Bounded Conversational Memory with Hybrid Retrieval and Evidence Highlighting for Multi-Session Dialogue Systems

Manasa Mudunuri
Department of Computer Science
Georgia State University
Atlanta, GA 30303, USA
mmudunuri3@student.gsu.edu

Chandra Kiran Guntupalli
Department of Computer Science
Georgia State University
Atlanta, GA 30303, USA
cguntupalli1@student.gsu.edu

Shiraj Pokharel
pokharel.shiraj@gmail.com

Murray Patterson
Department of Computer Science
Georgia State University
Atlanta, GA 30303, USA
mpatterson30@gsu.edu

Mario M. Kubek
Department of Computer Science
Georgia State University
Atlanta, GA 30303, USA
mkubek@gsu.edu

Abstract

Across independent sessions, dialogue systems built on large language models do not retain prior context unless an external memory layer organizes and retrieves it. Most memory systems treat this as a retrieval problem, assuming that surfacing the right past content is the primary lever for accuracy. We report evidence that this view is incomplete: holding retrieval fixed, *how* retrieved context is organized and presented to the model is an independent and substantial driver of answer accuracy. We demonstrate this with the Context Persistence Protocol (CPP), a modular memory layer that groups dialogue into topically coherent segments, retrieves them with hybrid lexical and semantic search, and highlights the single most informative sentence in each segment before generation. Across two long-term memory benchmarks, CPP attains session-level retrieval scores that are identical to those of an unbounded flat-retrieval baseline on LoCoMo and slightly lower on LongMemEval, yet it answers substantially more questions correctly, with the largest gains on multi-hop reasoning. These results suggest that the organization and presentation of retrieved context, rather than retrieval coverage alone, is a major driver of the difference.

1 Introduction

Conversational AI systems built on large language models typically do not maintain conversational state across independent sessions unless an external memory mechanism is introduced. A user who spent an hour explaining project requirements and preferences returns the next day to a blank slate. While prior interactions can in principle be stored in an external database, effectively organizing and retrieving the relevant information across sessions remains a significant challenge, and for systems aspiring to serve as long-term assistants, tutors, or collaborative agents, this is a fundamental architectural mismatch between how language models are built and how people use them over time (Chen et al., 2024b). Two distinct failure modes arise from this gap. The first is *volatile memory*: when a session ends, the entire conversational context is erased as if the conversation never happened. The second is *context overload*: even within a single session, retaining every prior turn eventually fills the context window, dilutes the model's attention, and degrades the quality of responses (Liu et al., 2024). Real conversations accumulate shared knowledge, evolve in purpose, and reference earlier exchanges in ways that cannot simply be reentered at the start of each new session. What is needed is not more storage, but smarter storage. Prior systems have navigated this tension through varied strategies: hierarchical memory architectures modelled on operating systems (Packer et al., 2023), atomic fact extraction into vector databases (Chhikara et al., 2025), and heat-based eviction

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 786–794
August 2–5, 2026.

with bounded capacity (Kang et al., 2025). These approaches have advanced the field considerably, but they share a common assumption that retrieval quality is the primary lever for downstream accuracy. We revisit that assumption. Our objective is not to propose a complete spoken or written dialogue architecture, but to isolate and study one specific challenge within conversational systems: long-term memory persistence across sessions. We treat memory as a foundational layer that supports higher-level dialogue capabilities such as personalization, continuity, and adaptation, and we position CPP as a memory-layer contribution rather than an end-to-end conversational system. We introduce the Context Persistence Protocol (CPP), a modular framework that addresses long-term conversational memory through four composable phases: grouping consecutive turns into topically coherent *segments*, dual-indexing each segment for hybrid keyword and semantic retrieval, retrieving the most relevant segments under a hard storage budget with cross-encoder reranking, and highlighting the single most informative sentence within each retrieved segment before passing context to the model. CPP is *bounded by design*: storage is explicitly capped, principled eviction prevents unbounded growth, and every component is independently replaceable. We evaluate CPP on LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025), two benchmarks spanning human-human and human-AI conversations respectively. Our results show that bounded CPP attains session-level retrieval metrics comparable to an unbounded baseline (identical on LoCoMo and marginally lower on LongMemEval) while answering more questions correctly under our benchmark settings, suggesting that the structure and presentation of retrieved context contribute substantially to the accuracy gain.

