CoMMA: Conflict-Aware Read-Time Control for Conversational Memory QA

Zixiao Wang, Xingbo Du, Duzhen Zhang, Le Song
Mohamed bin Zayed University of Artificial Intelligence
{zixiao.wang,le.song}@mbzuai.ac.ae

Abstract

Memory-augmented assistants must answer questions from long-horizon conversations in which facts are not only missing, but also revised, corrected, or mutually inconsistent. Existing methods largely optimize retrieval, storage, or updating, and many controllers still treat failure primarily as insufficient evidence. We argue that conversational memory QA also requires an explicit model of *inconsistency*: when remembered claims disagree, a system should not simply retrieve more or guess, but may need to resolve versions, return a qualified answer, ask for clarification, or abstain. We introduce **CoMMA**, a transparent read-time controller with explicit conflict state for conversational memory QA. CoMMA extracts provenance-aware claims from retrieved memories, groups potentially incompatible claims into conflict clusters, infers lightweight question intent, and chooses among direct answer, qualified answer, clarification, and abstention. In an evaluation centered on LongMemEval-Oracle (500 examples), a LongMemEval conflict-focused diagnostic slice, and FactConsolidation-20, CoMMA performs best on the main LongMemEval answer-quality comparison, achieves the highest token F1 and action accuracy on the conflict-focused diagnostic slice, and ties the strongest write-time overwrite baseline on FactConsolidation-20. These results suggest that making conflict explicit at read time improves decision quality and exposes an auditable state under the evaluated conditions.

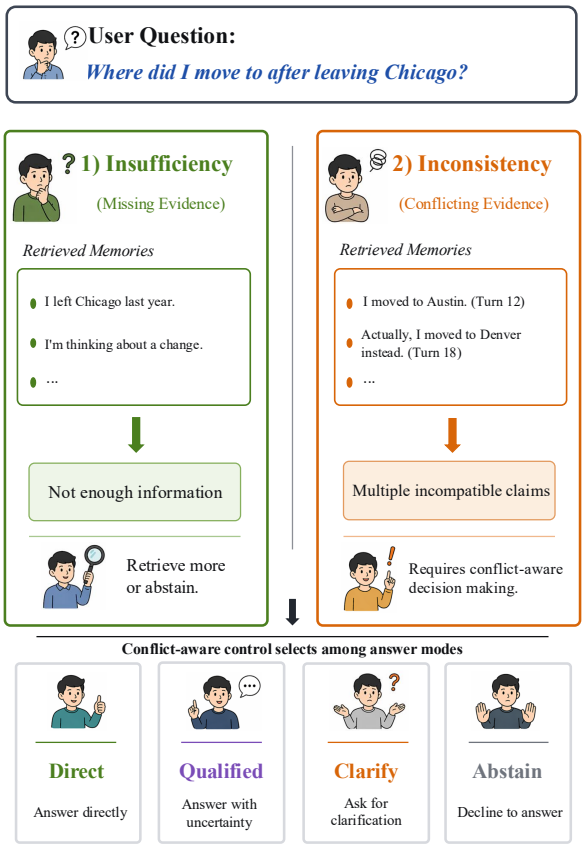


Figure 1: Insufficiency and inconsistency are distinct memory-QA failure modes: missing evidence calls for retrieval or abstention, while conflicting evidence requires conflict-aware answer-mode selection.

1 Introduction

Large language model assistants increasingly rely on external memory to support personalization and long-horizon interaction. Retrieval-augmented generation provides a standard way to couple generation with retrieved evidence (Lewis et al., 2020), and long-term conversational memory work shows that assistants can recover and reuse facts across sessions (Zhong et al., 2024; Packer et al., 2023; Xu et al., 2022a,b; Jang et al., 2023; Du et al., 2024). However, conversational memory is not only sparse; it is also dynamic, revisable, and sometimes self-contradictory (Bae et al., 2022; Lu et al., 2023; Ong et al., 2025; Wu et al., 2025a; Xu et al., 2025). In real multi-session use, users move cities, revise plans, correct earlier statements, or ask for historical rather than current facts. In such cases, failure does not arise only because evidence is miss-

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1925–1944
November 6-10, 2026 ©2026 Association for Computational Linguistics

ing. It also arises because retrieved evidence may disagree across time, sources, or reformulations.

Current memory systems do not cleanly separate these two failure modes. Much recent work improves memory organization, augmentation, compression, or retrieval efficiency (Salama et al., 2025; Ke et al., 2025; Li et al., 2025; Kang et al., 2025; Zhang et al., 2026; Chen et al., 2025; Lee et al., 2024; Xu et al., 2025). Another line makes retrieval more adaptive or conflict-aware in RAG settings, for example through active, selective, corrective, or self-reflective retrieval control, source reliability estimation, or conflict-aware evidence handling (Asai et al., 2024; Jiang et al., 2023; Jeong et al., 2024; Su et al., 2024; Yan et al., 2024; Baek et al., 2025; Hwang et al., 2025; Choi et al., 2025). In long-horizon conversational memory, however, the key difficulty is often internal inconsistency among remembered claims, together with the fact that different questions require different policies: a query may ask for the current state, a past state, the source of a claim, or whether the evidence remains disputed. Missing-evidence-oriented controllers such as MemR^3 explicitly track missing evidence, but they still treat the central control problem primarily as deciding whether to retrieve more or answer (Du et al., 2025a).

We argue that conversational memory QA requires separating *insufficiency* from *inconsistency*. Figure 1 illustrates this distinction: insufficiency is a lack of usable evidence, whereas inconsistency arises when retrieved memories provide incompatible but relevant claims and therefore requires an explicit answer-mode decision. When evidence is insufficient, the right response is often to retrieve more or abstain. When evidence is inconsistent, the right response may instead be to compare versions, resolve a conflict under an explicit basis, return a qualified answer, ask for clarification, or abstain rather than answer without surfacing uncertainty. Consider a simple example: if a user first says they moved to Chicago and later corrects this to the suburbs, a current-state query should prefer the revision, a historical query may require both versions, and a provenance-oriented query should surface the correction itself rather than collapse the memory into a single latest value. This distinction is especially important on benchmarks such as LoCoMo, LongMemEval, and MemoryAgentBench, which stress long-range memory, knowledge updates, abstention, and incremental multi-turn evaluation (Maharana et al., 2024; Wu et al., 2025a; Hu et al., 2025).

To operationalize this distinction, we introduce **CoMMA**, a transparent read-time controller with explicit conflict state for conversational memory QA. Here, *read-time* means that conflict handling is performed when answering the current question, without modifying the stored memory. CoMMA extracts provenance-aware claims from retrieved memories, groups potentially incompatible claims into conflict clusters, infers lightweight question intent, and chooses among direct answer, qualified answer, clarification, and abstention. Unlike write-time overwrite approaches, CoMMA preserves competing versions and resolves them at read time for the current query; unlike missing-evidence-only controllers, it treats conflicting evidence as an explicit conflict state rather than a side effect of retrieval. Our goal is not to replace write-time maintenance, but to complement it with query-conditioned conflict handling.

