls to distinguish the effect of temporal reconstruction from simply providing more retrieved turns.

Top-300 Raw. Top-300 Raw supplies the 300 highest-ranked conversation turns directly, without temporal expansion.

Random Expansion. Random Expansion starts from the same relevance-only anchors used by the temporal reconstruction condition and adds randomly sampled non-anchor turns from the full conversation until the final number of entries approximately matches the reconstructed context. The resulting context contains approximately 203 entries on average. This control matches evidence-entry count rather than exact token count.

Token-matched random expansion. This control starts from the same relevance-only anchors as temporal reconstruction and does not change them. It repeatedly samples non-anchor turns from the full conversation until the resulting evidence approximately matches the token volume of the temporal-reconstruction condition. We run five independent seeds (42–46) on the 100-question development diagnostic and report their mean accuracy, 68.4%. This comparison tests whether local same-session structure contributes beyond the amount of evidence supplied to the answer model.

A.5 LongMemEval-S Protocol

The main LongMemEval-S result comes from one fixed configuration applied to all 500 questions, with both user and assistant turns indexed. The answer model, prompt, and evaluation procedure are also fixed across question types. This run achieves 86.6% accuracy (433/500). A separate 56-question role-coverage diagnostic compares user-only indexing with user–assistant indexing; its results appear in Appendix B.1 and are not used to construct the 500-question result.

A.6 Evidence Budget and Context Volume

To examine whether the improvement of CoD is explained simply by providing more retrieved evidence, we compare four evidence-construction conditions on the fixed 100-question diagnostic subset of LoCoMo. Table 4 in the main paper reports the accuracy and context volume for these conditions. The four conditions provide three controlled contrasts. Dense-60 versus Dense-200 measures broader independent retrieval (+6 points). Dense-200 versus Dense + Temporal compares different evidence organization at similar final volume (+16 points). Dense + Temporal versus CoD changes only anchor allocation at nearly unchanged context volume (+4 points). These development diagnostics are not an additive decomposition of benchmark-wide performance.

The matched evidence volume in this experiment refers to the retrieved conversation content rather than the complete API prompt. The latter also contains evaluation instructions and other prompt tokens and is therefore larger than the retrieved evidence itself.

A.7 Query-Time Efficiency

We measure query-time memory-construction latency on the same conversational memory under 100 diagnostic queries. The optimized implementation precomputes the token sets and pairwise Jaccard similarities used by the selector. The selection comparison and tie-breaking rule remain unchanged.

The original implementation spends most of its time repeatedly tokenizing turns and computing Jaccard overlap during greedy selection. Caching these representations reduces selection latency from 7.05 s to 115 ms per query, reducing total memory-construction latency from 7.18 s to 262 ms. Relative to Dense-60, the optimized implementation therefore introduces approximately 241 ms of additional memory-organization overhead per query.

The optimization changes only the implementation of the selector rather than its scoring rule or tie-breaking rule. The aggregate evidence statistics over the 100 diagnostic queries remain unchanged at 202.82 turns and 4,574.87 words.

A.8 API Context and Token Usage

We additionally record representative API usage statistics for the dense baseline and CoD on the diagnostic queries.

Comparison	Held fixed or matched	Question isolated
Relevance-only vs. CoD	Memory, questions, candidate pool, anchor budget, reconstruction, model, judge	Does set-aware allocation improve answers?
Dense + Temporal vs. CoD	Candidate pool, anchor budget, reconstruction procedure, approximate context volume	Does allocation help beyond local reconstruction?
Random vs. temporal expansion	Same relevance anchors; approximate turn count	Does local context help beyond more turns?
Token-matched random vs. temporal	Same relevance anchors; approximate evidence-token volume	Does local context help beyond more text?

Table 7: Purpose of the controlled comparisons. Approximate matching is stated explicitly; only the allocator comparison over all 1,986 questions is the benchmark-wide matched test.

Method	Retrieval	Selection	Reconstruction	Total
Dense-60	21	–	–	21
CoD (original)	30	7046	108	7183
CoD (optimized)	25	115	122	262

Table 8: Mean query-time memory-construction latency in milliseconds per query over 100 diagnostic questions. Stage means are rounded independently; Total is measured end to end. Caching changes the implementation but not the scoring rule.

The larger input prompt of CoD primarily reflects the larger final evidence context. Dense retrieval supplies approximately 60 turns and 1,268 words, whereas temporal reconstruction in CoD produces approximately 203 turns and 4,575 words. The completion length is similar across the two conditions, with 129.4 tokens for Dense and 111.1 tokens for CoD.

A.9 Paired Uncertainty and Significance

We compare CoD with the matched relevance-only allocator on the same 1,986 LoCoMo questions. CoD corrects 135 questions that the baseline answers incorrectly and loses 98 questions that the baseline answers correctly, giving a net gain of

$$100 \times \frac{135 - 98}{1986} = 1.86 \tag{8}$$

percentage points.