2 Related Work

Research on conversational memory predates retrieval-augmented generation and large language models: earlier dialogue systems maintained continuity across interactions through explicit dialogue-state representations, user models, and memory-augmented neural architectures. Lewis et al. (2020) later demonstrated that injecting retrieved passages into the prompt substantially improves language model factuality, establishing the retrieval-augmented generation (RAG) paradigm. Early conversational applications stored raw dialogue turns and retrieved them by embedding similarity (Karpukhin et al., 2020). While straightforward, turn-level stores grow without bound and create noisy retrieval surfaces where relevant evidence is diluted among conversational filler. This limitation motivated two research directions. The first sought richer representations: MemGPT (Packer et al., 2023) introduced OS-style hierarchical memory but requires $\sim$16,900 tokens per query; Mem0 (Chhikara et al., 2025) extracts atomic facts efficiently but stores without bound. The second sought bounded memory with principled eviction: MemoryOS (Kang et al., 2025) uses heat-based eviction; SimpleMem (Liu et al., 2026) achieves comparable accuracy through entropy-gated compression. GAM (Wu et al., 2026) and A-MEM (Xu et al., 2025) use graph-based structures but have not been evaluated under explicit storage budgets. Three findings from the retrieval literature inform CPP's design. Pan et al. (2025) showed that segment-level chunks of 3–10 turns grouped by topic consistently outperform both turn-level and session-level chunking, and that hybrid BM25 and dense retrieval outperforms pure dense retrieval by 15–30% in recall. Glass et al. (2022) identified cross-encoder reranking as a high-impact retrieval upgrade across RAG approaches. CPP incorporates all three as first-class architectural decisions and adds a systematic investigation of whether context *presentation quality*, independent of retrieval quality, is a primary accuracy driver, a question prior systems have not isolated. With this landscape established, we introduce the retrieval and evaluation terminology used throughout the paper before presenting CPP's architecture in full.

3 The Context Persistence Protocol

CPP organizes conversational memory into four composable phases (Figure 1); each is independently replaceable. We use standard retrieval terminology throughout: BM25 (Robertson and Zaragoza, 2009) for exact-match lexical scoring, RRF (Cormack et al., 2009) for rank fusion (k=60), a cross-encoder for joint (query, segment) reranking, Recall@k for the fraction of gold-relevant sessions in the top-k, and verbatim storage recall for the fraction of questions whose gold string appears in any stored segment, a coverage diagnostic, not an accuracy ceiling.