We evaluate CoMMA on LongMemEval and MemoryAgentBench-derived settings (Wu et al., 2025a; Hu et al., 2025). Under a fixed model setting, CoMMA performs best on LongMemEval-Oracle, achieves the highest token F1 and action accuracy on the conflict-focused diagnostic slice, and ties the strongest write-time overwrite baseline on FactConsolidation-20. These results suggest that explicitly modeling conflict state can improve not only final answer quality, but also the transparency and appropriateness of memory-aware decisions.

Our contributions are threefold:

- We formulate conversational memory QA as a control problem that must distinguish *insufficiency* from *inconsistency*, and we argue that the latter requires actions beyond retrieve-or-answer.
- We propose CoMMA, a lightweight, auditable read-time controller that combines claim extraction, conflict clustering, intent-aware resolution, and explicit answer-mode decisions.
- We provide evidence from LongMemEval-Oracle, a LongMemEval conflict-focused diagnostic slice, and FactConsolidation-20, together with appendix diagnostics for action behavior and backbone sensitivity.

2 Related Work

Memory for long-horizon conversational assistants. Retrieval-augmented generation (RAG) es-

tablished the standard coupling of generation with non-parametric evidence (Lewis et al., 2020), and recent surveys frame RAG and agent memory as system-level problems involving storage, retrieval, updating, compression, and reasoning over remembered experience (Gao et al., 2023; Du et al., 2025b; Zhang et al., 2024; Wu et al., 2025b). Conversational-memory systems further study how historical interactions should be extracted, structured, updated, and reused (Bae et al., 2022; Ke et al., 2025; Salama et al., 2025; Li et al., 2025; Kang et al., 2025; Chen et al., 2025). Benchmarks such as LoCoMo, LongMemEval, and MemoryAgentBench show that long-term assistants must handle temporal reasoning, knowledge updates, abstention, and incremental multi-turn evaluation rather than simple recall (Maharana et al., 2024; Wu et al., 2025a; Hu et al., 2025). CoMMA builds on this setting, but focuses on the read-time control problem that arises when retrieved memories are relevant yet mutually inconsistent.

Conflict-aware retrieval and evidence control. Another line of work studies how models should control evidence use during retrieval and generation. Knowledge-conflict and RAG-control studies show that model behavior changes when parametric memory, retrieved context, and document evidence disagree, or when sources differ in reliability (Mallen et al., 2023; Xu et al., 2024; Kortukov et al., 2024; Ren et al., 2025; Choi et al., 2025; Hwang et al., 2025). Self-RAG and MemR^3 adapt retrieval or reflection when evidence is missing or uncertain (Asai et al., 2024; Du et al., 2025a), while decoding and credibility-aware generation methods mediate conflicts by changing generation behavior (Shi et al., 2024; Wang et al., 2025; Yuan et al., 2024; Pan et al., 2024). CoMMA is related in spirit, but targets a different failure mode: conflicts internal to conversational memory itself, including revisions across turns, source-sensitive disagreements, and question-dependent interpretation of multiple remembered versions.

Memory updating and read-time conflict handling. A third line emphasizes write-time memory maintenance. Systems such as Mem0 and HiMem consolidate salient information or reconsolidate conflicting memories during storage and updating (Chhikara et al., 2025; Zhang et al., 2026). These approaches are complementary to CoMMA rather than direct replacements. Our goal is not to replace write-time updating, but to add a query-conditioned read-time controller that preserves competing versions and selects among direct answer, qualified answer, clarification, and abstention for the current query. The contribution is a systems integration of query-conditioned version handling and explicit answer-mode decisions, rather than a new learning algorithm. HiMem also uses query-conditioned, retrieval-triggered reconsolidation; CoMMA instead makes a query-local decision without updating the store, while retaining competing claims and their resolution basis in an inspectable trace.

3 Problem Formulation and CoMMA

3.1 Task Setting and Failure Modes

We study conversational memory question answering over a multi-session dialogue history $\mathcal{H} = \{h_1, \ldots, h_n\}$ and a current user question q. A retriever first returns a small set of candidate memory turns $\mathcal{T} = R(q, \mathcal{H})$. The controller then produces a final response $\hat{y}$, an answer mode a, and an explicit conflict trace z:

$$(\hat{y}, a, z) = \pi(q, \mathcal{T}).$$

The answer mode is one of DIRECT, QUALIFIED, CLARIFY, or ABSTAIN.

Our key premise is that conversational memory QA fails under two distinct conditions. *Insufficiency* means that retrieved memory does not provide adequate support for the query: relevant evidence is absent, too weak, or too indirect. *Inconsistency* means that retrieved memory contains multiple relevant but mutually incompatible claims, for example different values for the same entity–slot pair. These failure modes are related but not equivalent. When evidence is insufficient, the right action is often additional retrieval or abstention. When evidence is inconsistent, the right action may instead be to compare versions, return a qualified answer, ask for clarification, or explicitly preserve disagreement rather than answer without surfacing uncertainty. This distinction is especially important in long-horizon benchmarks such as LongMemEval and MemoryAgentBench, which emphasize knowledge updates, abstention, and multi-turn memory evaluation (Wu et al., 2025a; Hu et al., 2025).

A second difficulty is that the meaning of a correct answer is query-dependent. A current-state question typically prefers the latest justified update, a historical question may require preserving older values, and a provenance-oriented question should

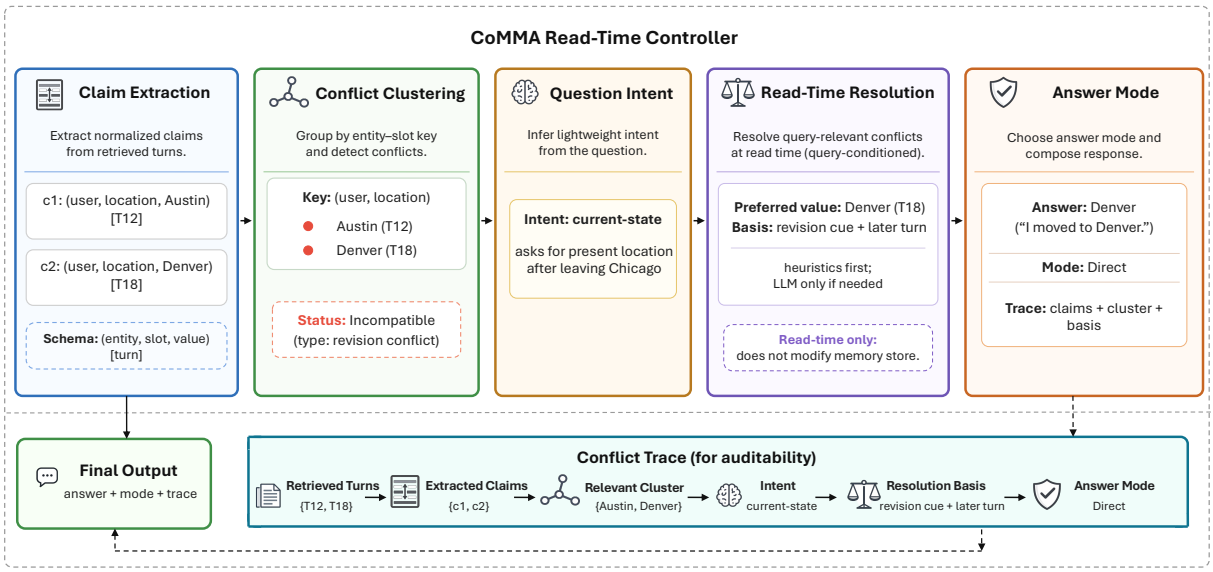


Figure 2: Overview of CoMMA. The read-time controller converts retrieved conversational memories into provenance-aware claims, forms conflict clusters, conditions resolution on question intent, and outputs an answer mode together with an explicit conflict trace, without modifying the underlying memory store.

surface who said what and when. CoMMA therefore models failure as a function of both *memory state* and *question intent*, rather than collapsing all retrieved memory into a single latest-value store.