Primary cluster uncertainty. Because multiple questions belong to the same conversations, the primary uncertainty estimate is a conversation-cluster bootstrap over the 10 LoCoMo conversations. Each of 10,000 replicates samples conversations with replacement, retains their paired question outcomes, and recomputes the accuracy difference. The 95% percentile interval, using the 2.5th and 97.5th percentiles, is [0.59, 3.15] percentage points. Pairing is preserved because both systems answer the same questions under the same generation and judging pipeline.

Question-level paired checks. The continuity-corrected McNemar statistic is

$$\frac{(|135 - 98| - 1)^2}{135 + 98} = 5.562, \tag{9}$$

corresponding to a two-sided $p = .018$. For reference, a paired question-level bootstrap gives a 95% interval of [0.40, 3.37] percentage points. These question-level checks do not replace the conversation-cluster interval.

	Dense-60	**CoD**
Retrieved turns	60.0	202.82
Retrieved words	1,268	4,575
Prompt tokens	4,206	13,218
Completion tokens	129.4	111.1
Total tokens	4,335	13,330

Table 9: Per-query API usage statistics over the 100 diagnostic questions. Prompt-token counts include the surrounding evaluation prompt and are therefore not equivalent to retrieved-evidence tokens.

B Additional Results

B.1 LongMemEval-S Role Coverage

The main LongMemEval-S result uses a single dual-role indexing configuration applied uniformly to all 500 questions, where both user and assistant turns are searchable. This configuration achieves 86.6% accuracy (433/500).

Indexing	N	**Accuracy**
User-only	56	19.6
Dual-role	56	**78.6**

Table 10: Role coverage on 56 assistant-memory questions (%).

We additionally examine the role of assistant-side memory using the 56 single-session assistant questions. Under user-only indexing, the system answers 11/56 questions correctly (19.6%). When both user and assistant turns are searchable, the same 56-question subset reaches 44/56 (78.6%).

The 59.0 percentage-point difference indicates that restricting retrieval to user turns can prevent access to information explicitly provided by the assistant. This diagnostic is not used to construct the overall 86.6% result; the main result is obtained from the single dual-role indexing configuration applied uniformly to all 500 questions.

B.2 Temporal Reconstruction Controls

On the fixed 100-question diagnostic set, the evidence-volume controls give the following results:

The temporal condition achieves higher accuracy than raw top-ranked retrieval and both random-expansion controls. In particular, approximately matching the token volume with random expansion does not reproduce the temporal result.

B.3 Selector Ablations

We evaluate the components of the anchor allocator on the fixed 100-question diagnostic set. All variants retain relevance and differ only in whether they include the diversity term D, the coverage term C, or both.

Among the tested variants, $R + D$ and $R + C$ do not improve answer accuracy over relevance-only allocation, while $R + D + C$ improves it from 73.0% to 77.0%. Thus, only the combined rule reproduces the diagnostic gain; the complete-set matched experiment carries the benchmark-wide allocator claim.

Condition	Accuracy	Δ
Reconstruction and volume controls		
Top-300 Raw	63.0	–
Random Expansion	67.0	–
Token-matched Random Expansion	68.4	–
Temporal Reconstruction	**73.0**	–
Allocator ablations		
Relevance-only	73.0	–
$R + D$	70.0	-3.0
$R + C$	73.0	0.0
$R + D + C$ (CoD)	**77.0**	+4.0

Table 11: Reconstruction controls and allocator ablations on the fixed 100-question diagnostic set (%). Token-matched Random Expansion reports the mean over seeds 42–46. All allocator variants share candidates, budget, and reconstruction.

B.4 Pre-Reconstruction Anchor Availability

We characterize gold-evidence status at the anchor stage, before temporal reconstruction. On the diagnostic set, 142 dataset-provided gold evidence items are identified. The initial candidate pool contains 125 of these items, while 17 gold items are absent from the candidate pool. Of the gold items available in the candidate pool, 84 are selected as anchors and 41 are not.

Outcome	Count	Fraction
Gold items	142	100.0
Selected anchor	84	59.2
Candidate, not anchor	41	28.9
Absent from candidates	17	12.0

Table 12: Gold-evidence status before reconstruction (%).

Among the 58 gold items not selected as anchors, 41 (70.7%) are already present in the initial candidate pool and visible to the allocator. Candidate access therefore leaves an allocation opportunity on this diagnostic, although 17 gold items remain unavailable. Because reconstruction reads the original memory, an item not selected as an anchor may still enter the final context as a neighbor; the table therefore describes the anchor stage only.