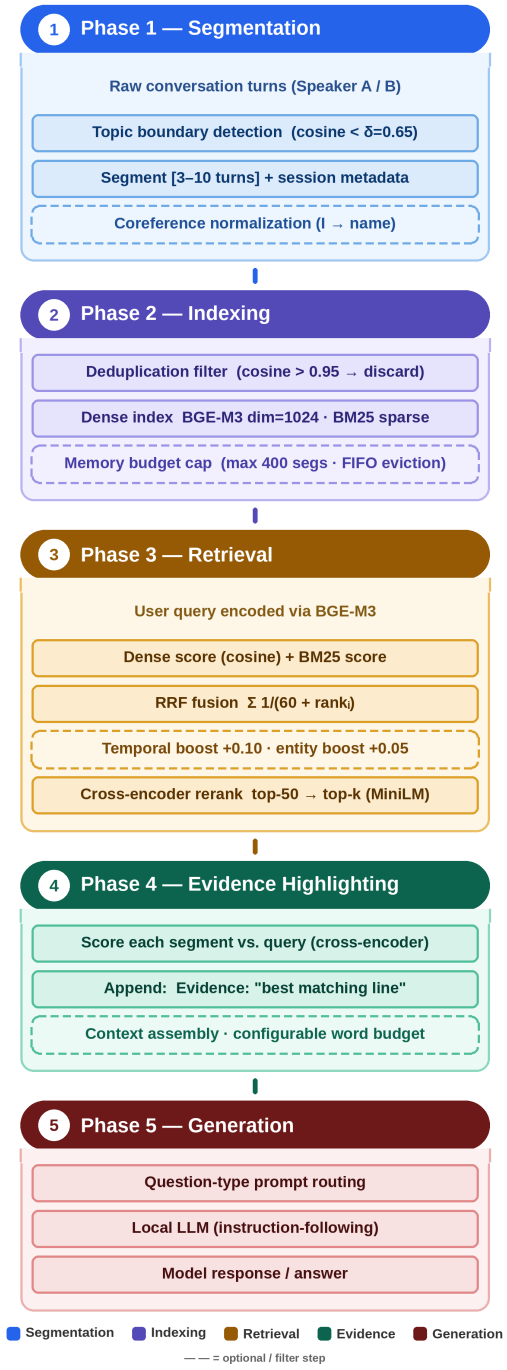

Figure 1: CPP pipeline. Turns are segmented, dual-indexed (dense and BM25), retrieved under a hard budget cap, and enriched with evidence highlighting before being passed to a local LLM. The framework is dataset- and model-agnostic.

3.1 Phase 1: Segmentation

Raw turns are not stored individually. CPP groups consecutive turns into *segments*: topically coherent windows of 3–10 turns. A new boundary is placed between turns i and $i{+}1$ when their embedding cosine similarity drops below threshold δ. Each segment is stored with a session identifier, a session datetime, and a temporal-signal flag indicating relative time expressions such as "yesterday." Coreference normalization resolves first-person pronouns to speaker names at ingest via a rule-based pass over speaker-tagged turns ("I went to the gym" in Alice's turn becomes "Alice went to the gym"), so entity-based retrieval succeeds on speaker-centric text. The 3–10 turn range follows Pan et al. (2025), who report that medium-sized topic-based segments consistently outperform both single-turn retrieval and very large session-level chunks: smaller segments lack sufficient conversational context, while larger segments reduce retrieval granularity and increase reranking cost. Although topic coherence could in principle produce arbitrarily long segments, unconstrained segments reduce retrieval specificity and can monopolize storage capacity when a conversation stays on a single topic for an extended period; the upper bound is therefore a practical retrieval and storage constraint rather than a claim about discourse structure. We deliberately restrict normalization to first-person references because memory queries frequently refer to participants by name while dialogue turns are dominated by first-person expressions ("I", "me", "my"). Full neural coreference resolution was not adopted because long conversational chains can introduce error propagation and incorrect entity substitutions that would be permanently stored in memory; evaluating broader coreference approaches remains future work. Segmentation produces richer retrieval units (a topical exchange carries far more discriminative signal than an isolated turn) and compresses the store 4–5$\times$ over turn-level storage.

3.2 Phase 2: Indexing

Each segment is indexed in two complementary ways. A *dense index* stores the mean-pooled BGE-M3 embedding (Chen et al., 2024a) (dimension 1024). A *sparse BM25 index* (Robertson and Zaragoza, 2009) stores tokenized segment text. Two mechanisms constrain the store: *deduplication* (segments with cosine similarity $\geq \tau$ to any stored segment are discarded) and a *budget cap* (the store is limited to $B{=}400$ segments; oldest are evicted FIFO when the cap is exceeded). We distinguish the roles of these two mechanisms. Deduplication is active throughout our experiments and contributes directly to the compression CPP

achieves. The budget cap, by contrast, is a *worst-case guarantee for production-scale deployment* rather than a mechanism exercised by our benchmarks: CPP stores $\approx$124 segments per conversation on LoCoMo (31% of the cap) and $\approx$108 on LongMemEval (27%), so FIFO eviction is never triggered, and on these two benchmarks the storage layer is effectively unbounded. The cap would activate only for conversations accumulating more than $\approx$2,000 turns, beyond the scale of either benchmark. We therefore do *not* claim that eviction contributes to the accuracy results reported here; its value is that it bounds memory growth for open-ended, indefinitely long deployments where turn counts are not fixed in advance. Empirically characterizing the accuracy–compression trade-off under tighter budgets $B \in \{50, 100, 200\}$ that force eviction is left for future work.