3.2 Overview

CoMMA is a *read-time* controller layered on top of an existing memory store. It does not modify the underlying store during inference. Given a question q and retrieved turns $\mathcal{T}$, it (i) extracts normalized claims from raw turns, (ii) groups potentially incompatible claims into conflict clusters, (iii) infers a lightweight question intent, (iv) resolves or preserves query-relevant conflicts at read time, using heuristics first and optional LLM refinement only when needed, and (v) selects an answer mode together with an explicit conflict trace. Figure 2 summarizes this pipeline.

This design serves two goals. First, it separates missing-evidence cases from conflict cases, rather than treating both as generic uncertainty. Second, it keeps the controller transparent: every answer is tied to a small set of extracted claims, relevant conflict clusters, and a concrete decision basis. Compared with missing-evidence-oriented controllers such as MemR^3, CoMMA explicitly represents inconsistent evidence rather than treating all uncertainty as missing evidence (Du et al., 2025a). Compared with write-time overwrite strategies, it preserves multiple remembered versions and resolves them only for the current query.

3.3 Claim Extraction and Conflict Clustering

From each retrieved turn $t \in \mathcal{T}$, CoMMA extracts zero or more normalized claim records:

$$c = (e, s, v, \tau, p, \gamma),$$

where e is the entity or subject, s is the slot or property, v is the normalized value, τ captures temporal information such as timestamp, temporal scope, or revision cue, p stores provenance metadata such as turn id and speaker, and γ is an optional confidence score. In implementation, this step combines lightweight surface heuristics with optional LLM-assisted normalization, so that downstream modules operate over compact structured records rather than raw turns.

CoMMA then groups claims by canonicalized entity–slot keys:

$$g_{e,s} = \{c_j \mid \text{entity}(c_j) = e \ \wedge \ \text{slot}(c_j) = s\}.$$

Within each group, claims are treated as *compatible* when they normalize to the same value or one clearly refines the other; otherwise they form a candidate conflict set. Each cluster stores its candidate values, supporting turn ids, temporal markers, and explicit revision cues such as "actually", "now", or "used to". Only clusters that are relevant to the current question are passed to later stages, which keeps reasoning local and interpretable.

This claim-first representation is intentionally lightweight. The controller reasons over a small

set of typed memory statements, while the original retrieved turns remain available as provenance. As a result, downstream decisions can be inspected in terms of *which claims conflicted*, *which ones were preferred*, and *why*.

3.4 Intent-Aware Read-Time Resolution

CoMMA next infers a lightweight intent label from the question surface form, such as CURRENT, HISTORICAL, PROVENANCE, DISPUTE, TEMPORAL, COMPARISON, or UNKNOWN. These labels determine what counts as an appropriate resolution policy. For example, a current-state question often prefers a revised value, whereas a historical or provenance-oriented question may require preserving multiple versions.

For each query-relevant conflict cluster, the resolver predicts one of three statuses: RESOLVED, UNRESOLVED, or IRRELEVANT. Resolution proceeds in two stages. First, a heuristic stage exploits explicit correction cues, temporal order, speaker metadata, and simple dominance rules to select a preferred claim when the basis is clear. Second, if a relevant cluster remains ambiguous, a compact LLM resolver compares candidate claims under the inferred intent and either returns a preferred value together with a basis or leaves the cluster unresolved.

Importantly, this process is read-time only. CoMMA does not overwrite, delete, or merge stored memory during inference. This makes the controller complementary to write-time updating: historical and source-sensitive questions can preserve multiple remembered versions even when a current-state question would prefer one of them.

3.5 Answer Modes and Conflict Trace

After resolution, CoMMA chooses an answer mode rather than forcing all cases into direct response. This design is aligned with uncertainty-aware generation, which motivates surfacing uncertainty rather than always producing a single unqualified answer when evidence is ambiguous (Kuhn et al., 2023). We use four modes. DIRECT returns a single answer when evidence is sufficient and one claim clearly dominates. QUALIFIED returns an answer together with a brief uncertainty or provenance qualifier when a value is usable but the conflict state should remain visible. CLARIFY asks the user for disambiguation when the correct response depends on missing intent or unresolved version choice. ABSTAIN declines to answer when evidence is insufficient or the controller cannot safely resolve the relevant conflict state.

Alongside the final response, CoMMA emits an explicit conflict trace. The trace records the retrieved turns $\mathcal{T}$, the extracted claims $\mathcal{C}$, the relevant conflict clusters $\mathcal{G}$, the inferred intent $I(q)$, the resolution basis B, and the final answer mode a. This trace is not merely a debugging artifact. It is part of the method: it makes decisions auditable, supports action-level evaluation beyond answer accuracy, and exposes whether the system resolved a conflict, preserved it, asked for clarification, or abstained. In this sense, CoMMA extends transparent retrieval control from missing-evidence tracking to explicit conflict-state handling.

4 Experimental Setup

4.1 Datasets

We center our evaluation on benchmark-derived settings represented in a shared evaluation format. The main text reports **LongMemEval-Oracle** for the primary comparison, a **LongMemEval conflict-focused diagnostic slice** for explicit conflict diagnostics, and **FactConsolidation-20** for external validation. We report a labeled **LongMemEval action diagnostic slice** in the appendix because, in the current main setting, it primarily informs action behavior rather than the paper's core explicit-conflict claim.

Our main table uses **LongMemEval-Oracle**, the official 500-example oracle-retrieval variant of LongMemEval (Wu et al., 2025a), whose histories contain only the evidence sessions relevant to each question. In the adapted records, those evidence sessions are flattened into a single turn sequence while preserving session metadata. This setting reduces benchmark-side retrieval ambiguity, but all methods still answer through the same top-$k = 5$ read-time retrieval and answering pipeline over the adapted evidence-session turns.

The **conflict-focused diagnostic slice** contains 50 manually labeled examples concentrated around explicit contradictions, revision chains, historical–current mismatches, and stale-value cases. The appendix **action diagnostic slice** contains 34 labeled examples designed to expose whether the preferred behavior is to answer, clarify, or abstain. Both diagnostic slices are targeted subsets derived from adapted LongMemEval examples rather than new standalone benchmarks.