B.5 Additional Allocation Comparisons

We compare additional allocation rules on the same fixed 100-question diagnostic set.

The protected variants retain a prefix of the dense ranking while applying additional selection criteria to the remaining budget. In this diagnostic, protecting the prefix does not improve over relevance-only selection. The gold-informed allocator provides a reference point for the remaining allocation headroom, but uses gold evidence identities and is therefore not deployable.

Allocation rule	Accuracy
Relevance-only	73.0
Protected coverage	72.0
Protected $D + C$	73.0
CoD (unprotected $R + D + C$)	**77.0**
Gold-informed reference	78.0

Table 13: Allocation controls on 100 diagnostic questions (%). The gold-informed reference is not deployable.

B.6 Selection-Only Iterative Evidence Acquisition

We additionally evaluate a gap-directed acquisition variant only as a selection-level diagnostic; it is not part of any end-to-end CoD result. Given an initial evidence state, the procedure iteratively identifies unresolved evidence gaps, scores candidate turns with respect to the current gap, and updates the evidence state. The candidate pool itself is fixed throughout the procedure; thus, this extension does not introduce an additional retrieval stage or expand the set of retrievable turns.

The temporary evidence state records selected evidence, grounded entities and events, temporal matches, unresolved information gaps, and unsatisfied query terms. We consider several types of unresolved gaps, including missing entities, missing facts or events, missing temporal relations, missing multi-hop bridges, and insufficient evidence for multi-faceted questions. Given an unresolved gap g, a candidate m is scored by extending the base allocation rule with gap-conditioned relevance and temporal usefulness:

$$u_t^{\text{gap}}(m; q, g) = R(m, q) + \lambda D(m, S_t) + \beta C(m, q, S_t) + \gamma G(m, g) + \delta T(m, S_t), \tag{10}$$

where $G(m, g)$ measures the relevance of candidate m to the unresolved gap, and $T(m, S_t)$ measures its temporal usefulness for the current evidence state. For this diagnostic, we use $\lambda = 0.06$, $\beta = 0.12$, $\gamma = 0.18$, and $\delta = 0.08$.

The procedure initializes the evidence state with two anchors and performs at most 80 acquisition steps, subject to a maximum of 60 anchors. For multi-faceted questions, the additional-evidence requirement is $\max(3, \min(8, 2+|\mathcal{T}_q|))$, where $\mathcal{T}_q$ denotes the query facets tracked in the temporary evidence state. At each step, the procedure selects evidence for an unresolved gap and updates the evidence state. If the procedure terminates before the 60-anchor budget is exhausted, the remaining slots are filled from the candidate pool so that the final anchor budget is comparable to the fixed-budget base selector. The gap-conditioned terms are used only in this iterative diagnostic; the main CoD configuration does not require an iterative acquisition loop.

Evidence-coverage metrics. Let G_q denote the dataset-provided gold evidence items for question q, and let S_q^{60} denote the final set of at most 60 selected anchors. We measure Gold Recall@60 as the fraction of gold evidence items directly selected as anchors:

$$\text{GoldRecall@60} = \frac{1}{N} \sum_{q=1}^{N} \frac{|G_q \cap S_q^{60}|}{|G_q|}. \tag{11}$$

We additionally measure Full-Gold Coverage after temporal reconstruction. Gold Recall@60 is the mean per-question item recall, whereas Full-Gold Coverage is a question-level all-evidence criterion: it asks whether all dataset-provided gold evidence for a question is contained in the final reconstructed evidence context $\mathcal{E}_q$:

$$\text{FullGoldCoverage} = \frac{1}{N} \sum_{q=1}^{N} \mathbf{1}[G_q \subseteq \mathcal{E}_q]. \tag{12}$$

Thus, Gold Recall@60 averages direct anchor recall across questions, whereas Full-Gold Coverage measures the fraction of questions whose complete gold evidence set is recovered after temporal reconstruction. The pooled 84/142 anchor count in Table 12 uses a different aggregation and is not directly comparable to mean Gold Recall@60.

On the fixed 100-question diagnostic set, iterative acquisition increases Gold Recall@60 from 85.2% to 86.6% and Full-Gold Coverage from 84.0% to 87.0% relative to the fixed-budget base selector. We additionally report an offline gold-informed planner as a reference. This planner directly uses gold evidence identities to guide selection within the candidate pool and is therefore not deployable. It reaches 90.8% Gold Recall@60 and 90.0% Full-Gold Coverage.

These results are selection-level diagnostics rather than end-to-end generative evaluation. They indicate that unresolved evidence gaps can provide useful signals for reallocating a fixed anchor budget, but they do not establish an answer-accuracy improvement from adding an iterative agent loop to the complete memory system. We therefore treat iterative evidence acquisition as an exploratory extension rather than a component of the main CoD evaluation.