3.3 Phase 3: Retrieval

Dense cosine scores and BM25 scores are fused via RRF (Cormack et al., 2009). Temporal-signal segments receive $+0.10$ when the query contains temporal keywords (matched against a pattern list of relative expressions: "yesterday," "last week," "ago," "since," and similar phrases); named-entity matches receive $+0.05$ per entity (entities extracted from the query via a lightweight capitalized-phrase detector). The top-50 candidates are reranked by a cross-encoder (ms-marco-MiniLM-L-6-v2) (Reimers and Gurevych, 2019), and the final top-k segments are returned (k=10 for LoCoMo; k=15 for LongMemEval, selected on the development split). A low-confidence filter suppresses results when the maximum cosine similarity of the retrieved segments falls below a threshold, preventing weakly related context from misleading the model.

3.4 Phase 4: Evidence Highlighting

The cross-encoder scores every line of each retrieved segment against the query; the highest-scoring line is appended as:

```
Evidence: <best line>
```

This provides both full segment context (preserving narrative coherence) and an explicit pointer to the most relevant sentence, addressing the "lost in the middle" failure mode (Liu et al., 2024) in which models underuse information in the interior of long contexts.

3.5 Generation

Segments are formatted with session datetime headers, concatenated within a configurable word budget (the context-assembly step shown in Figure 1), and passed to a local LLM with a question-type-specific system prompt. For temporal questions, the prompt instructs the model to use session datetimes to anchor relative time expressions. For multi-session synthesis questions, the prompt instructs explicit enumeration before counting. The routing uses the benchmark-provided question type label; no label leakage occurs because the type label is not used during retrieval, only for prompt selection. Full prompt templates are available in the project repository. The generation step is model-agnostic.

4 Experiments

4.1 Research Hypotheses

We state four hypotheses, each tested by Tables 2 and 4. *H1:* a bounded segment-based system matches or exceeds unbounded flat retrieval despite storing far fewer items. *H2:* the largest gain is on multi-hop questions, where segments preserve topical context that flat retrieval scatters across isolated turns. *H3:* CPP may show a marginal deficit on open-domain inferential questions where diffuse signals are lost in segmentation. *H4 (presentation-quality isolation):* identical Recall@k between C2 and C3 (the flat-retrieval and full-CPP conditions defined in Section 4.6) would indicate that the advantage is presentation-driven rather than retrieval-driven, which is the central question CPP is designed to test.

4.2 Component Selection

Prior to benchmark evaluation we ran a component selection study on a curated 100-prompt knowledge base spanning ten domains with semantic distractors. Among three embedding models evaluated at top-5 retrieval on 50 held-out queries (Table 1), E5-base-v2 substantially outperformed the alternatives; we then upgraded to BGE-M3 for the main evaluation, whose learned sparse mode enables the hybrid dense+BM25 stack. The main evaluation uses Qwen2.5-14B (Yang et al., 2024) as the generator.

4.3 Experimental Setup

All experiments run on a single NVIDIA RTX 4090 GPU (24 GB). The LLM is Qwen2.5-14B-Instruct (Yang et al., 2024) in FP16. The embedding model is BAAI/BGE-M3 (Chen et al., 2024a)

Table 1: Embedding model selection (50 test queries, top-5 retrieval). E5-base-v2 was selected and subsequently upgraded to BGE-M3.