For external validation, we use

Method	Answer (%)		Control (%)		Efficiency		
	EM	Tok. F1	Act. Acc.	Over-Ab.	Calls	Tokens	Cost ($)
CoMMA	**29.8**	**48.8**	**87.0**	**10.4**	2.5	3,034	0.0016
Full context	28.4	44.7	79.2	17.2	2.0	9,150	0.0041
Write-time overwrite	26.4	41.7	74.4	21.0	2.0	7,725	0.0035
MemR^3-style	26.2	41.5	75.2	18.0	2.9	4,305	0.0021
No-conflict controller	23.4	37.9	63.6	31.4	2.0	2,711	0.0014
RAG only	14.8	23.3	37.8	56.0	2.0	1,029	0.0006
Recency only	7.8	16.3	26.4	70.4	2.0	2,636	0.0012

Table 1: Main results on LongMemEval-Oracle (500 examples). Values are percentages except calls, tokens, and estimated cost. Lower over-abstention indicates fewer incorrect abstentions on answerable examples.

FactConsolidation-20, a 20-example slice adapted from MemoryAgentBench (Hu et al., 2025). This slice tests whether the same read-time conflict-handling pattern transfers beyond LongMemEval and into a more consolidation-focused setting.

4.2 Baselines

We compare CoMMA against six baselines implemented in the same framework. The **Full context** baseline answers from all turns without explicit conflict state. **RAG only** answers from the single top retrieved memory, while **Recency-only** uses the most recent turns. The **No-conflict controller** shares CoMMA's retrieval and claim extraction but removes conflict clustering. **MemR^3-style** approximates a missing-evidence-only retrieve-more controller, and **Write-time overwrite** collapses extracted claims into a latest-value memory state before answering. Detailed operational definitions are given in Appendix D.

On the conflict-focused slice, we additionally evaluate **Timestamp-aware RAG**, which uses lexical and recency-based retrieval over the same candidate pool, top-$k = 5$ timestamped snippets, and an update-aware answering prompt. Timestamps are available to all methods, including the recency and overwrite baselines. This comparison tests whether the specified timestamp-aware pipeline suffices without CoMMA's explicit conflict state.

4.3 Metrics

We report answer exact match (EM) and token-level F1 as the primary answer-quality metrics. Following the evaluation protocol, lexical answer metrics are computed only on examples whose gold expected action is `answer` and that provide a non-empty reference answer. Examples labeled `clarify` or `abstain` are excluded from lexical answer scoring but remain part of action-level evaluation.

Because CoMMA makes explicit control decisions, we also report action accuracy over {`answer`, `clarify`, `abstain`}. Both `direct` and `qualified` outputs are mapped to the `answer` action class.

When conflict annotations are available, we compute conflict precision, conflict recall, and conflict F1. These metrics evaluate emitted structured conflict objects. Methods that emit no such objects receive zero under this protocol, which does not imply an absence of implicit conflict reasoning. CoMMA's absolute conflict F1 is low, so this metric should be interpreted as limited recovery of explicit, auditable conflict state rather than a symmetric comparison of latent reasoning ability.

We also report over-abstention rate in the main tables. For efficiency comparisons, we summarize average API calls, average total tokens, and average estimated cost. Additional diagnostic metrics, including stale-answer analysis and supplementary backbone comparisons, are deferred to the appendix.

4.4 Implementation Details

All main-text experiments use gpt-4.1-mini as the shared backbone model. We keep the backbone fixed across methods so that the comparison isolates controller behavior rather than model variation.

For retrieval-based methods, the main-text setting uses a target retrieval budget of top-$k = 5$. Non-retrieval baselines such as `full_context` and `write_time_overwrite` consume all turns by design, and controller-style methods may request a broader candidate pool internally before finalizing the memories used downstream. We therefore treat fairness at the level of the down-

stream memory budget rather than identical low-level retrieval calls.

We report one run per method in the main tables; the appendix provides additional details on the evaluation slices, action diagnostics, supplementary backbone comparisons, and cost estimation.

5 Results

Tables 1–3 show a selective but consistent pattern. CoMMA performs best on LongMemEval-Oracle, achieves the highest token F1 and action accuracy on the conflict-focused diagnostic slice, and ties the strongest write-time overwrite baseline on FactConsolidation-20. At the same time, the results are not uniformly dominated by a single method: overwrite-style and full-context baselines remain competitive on some raw answer metrics. This is precisely why we separate answer quality from conflict-aware control behavior.

5.1 Main Results on LongMemEval-Oracle

Table 1 reports the main comparison on all 500 adapted LongMemEval-Oracle examples. CoMMA achieves the strongest overall answer profile, with 29.8 EM and 48.8 token F1, improving over Full context at 28.4 EM and 44.7 token F1, Write-time overwrite at 26.4 EM and 41.7 token F1, and the missing-evidence-only MemR^3-style baseline at 26.2 EM and 41.5 token F1. CoMMA also attains the highest action accuracy at 87.0 while keeping over-abstention lowest at 10.4.

The main table also shows that CoMMA does not win simply by using the largest inference budget. Although it is more expensive than lightweight retrieval-only baselines, it remains substantially cheaper than Full context and Write-time overwrite while outperforming both on answer quality. Full context consumes the largest average token budget yet trails CoMMA, consistent with prior findings that longer contexts are not always used reliably and may harm reasoning performance (Liu et al., 2024; Levy et al., 2024); this suggests that exposing the full conversation is not an adequate substitute for conflict-aware control. Similarly, Write-time overwrite lags CoMMA on the main answer and control metrics despite much higher token usage, suggesting that collapsing memory into a single latest-value view can hurt when the query requires version-sensitive interpretation rather than simple recency.

Taken together, the LongMemEval-Oracle table supports our main controller-level claim: explicit read-time conflict modeling improves the primary comparison in the main evaluation setting.

5.2 Conflict-Focused Diagnostic Evaluation

Table 2 reports the LongMemEval conflict-focused diagnostic slice. Here the main takeaway is not raw EM alone. Write-time overwrite attains the best EM at 31.3, but CoMMA achieves the highest token F1, the highest action accuracy, and the only non-zero conflict F1. This pattern is consistent with CoMMA's design goal: improving explicit conflict handling rather than uniformly maximizing lexical answer metrics.

Timestamp-aware RAG obtains 20.8 EM, 45.4 token F1, and 82.0 action accuracy, with 14.0 over-abstention. CoMMA improves the first three metrics by 6.3, 11.8, and 10.0 percentage points, respectively, and reduces over-abstention to 4.0. These results support benefits beyond the tested timestamp-aware retrieval and prompting strategy on this slice, without isolating the independent effect of each controller component.

The conflict slice also clarifies the difference between read-time and write-time handling. Write-time overwrite is strong on resolved latest-value cases, but it never surfaces conflict explicitly. CoMMA instead trades a small amount of raw EM for markedly better conflict visibility and stronger action behavior, while remaining much cheaper than full-context answering.

5.3 External Validation on FactConsolidation-20

Table 3 evaluates the same methods on FactConsolidation-20. CoMMA ties Write-time overwrite at 95.0 EM and 95.0 token F1, while outperforming the remaining read-time baselines. This small external check suggests that the CoMMA pattern is not confined to LongMemEval and that read-time conflict handling can remain effective in a more consolidation-focused setting, while still acknowledging that overwrite-style maintenance can be a strong strategy on some tasks.