Embedding Model	P	R	F1
all-MiniLM-L6-v2	0.74	0.71	0.72
bge-small-en-v1.5	0.78	0.75	0.76
intfloat/e5-base-v2	**0.87**	**0.86**	**0.86**

(dim 1024). The reranker is ms-marco-MiniLM-L-6-v2 (Reimers and Gurevych, 2019). Generation uses greedy decoding (temperature $= 0$), seed 42, `max_new_tokens`$= 16$ for LoCoMo's single-letter MC format and 384 for LongMemEval's free-form answers. The cross-encoder truncates (query, segment) pairs to 512 tokens. The low-confidence filter suppresses retrieval when the maximum cosine similarity of retrieved segments falls below 0.25. Several parameters are fixed by design: RRF k=60, BGE-M3 dim 1024, context budgets of 1,200 and 2,400 words, B=400, and τ=0.95 for LoCoMo and 0.90 for LME. A development sweep over $\delta \in \{0.55\text{–}0.70\}$ and top-$k \in \{5, 7, 10, 15\}$ on a held-out 10% subset found δ=0.65 and k=10 to be optimal. Both benchmarks use only 27–31% of the B=400 cap, so FIFO eviction is not activated.

4.4 Dataset 1: LoCoMo

LoCoMo (Maharana et al., 2024) is a widely used benchmark for long-term conversational memory, used to evaluate MemoryOS, SimpleMem, and GAM (Kang et al., 2025; Liu et al., 2026; Wu et al., 2026), and its five question types give comprehensive coverage of the challenges CPP targets. The 10-option multiple-choice format removes generation noise from accuracy measurement. Each of its ten conversations spans 19–32 sessions (∼600 turns); we use the `locomo_mc10` split of 1,986 QA pairs across single-hop (n=282), multi-hop (n=321, directly testing H2), temporal (n=96), open-domain (n=841), and adversarial (n=446) types. The reported human ceiling is F1 = 87.9.

4.5 Dataset 2: LongMemEval

LongMemEval (Wu et al., 2025) complements LoCoMo on every axis: human-AI conversations with free-form answers judged by a local LLM (Qwen2.5-14B, binary), stressing generation quality. Its knowledge-update category, which is absent from LoCoMo, tests whether a system retrieves the *most recent* version of information that changed across sessions. Each of its 500 questions is answered from ∼40 sessions (∼115K tokens), spanning single-session (n=150), multi-session (n=121), temporal (n=127), knowledge-update (n=72), and abstention (n=30) types.

4.6 Experimental Conditions

C1 (No Memory) gives the LLM only the question, establishing the parametric knowledge lower bound. *C2 (Flat Retrieval)* uses an unbounded turn-level store (∼601 turns/conversation for LoCoMo, ∼493 for LongMemEval) with the identical BGE-M3, BM25, RRF, and MiniLM stack as C3 but without segmentation or evidence highlighting. This isolates CPP's structural contribution. *C3 (CPP)* is the full system: 124 segments/conversation (LoCoMo), 108 (LongMemEval), hybrid retrieval, reranking, evidence highlighting, and question-type-specific prompts. We selected flat retrieval as the primary baseline because it shares the same retrieval backbone as CPP (BGE-M3 embeddings, BM25 indexing, RRF fusion, and MiniLM reranking) while removing only the segmentation and evidence-highlighting components. This controlled design isolates CPP's structural contribution and avoids confounding it with retrieval-model differences. Direct comparisons against systems such as MemGPT, MemoryOS, or SimpleMem remain valuable but are complicated by differences in implementation availability, benchmark protocols, and underlying language models; we therefore leave them to future work.

Context-length asymmetry. At equal top-k, C2 delivers k turns as raw text (≈350 words at k=10 on LoCoMo) while C3 delivers up to a 1,200-word budget (≈3.4× more). To examine this possible confound, we consider three lines of evidence on whether the accuracy gain is primarily a function of context volume. First, session-level retrieval is not higher for CPP: on LoCoMo, Recall@k, Hit@k, and MRR are identical between C2 and C3 (Table 2), and on LongMemEval CPP's Recall@15 is in fact slightly lower than the baseline's (Table 4); in neither case does CPP surface more relevant material than flat retrieval. Second, C2's unbounded store exposes the model to more raw conversational content overall (≈601 turns per conversation versus CPP's ≈124 segments), so the baseline does not lack relevant information at the corpus level. Third, the component ablation (Table 3) adds evidence highlighting while holding the retrieved set fixed,

and this single change is associated with the largest accuracy increase (+26.3 pp on LongMemEval) without changing the amount of retrieved context. Together, these observations suggest that how context is organized and presented, rather than its volume alone, is a major driver of the gain. We do not claim that the volume effect is fully eliminated: a token-matched baseline, namely flat retrieval at $k \approx 34$ to equalize the word budget at $\approx$1,200 words, would provide a more direct test, and we identify it as an important item for future work (Section 8).