6 Discussion

The results are consistent with the hypothesis that CoMMA benefits from preserving competing memory versions until answer time and resolving them under the current question, rather than collapsing memory too early or treating all uncertainty

Method	EM	Tok. F1	Act. Acc.	Conf. F1	Over-Ab.
CoMMA	27.1	**57.2**	**92.0**	**5.3**	**4.0**
Write-time overwrite	**31.3**	51.3	78.0	0.0	16.0
MemR^3-style	29.2	49.5	88.0	0.0	6.0
Full context	27.1	48.5	80.0	0.0	14.0
No-conflict controller	25.0	45.6	76.0	0.0	18.0
Timestamp-aware RAG	20.8	45.4	82.0	0.0	14.0
RAG only	16.7	26.7	40.0	0.0	54.0
Recency only	6.3	17.7	34.0	0.0	62.0

Table 2: Conflict-focused diagnostic evaluation on LongMemEval (50 examples). Values are percentages; EM and token F1 are computed on answerable examples only. Conf. F1 measures structured conflict outputs; zero for a method emitting none is not evidence of absent implicit conflict reasoning.

Method	EM	Tok. F1
CoMMA	**95.0**	**95.0**
Write-time overwrite	**95.0**	**95.0**
Full context	45.0	45.0
MemR^3-style	35.0	35.0
No-conflict controller	35.0	35.0
RAG only	20.0	22.2
Recency only	5.0	6.0

Table 3: External validation on FactConsolidation-20. All methods attain 100.0 action accuracy, so the table reports the answer-quality comparison.

as generic missing evidence. On LongMemEval-Oracle, this yields the strongest overall answer profile. On the conflict slice, the gains are more diagnostic than absolute: CoMMA does not maximize every lexical answer metric, but it is the only method that makes conflict explicit while also attaining the highest action accuracy.

Table 4 illustrates the mechanism behind these gains on a representative revision-sensitive example. The key failure mode is not missing evidence, but stale evidence: the retrieved memories contain two relevant location claims, and the controller must recognize that they conflict before deciding which one should answer the current-state question. CoMMA exposes this intermediate conflict state, resolves it using the later correction and revision cue evidence, and records the answer basis. Missing-evidence-only and no-conflict baselines instead tend to return the stale earlier value because they never represent the revision itself as an explicit control state. Additional qualitative traces are deferred to Appendix F.

The comparison with Write-time overwrite also clarifies the scope of the claim. Write-time overwrite remains strong on consolidation-focused settings such as FactConsolidation-20 and on some stale-value cases, which suggests that read-time conflict handling should be viewed as complementary to write-time maintenance rather than as its replacement. Likewise, the appendix action slice shows that not every diagnostic slice yields the same explicit-conflict signal; some are better interpreted as action-level diagnostics than as conflict-detection benchmarks. Overall, retrieval and write-time maintenance remain necessary, while CoMMA makes the version-choice policy explicit once incompatible evidence has been retrieved.

7 Conclusion

We presented CoMMA, a transparent read-time controller for conversational memory QA that explicitly distinguishes *insufficiency* from *inconsistency*. Unlike missing-evidence-oriented or overwrite-based approaches, CoMMA extracts provenance-aware claims, clusters conflicting memory versions, infers lightweight question intent, and chooses among direct answer, qualified answer, clarification, and abstention. Across LongMemEval-Oracle, a LongMemEval conflict-focused diagnostic slice, and FactConsolidation-20, CoMMA performs best on the main answer-quality comparison, achieves the highest token F1 and action accuracy on the conflict-focused diagnostic slice, and ties the strongest write-time overwrite baseline on external validation. These results suggest that treating conflicting evidence as an explicit read-time state improves not only answer quality, but also the transparency and appropriateness of memory-aware decisions. These findings concern read-time control under the evaluated oracle and diagnostic settings; broader end-to-end retrieval and model-family generalization remain to be established.

Component	Trace
Question	Where did Rachel move to after her recent relocation?
Retrieved turns	Earlier memory: Rachel moved to Chicago. Later correction: Rachel clarified that she moved to the suburbs.
Extracted claims	`location=Chicago` from the earlier turn; `location=the suburbs` from the later correction.
Conflict cluster	Same entity–slot pair with incompatible values: `Rachel/location = {Chicago, the suburbs}`.
Intent and resolution	Current-state question. CoMMA prefers the later correction using revision cue evidence and later-turn priority.
Answer mode and output	DIRECT: the suburbs.
Baseline failure	Missing-evidence-only and no-conflict baselines return the stale earlier value, Chicago.

Table 4: Qualitative conflict trace for a revision-sensitive example. CoMMA exposes the conflicting claims, resolves the current-state query using the later correction, and records the answer basis.

Limitations

Our work has several limitations. First, our evaluation focuses on benchmark-derived settings, including LongMemEval-Oracle, diagnostic slices, and FactConsolidation-20. The oracle setting reduces retrieval noise relative to realistic deployment conditions, so our findings primarily concern read-time control under the evaluated conditions rather than end-to-end retrieval robustness. Second, main-text experiments use a single backbone model. Supplementary comparisons remain limited in scale and show variation across settings; broader model-family coverage would strengthen generalizability. Third, CoMMA relies on lightweight claim extraction and heuristic conflict clustering, which may miss implicit, cross-slot, or compositional conflicts. Its low absolute conflict F1 also indicates limited recovery of annotated conflict objects. This metric evaluates explicit structured outputs: zero scores for methods that emit no such objects do not imply an absence of implicit conflict reasoning. Finally, CoMMA is a read-time controller intended to complement, not replace, write-time memory maintenance; integrating the two remains an important direction for future work.

Ethical Considerations

Our experiments use only public benchmark data and do not involve newly collected user information. However, conversational memory assistants can raise privacy and overconfidence risks in deployment. CoMMA is designed to help mitigate overconfidence arising from conflicting memories by making conflicts explicit and enabling clarification or abstention, but it does not replace privacy-preserving storage, access control, or broader safety mechanisms. We advise against deploying such systems as sole decision makers in high-stakes settings without additional safeguards.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Sanghwan Bae, Donghyun Kwak, Soyoung Kang, Min Young Lee, Sungdong Kim, Yuin Jeong, Hyeri Kim, Sang-Woo Lee, Woomyoung Park, and Nako Sung. 2022. Keep me updated! memory management in long-term conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3769–3787, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ingeol Baek, Hwan Chang, ByeongJeong Kim, Jimin Lee, and Hwanhee Lee. 2025. Probing-rag: Self-probing to guide language models in selective document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*.

Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. 2025. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 755–773, Abu Dhabi, UAE. 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. *arXiv preprint arXiv:2504.19413*.

Eunseong Choi, June Park, Hyeri Lee, and Jongwuk Lee. 2025. Conflict-aware soft prompting for retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26981–26995, Suzhou, China. Association for Computational Linguistics.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. 2025a. Memr^3: Memory retrieval via reflective reasoning for LLM agents. *arXiv preprint arXiv:2512.20237*.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025b. Rethinking memory in AI: Taxonomy, operations, topics, and future directions. *arXiv preprint arXiv:2505.00675*.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. 2024. PerLTQA: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing*, pages 152–164, Bangkok, Thailand. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. Evaluating memory in LLM agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*.

Jeongyeon Hwang, Junyoung Park, Hyejin Park, Dongwoo Kim, Sangdon Park, and Jungseul Ok. 2025. Retrieval-augmented generation with estimation of source reliability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34279–34303, Suzhou, China. Association for Computational Linguistics.

Jihyoung Jang, Minseong Boo, and Hyounghun Kim. 2023. Conversation chronicles: Towards diverse temporal and relational dynamics in multi-session conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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*, pages 25961–25970, Suzhou, China. Association for Computational Linguistics.

Cai Ke, Yiming Du, Bin Liang, Yifan Xiang, Lin Gui, Zhongyang Li, Baojun Wang, Yue Yu, Hui Wang, Kam-Fai Wong, and Ruifeng Xu. 2025. Flexibly utilize memory for long-term conversation via a fragment-then-compose framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21119–21136, Suzhou, China. Association for Computational Linguistics.

Evgenii Kortukov, Alexander Rubinstein, Elisa Nguyen, and Seong Joon Oh. 2024. Studying large language model behaviors under context-memory conflicts with real documents. In *First Conference on Language Modeling*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26396–26415. PMLR.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K"uttler, Mike Lewis, Wen-tau Yih, Tim Rockt"aschel, 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.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2025. Hello again! LLM-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5259–5276, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023.

MemoChat: Tuning LLMs to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

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.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seungwon Hwang, Dongha Lee, and Jinyoung Yeo. 2025. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8631–8661, Albuquerque, New Mexico. 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*.

Ruotong Pan, Boxi Cao, Hongyu Lin, Xianpei Han, Jia Zheng, Sirui Wang, Xunliang Cai, and Le Sun. 2024. Not all contexts are equal: Teaching LLMs credibility-aware generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19844–19863, Miami, Florida, USA. Association for Computational Linguistics.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2025. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *Proceedings of the 31st International Conference on Computational Linguistics*.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33136–33152, Suzhou, China. Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. AdaCAD: Adaptively decoding to balance conflicts between contextual and parametric knowledge. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025b. From human memory to AI memory: A survey on memory mechanisms in the era of LLMs. *arXiv preprint arXiv:2504.15965*.

Jing Xu, Arthur Szlam, and Jason Weston. 2022a. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8565, Miami, Florida, USA. Association for Computational Linguistics.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-MEM: Agentic memory for LLM agents. *arXiv preprint arXiv:2502.12110*.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022b. Long time no see! open-domain conversation with long-term persona memory. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Xiaowei Yuan, Zhao Yang, Yequan Wang, Shengping Liu, Jun Zhao, and Kang Liu. 2024. Discerning and resolving knowledge conflicts through adaptive

decoding with contextual information-entropy constraint. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Ningning Zhang, Xingxing Yang, Zhizhong Tan, Weiping Deng, and Wenyong Wang. 2026. Himem: Hierarchical long-term memory for LLM long-horizon agents. *arXiv preprint arXiv:2601.06377*.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Additional Experimental Details

Unless otherwise noted, the appendix uses the same experimental setting as the main text: gpt-4.1-mini, one run per method, and top-$k = 5$ for retrieval-limited methods. Table 5 summarizes the evaluation settings used in the paper. Cost values are estimated offline from token counts and the corresponding model pricing.

We make only setting-specific empirical claims in the main paper: CoMMA performs best on LongMemEval-Oracle, achieves the highest token F1 and action accuracy on the LongMemEval conflict-focused diagnostic slice, and ties the strongest write-time overwrite baseline on FactConsolidation-20. A supplementary gpt-5.4-mini comparison is reported later in the appendix rather than treated as part of the main experimental setting.

B Dataset Details

The evaluation reported here is intentionally centered on benchmark-derived settings rather than all benchmark variants. Its purpose is to isolate the behaviors CoMMA is designed to model: version-sensitive answering, conflict handling, clarification, and abstention under a shared evaluation format.

B.1 Unified Example Format

All reported settings are represented in a unified example format so that every method is evaluated through the same pipeline. Each example contains the current question, a turn-level conversation, and optional supervision for the reference answer, expected action, expected conflicts, and question intent. This shared format supports both answer-oriented evaluation and controller-oriented action/conflict evaluation while preserving provenance to the original turns.

B.2 LongMemEval-Oracle

The main table uses LongMemEval-Oracle, the official 500-example oracle-retrieval variant of LongMemEval whose histories contain only the evidence sessions relevant to each question. In the adapted records, those evidence sessions are flattened into a single turn sequence while preserving session metadata. The evaluation follows the same unified format, backbone, top-$k = 5$ retrieval setting, and method implementations across all methods.

B.3 Diagnostic Slices

The two diagnostic slices are derived from the same adapted LongMemEval pool using manual labels that specify the expected action, the relevant conflict object, and, when needed, the intended temporal reading of the question. The conflict-focused diagnostic slice remains in the main text because it is the clearest diagnostic of explicit conflict handling. The action diagnostic slice is reported only in the appendix because, under the current main setting, no method attains non-zero conflict F1 on that slice.

The action slice contains 34 examples, including answer, abstain, and clarify cases, and is best read as an action-level diagnostic rather than as a conflict-detection benchmark. The conflict slice contains 50 examples centered on explicit contradictions, revision chains, historical–current mismatches, and stale-value cases.

B.4 Manual Labeling

The LongMemEval diagnostic slices are manually labeled subsets derived from the same adapted source pool. We use these labels to define the preferred high-level action for each example and, on the conflict-focused slice, to identify the relevant conflict object and its status. Two annotators independently labeled all retained examples in the two diagnostic slices. The action slice contains 34 examples and is used to distinguish among answer, clarify, and abstain. The conflict-focused slice contains 50 examples and is used to identify explicit value conflicts, revision chains, and stale-value cases.

Setting	Size	Source	Role in the paper
LongMemEval-Oracle	500	Adapted LongMemEval	Main answer-quality and controller comparison
LongMemEval conflict-focused diagnostic	50	Adapted LongMemEval + manual labels	Main-text diagnostic for explicit conflict handling
LongMemEval action diagnostic	34	Adapted LongMemEval + manual labels	Appendix action-level diagnostic of answer / clarify / abstain behavior
FactConsolidation-20	20	Adapted MemoryAgentBench	Main-text external validation

Table 5: Evaluation settings used in the paper.

For each example, annotators labeled the expected action and, when needed, the question intent. On the conflict-focused slice, annotators additionally labeled a structured conflict object consisting of the subject, slot, entity key (if applicable), the competing values, conflict status (resolved or unresolved), the resolved value when applicable, and a coarse conflict type. Before labeling the reported slices, the annotators completed a small pilot round on held-out examples not included in the final tables. This pilot was used to clarify the boundary between clarify and abstain, to standardize the treatment of revision cues such as "actually" and "instead," and to refine the conflict-type definitions. After this calibration step, the guidelines were frozen and both annotators labeled the retained examples independently.