Computational cost. CPP's gains carry a processing cost. Beyond the shared retrieval stack, it adds one-time segmentation and deduplication at ingest and a query-time cross-encoder pass that scores each sentence of the k retrieved segments; this sentence-level scoring is the dominant added cost and scales with retrieved content, not store size. CPP also supplies a larger prompt than flat retrieval (the word budgets above), increasing input tokens proportionally. We note this trade-off explicitly; a controlled latency and token-throughput comparison on identical hardware is left for future work.

Code availability. The source code is available at `https://github.com/MudunuriManasa/Bounded-Conversational-Memory-with-Hybrid-Retrieval-and-Evidence-Highlighting`.

5 Results and Analysis

5.1 LoCoMo

Table 2 presents results across all three conditions. CPP achieves 81.4% overall MC accuracy, outperforming flat retrieval by 3.4 pp using 4.8× fewer indexed retrieval units. The results support H2: the largest gain is on multi-hop questions (+15.9 pp), followed by single-hop (+6.0 pp) and temporal (+3.1 pp). The small open-domain decrease (−0.5 pp, $\approx$4 questions) is consistent with the direction anticipated by H3. Adversarial accuracy reaches 0.998 across all conditions, including C1, indicating that this category is saturated in MC10 and does not measure memory quality.

The results also support H4: on LoCoMo, Recall@k, Hit@k, and MRR are identical between C2 and C3 (Table 2), so both conditions surface the same sessions, yet CPP answers more questions correctly. This pattern is consistent with a presentation-driven account, which we examine further in the component ablation below.

Table 2: Results on LoCoMo (`locomo_mc10`, n=1,986, Qwen2.5-14B). † C3 significantly outperforms C2 ($p<0.05$, McNemar's test) on all categories except open-domain.

Metric	C1	C2	C3	Δ(C3−C2)
MC Accuracy				
Overall	0.484	0.781	**0.814**†	+3.4 pp
Multi-hop	0.299	0.424	**0.583**†	+15.9 pp
Single-hop	0.330	0.727	**0.787**†	+6.0 pp
Temporal	0.354	0.531	**0.563**†	+3.1 pp
Open-domain	0.348	**0.848**	0.843	−0.5 pp
Adversarial	0.998	0.998	0.998	±0
Memory & Retrieval				
Retrieval units	0	601	**124**	4.8× fewer
Verbatim stor. recall	–	–	0.671	–
Recall@k	0.000	0.288	0.288	±0
Hit@k	0.000	0.775	0.775	±0
MRR	0.000	0.775	0.775	±0

MRR (Mean Reciprocal Rank): $\frac{1}{|Q|}\sum_q \frac{1}{\text{rank}_q}$; Recall@$k$: fraction of gold-relevant sessions in top-k; Hit@k: fraction of queries with at least one gold session in top-k.

Verbatim storage recall of 0.671 indicates that the exact gold string is absent from the store for 32.9% of questions, a coverage diagnostic rather than a hard ceiling, since the model can answer from paraphrased evidence. Misses concentrate in rare-entity open-domain questions that the cosine-threshold segmenter splits ambiguously, placing segmentation quality, rather than retrieval depth, as the primary remaining bottleneck.

5.2 LongMemEval

Table 3: Development-study component ablation on LongMemEval (n=500). Each configuration adds one CPP component cumulatively (topic-boundary segmentation, sentence-level evidence highlighting, question-type-aware prompting, and final k tuning), measured across sequential development iterations on the full benchmark. Because these are development runs rather than controlled held-out reruns with a single varied factor, they should be read as directional evidence of per-component contribution rather than a rigorous randomized ablation. A controlled single-factor ablation on held-out data is identified as future work.