After independent labeling, the retained examples were reconciled into a single final label set used by the experiments. These labels are intended to support targeted diagnostic analysis rather than to define a new benchmark release, so we keep our empirical claims slice-specific rather than benchmark-global.

B.5 FactConsolidation-20

For external validation, we use a 20-example slice derived from MemoryAgentBench. We convert each source item into one or more question-level examples by splitting bundled contexts and QA pairs into a shared fact-list conversation plus per-question labels. The slice is intentionally small, but it is useful for testing whether the same read-time conflict-handling pattern transfers beyond LongMemEval and into a more consolidation-focused setting.

C Metrics and Labels

We evaluate each method along three dimensions: answer quality, control behavior, and conflict-state traceability. All metrics are computed after matching prediction rows to dataset examples by example id.

C.1 Evaluation Subsets

We distinguish four evaluation subsets. Matched examples are prediction rows whose example id appears in the evaluation set. Answerable examples are matched examples with a non-empty reference answer and gold expected action ANSWER. Action-evaluated examples are matched examples with an expected action label. Gold conflicts are the annotated conflict objects attached to matched examples.

Answer EM and token F1 are averaged over answerable examples only. Action metrics are averaged over action-evaluated examples. Conflict metrics are computed against gold conflict annotations when available. Efficiency metrics are reported as per-example averages. This separation is important for the diagnostic slices, where the preferred behavior may be to answer, clarify, or abstain rather than always produce a reference string.

C.2 Surface-Form Answer Metrics

We report exact match (EM) and token-level F1 after lightweight normalization that lowercases text and retains only alphanumeric tokens. Let $\mathrm{Norm}(\cdot)$ denote this normalization. Then

$$\mathrm{EM}(y, \hat{y}) = \mathbf{1}[\mathrm{Norm}(y) = \mathrm{Norm}(\hat{y})].$$

Token F1 is the standard overlap F1 between normalized token multisets. These metrics are intentionally conservative and do not use paraphrase matching or judge-based grading.

C.3 Conflict and Action Metrics

Conflict metrics compare predicted conflict clusters with annotated gold conflicts. Each conflict is keyed by subject, slot, and, when available, entity key. We report conflict precision, recall, and F1 from true-positive, false-positive, and

false-negative counts, together with supplementary diagnostics such as resolved-conflict accuracy, unresolved-conflict recall, and stale-answer rate.

For action evaluation, direct and qualified outputs are mapped to ANSWER, while clarify and abstain remain separate action classes. We report action accuracy, over-guess rate, and over-abstention rate.

C.4 Trace Checks

To make the transparency claim operational, we include two lightweight structural checks. Support-text faithfulness verifies that an extracted claim's recorded support text appears in its evidence text. Answer-citation faithfulness verifies that cited claim ids in the final answer correspond to extracted claims in the same example output. These checks do not establish semantic entailment, but they catch structural failures that would weaken the auditability of the controller trace.

D Baselines and Implementation Details

All methods are implemented within the same experimental framework and are scored by the same evaluator. Method differences are therefore localized to what memory is exposed downstream and whether explicit conflict state is maintained.

All methods are run in the same evaluation framework and use the same backbone model, output schema, evaluator, and cost accounting. Retrieval-limited methods use a target downstream memory budget of top-$k = 5$. Full-context and write-time-overwrite baselines consume all turns by design, while controller-style methods may inspect a broader candidate pool before selecting the memories passed to downstream answering. We therefore enforce comparability at the level of downstream memory exposure and shared evaluation, rather than requiring identical low-level retrieval calls.

E Additional Results

E.1 Supplementary Action Diagnostic Slice

Table 8 reports the appendix action slice. Under the current main setting, no method attains non-zero conflict F1 on this slice. We therefore read it primarily as an action-level diagnostic. Under that interpretation, CoMMA attains the highest action accuracy, the highest token F1, and the lowest over-abstention rate, even though full context remains strongest on raw EM.

E.2 Supplementary gpt-5.4-mini Comparison

We include an auxiliary gpt-5.4-mini comparison to assess whether the qualitative patterns above are tied to a single backbone. This analysis is exploratory: the primary experimental setting remains gpt-4.1-mini on LongMemEval-Oracle, while the gpt-5.4-mini comparison uses a 20-example LongMemEval-Oracle subset and is reported as a supplementary robustness check.

The auxiliary results are broadly consistent with the main interpretation. CoMMA remains strongest on the 20-example LongMemEval-Oracle subset in Table 9, and it is the only method with non-zero conflict F1 on both gpt-5.4-mini diagnostic slices in Table 10. At the same time, the contrast with the gpt-4.1-mini action slice in Table 8 reinforces our use of the action slice as a control-behavior diagnostic rather than as a standalone conflict-detection benchmark.

F Additional Qualitative Cases

The cases below are intended as illustrative traces of CoMMA's control behavior rather than as a separate evaluation set. They complement the main revision-sensitive trace in Table 4 by showing comparison-sensitive, slot-sensitive, and external-consolidation behavior.

Temporal comparison. For the question "Which book did I finish reading first, 'The Hate U Give' or 'The Nightingale'?" the retrieved evidence contains two book-completion events with different temporal cues. At the generic cluster level, CoMMA leaves the state unresolved because the statements describe relative time in different ways and the turn timestamps reflect mention time rather than event time. Its comparison-specific read-time path nevertheless returns The Hate U Give by comparing the event descriptions directly rather than defaulting to recency. Representative baselines that lack this comparison path return the wrong competing book.

Pairwise device ordering. For the question "Which device did I set up first, the smart thermostat or the mesh network system?" the extracted evidence contains two setup events, one for the smart thermostat and one for the mesh network system. A generic slot-level resolution is not sufficient because the question asks for event ordering rather than for the latest value of a slot. CoMMA's comparison path therefore re-reads the retrieved memories and returns smart thermostat, while overwrite-

Layer	Representative information	Role in evaluation
Example	Question, conversation, reference answer	Defines the instance, the memory context, and the lexical answer target when the expected action is ANSWER.
Control supervision	Question intent, expected action	Specifies the intended temporal reading and the preferred high-level response.
Conflict supervision	Conflict slot, status, resolved value, conflict type	Defines the gold conflict object for conflict F1 and stale-answer diagnostics.
Predicted claim state	Extracted claims, conflict clusters	Intermediate controller state used for conflict metrics and trace checks.
Predicted answer state	Answer mode, cited claims, unresolved clusters	Encodes the final control decision and its trace links.
Budget state	API calls, token counts, estimated cost	Supports efficiency and cost summaries.

Table 6: Operational label layers used by the reported CoMMA evaluation setting.