Configuration	QA Acc	Δ
Session-level storage	0.133	–
+ Segment-level storage	0.267	+0.134
+ Evidence extraction	0.530	+0.263
+ Type-aware routing	0.600	+0.070
Full CPP (k=15, C3)	**0.606**	+0.473

Table 4 presents CPP's performance on LongMemEval at k=15: CPP reaches 0.606 versus the unbounded baseline's 0.360 (+24.6 pp) using 4.6× fewer indexed retrieval units. The component ablation (Table 3) associates the largest single gain with evidence highlighting, consistent with the presentation-oriented account developed above. Storage recall reaches 0.942, indicating that segmentation retains nearly all answer-bearing content in structured user-AI conversations, well above the 0.671 figure on LoCoMo. At k=15, fifteen raw turns introduce substantial noise into the context window and the unbounded baseline's accuracy drops sharply, whereas CPP's extraction pipeline filters noise before generation and remains comparatively stable. CPP improves over the baseline in every category except single-session assistant (−1.8 pp) and abstention (tied), indicating that the structured memory layer remains robust, and not merely competitive, as retrieval depth grows.

Table 4: Results on LongMemEval (n=500, k=15, free-form, Qwen2.5-14B judge). ⋆ CPP outperforms C2 (p<0.05). As retrieval depth increases, the unbounded baseline's accuracy decreases, whereas CPP remains comparatively stable, since its sentence-extraction pipeline filters noise before generation.

Metric	C2	C3	Δ
QA Accuracy (k=15)			
Overall	0.360	**0.606**⋆	+24.6 pp
Single-sess. (user)	0.359	**0.750**⋆	+39.1 pp
Single-sess. (asst.)	**0.750**	0.732	−1.8 pp
Single-sess. (pref.)	0.367	**0.600**⋆	+23.3 pp
Multi-session	0.223	**0.521**⋆	+29.8 pp
Temporal reasoning	0.252	**0.465**⋆	+21.3 pp
Knowledge-update	0.264	**0.667**⋆	+40.3 pp
Abstention	0.867	0.867	±0
Memory & Retrieval			
Retrieval units	493	**108**	4.6× fewer
Verbatim stor. recall	–	**0.942**	–
Recall@15	0.919	0.881	−3.8 pp

6 Discussion

6.1 Why CPP Outperforms Flat Retrieval

Segment-level storage addresses a key weakness of turn stores: an isolated turn such as *"That sounds great!"* carries almost no retrieval signal, whereas a five-turn topical segment embeds a richer semantic fingerprint. Because session-level retrieval is matched (identical on LoCoMo and slightly lower for CPP on LongMemEval), the results suggest that context presentation is a major driver of the observed accuracy gain rather than retrieval coverage. Evidence highlighting is the largest single component (Table 3, +26.3 pp on LongMemEval): the cross-encoder appends the highest-scoring line to the end of each segment, a high-attention position in the sense of Liu et al. (2024), without changing what was retrieved. This may explain why CPP maintains accuracy at higher retrieval depths while flat retrieval degrades under 15 raw turns of noise.

6.2 Bounded Memory in Practice

CPP uses 27–31% of its B=400 budget on both benchmarks, so the FIFO eviction policy is enforced architecturally but is not exercised in our experiments. Characterizing CPP under tighter budgets that force eviction, and isolating the contribution of the eviction policy itself, is left for future work.

6.3 Scope: Episodic Conversational Memory

Human conversational memory encompasses affective, multimodal, episodic, and semantic components. CPP intentionally focuses on the *episodic* conversational memory problem (retaining and recalling what was said in prior sessions) because this aspect can be evaluated directly using existing long-term memory benchmarks such as LoCoMo and LongMemEval. Affective memory (tracking emotional state over time) and multimodal memory (grounding recall in images, audio, or shared artifacts) are important dimensions of conversational memory that fall outside our current evaluation and represent natural directions for future work.

7 Conclusion

We presented the Context Persistence Protocol, a modular four-phase framework for bounded conversational memory evaluated across two complementary benchmarks. On LoCoMo, CPP achieves 81.4% MC accuracy, outperforming an unbounded flat retrieval baseline by 3.4 pp overall and 15.9 pp on multi-hop questions using 4.8× fewer indexed retrieval units. On LongMemEval at k=15, CPP surpasses the unbounded baseline by 24.6 pp (0.606 vs. 0.360), indicating that bounded structured memory remains robust, and not merely competitive, as retrieval depth increases: the unbounded system degrades under noise while CPP's extraction pipeline maintains accuracy. The central result is that, although session-level retrieval metrics are identical (LoCoMo) or slightly lower for CPP (LongMemEval), CPP answers substantially more

questions correctly. This suggests that context presentation, with evidence highlighting as the largest single contributor, is a major driver of accuracy that is largely independent of retrieval coverage. We frame boundedness as a production-scale design property rather than a claim exercised by these benchmarks, and we identify a token-matched baseline as the most important experiment for fully isolating the structural contribution.

8 Future Directions

A key next experiment is a token-budgeted baseline: running C2 at $k{\approx}34$ to match CPP's 1,200-word context would more cleanly separate context *volume* from *organization*, and a budget sweep ($B{\in}\{50, 100, 200\}$) on longer corpora would activate FIFO eviction to characterize the accuracy–compression trade-off. On the representation side, a learned boundary detector (Pan et al., 2025; Grootendorst, 2022) with an entity-coverage pass would target the 0.671 storage-recall bottleneck on LoCoMo, and resolving relative time expressions to ISO dates at ingest could further improve temporal recall (Wu et al., 2025).

Limitations

Our experiments were conducted on a single GPU and used English-only data, so we have not yet evaluated multilingual behavior. The LongMemEval judge belongs to the same model family as the generator (Qwen2.5-14B), and we did not calibrate its judgments against human annotations. The ablation in Table 3 is a development study rather than a controlled single-factor experiment on held-out data, so its component contributions should be read as directional. CPP uses only 27–31% of its $B{=}400$ budget on both benchmarks, and we have not yet tested the system under tighter budgets that would force eviction. Finally, because C3 delivers roughly $3.4\times$ more context words than C2 at equal top-k, a token-budgeted comparison remains the most important methodological gap to close in future work.

Ethical Considerations

CPP is designed to assist users of long-term conversational AI systems, but its deployment raises privacy considerations that should be acknowledged. Storing multi-session conversation history, even in bounded compressed form, involves retaining personal information that users may have shared incidentally during prior interactions. Any production deployment of CPP should be accompanied by clear user disclosure of what is stored, for how long, and how it can be deleted. The bounded storage design (a hard cap of 400 segments with FIFO eviction) is itself a partial mitigation: older information is automatically evicted rather than retained indefinitely. Bounded storage does not by itself solve privacy, however; production deployments would still require user-visible deletion controls, explicit retention policies, access safeguards, and consent mechanisms appropriate to the jurisdiction. All experiments in this paper use publicly available benchmark datasets (LoCoMo and LongMemEval) that contain only simulated or consented conversations; no real user data was collected or used.

References

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. M3-Embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024b. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. In *ECAI 2025 — 28th European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pages 2993–3000. IOS Press.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759, Boston, MA, USA. Association for Computing Machinery.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint*, arXiv:2203.05794.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 25961–25970, Suzhou, China. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. 2026. SimpleMem: Efficient lifelong memory for LLM agents. *arXiv preprint*, arXiv:2601.02553.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. 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)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

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

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. 2025. SeCom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

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

Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Henry Peng Zou, Shaowen Chen, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. 2026. GAM: Hierarchical graph-based agentic memory for LLM agents. *arXiv preprint*, arXiv:2604.12285.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, and 11 others. 2024. Qwen2.5 technical report. *arXiv preprint*, arXiv:2412.15115.