Method	Memory exposed at answer time	Explicit conflict state	Distinguishing feature
Full context	All turns in original order	No	Uses the full conversation as memory context and answers without conflict clustering or conflict-aware resolution.
RAG only	Top retrieved memory only	No	Uses the shared retriever but truncates to a single retrieved memory before answer generation.
Recency-only	Most recent top-k turns	No	Bypasses learned retrieval and relies on a latest-turn heuristic.
No-conflict controller	Retrieved top-k memories	No	Uses the same retrieval and claim-extraction stack as CoMMA, but passes an empty conflict set to the answerer.
MemR^3-style	Small retrieval, optionally expanded retrieval	No	Approximates a missing-evidence-only baseline and never constructs conflict clusters.
Write-time overwrite	Full conversation collapsed into a latest-value state	No	Extracts claims from all turns and retains only the latest claim for each subject–slot–entity key.
CoMMA	Retrieved top-k memories or structured fact list	Yes	Runs claim extraction, conflict grouping, heuristic-first resolution, and answer-mode selection with an explicit conflict trace.

Table 7: Detailed operational definitions of CoMMA and the six baselines used in the main experiments.

style and no-conflict baselines select the mesh network system.

Routine-update conflict with slot-aware answering. For the question "What time do I usually go to the gym?" the competing claims are 7:00 pm and 6:00 pm. CoMMA resolves the later updated self-report and returns 6:00 pm. The most informative baseline failure is not merely the wrong value but the wrong slot: a missing-evidence-only controller can return a schedule answer such as "Mondays, Wednesdays, and Fridays" instead of returning a time value.

External validation on FactConsolidation. For the question "What is the country of origin of the sport played by Christian Abbiati?" CoMMA returns Philippines, while weaker baselines stop at an intermediate node or follow the wrong relation target. This case does not primarily stress clarification or abstention; instead, it shows that the same read-time conflict-handling pattern can transfer to a structured fact-consolidation setting.

G Prompt Templates

This section reports the main LLM-facing prompt templates used by CoMMA. Table 11 defines the claim-extraction prompt that converts retrieved conversational snippets into provenance-aware structured claims. Table 12 gives the conflict-resolution prompt used when a query-relevant cluster cannot be resolved by deterministic heuristics alone. Table 13 shows the answer-generation prompts for resolved claim states and direct reading from retrieved snippets. Table 14 reports the comparison-read prompt used for ordering, comparison, and time-delta questions.

Method	EM	Tok. F1	Act. Acc.	Conf. F1	Over-Ab.	Cost ($)
Full context	**33.3**	51.6	67.6	0.0	11.8	0.0038
MemR^3-style	29.2	55.8	70.6	0.0	8.8	0.0019
Write-time overwrite	29.2	43.8	61.8	0.0	14.7	0.0036
CoMMA	20.8	**59.0**	**73.5**	0.0	**2.9**	0.0011
No-conflict controller	20.8	46.5	64.7	0.0	17.6	0.0013
RAG only	20.8	29.7	44.1	0.0	38.2	0.0005
Recency only	4.2	10.5	41.2	0.0	50.0	0.0011

Table 8: Appendix action diagnostic slice under the main setting. Values are percentages except cost.

	Answer (%)		Control (%)		Efficiency		
Method	EM	Tok. F1	Act. Acc.	Over-Ab.	Calls	Tokens	Cost ($)
CoMMA	**35.0**	**50.1**	**95.0**	**5.0**	2.9	6,219	0.0059
MemR^3-style	30.0	45.9	90.0	10.0	2.9	5,476	0.0053
No-conflict controller	20.0	40.8	75.0	25.0	1.9	3,069	0.0033
RAG only	20.0	25.0	35.0	65.0	1.8	1,224	0.0015
Full context	10.0	33.4	75.0	25.0	0.5	11,387	0.0097
Write-time overwrite	10.0	23.8	60.0	40.0	2.0	9,638	0.0083
Recency only	10.0	13.7	35.0	65.0	1.9	3,200	0.0031

Table 9: Supplementary gpt-5.4-mini main table on a 20-example LongMemEval-Oracle subset. Values are percentages except calls, tokens, and cost.

(a) Supplementary action slice under gpt-5.4-mini

Method	EM	Tok. F1	Act. Acc.	Conf. F1	Over-Ab.	Cost ($)
Full context	**33.3**	**61.2**	70.6	0.0	5.9	0.0074
CoMMA	29.2	55.9	70.6	**12.9**	2.9	0.0026
No-conflict controller	25.0	50.8	70.6	0.0	5.9	0.0028
Write-time overwrite	25.0	41.4	64.7	0.0	11.8	0.0069
MemR^3-style	20.8	42.5	**73.5**	0.0	0.0	0.0026
RAG only	20.8	31.9	50.0	0.0	29.4	0.0013
Recency only	4.2	9.1	47.1	0.0	41.2	0.0024

(b) Supplementary conflict slice under gpt-5.4-mini

Method	EM	Tok. F1	Act. Acc.	Conf. F1	Over-Ab.	Cost ($)
Full context	**33.3**	**57.3**	**92.0**	0.0	4.0	0.0078
Write-time overwrite	**33.3**	53.4	86.0	0.0	10.0	0.0071
CoMMA	31.3	51.8	**92.0**	**14.0**	4.0	0.0026
No-conflict controller	29.2	50.9	86.0	0.0	10.0	0.0030
MemR^3-style	27.1	45.1	90.0	0.0	4.0	0.0025
RAG only	16.7	28.7	60.0	0.0	36.0	0.0013
Recency only	8.3	21.0	46.0	0.0	48.0	0.0026

Table 10: Auxiliary diagnostic-slice results under gpt-5.4-mini.

Claim Extraction Prompts

System prompt.

```
You extract structured memory claims from conversational snippets.

Return strict JSON with this schema:
{
  "claims": [
    {
      "turn_id": "3",
      "subject": "user",
      "slot": "location",
      "value": "Seattle",
      "support_text": "I moved to Seattle",
      "entity_key": null,
      "polarity": "positive",
      "temporal_scope": "current",
      "revision_cue": true,
      "confidence": 0.0
    }
  ]
}

Rules:
- Extract only explicit claims grounded in the provided snippets.
- Every claim MUST include the exact turn_id of the snippet that supports it.
- support_text must be a short exact quote copied from the supporting snippet.
- Never combine information across different snippets into one claim.
- Use short slot names such as location, preference, occupation, event_date.
- Use revision_cue=true when the snippet explicitly updates, corrects, or replaces
    an older memory.
- temporal_scope should be one of current, past, future, unknown.
- If nothing usable is present, return {"claims": []}.
```

User prompt template.

```
Question: {question}

Retrieved snippets:
{snippets}

Return JSON only.
```

Table 11: Prompt templates for claim extraction in the current CoMMA implementation.

Conflict Resolution Prompts

System prompt.

```
You resolve conflicts among structured conversational memory claims.

Return strict JSON with this schema:
{
  "status": "resolved or unresolved",
  "resolved_value": "value or null",
  "confidence": 0.0,
  "basis": ["short reason"],
  "accepted_claim_id": "claim id or null",
  "reason": "brief explanation"
}

Use only the evidence provided. Prefer explicit corrections and later timestamps.
If the evidence is genuinely ambiguous, choose unresolved.
```

User prompt template.

```
Question: {question}

Conflict cluster:
{cluster_json}

Return JSON only.
```

Table 12: Prompt templates for conflict resolution in the current CoMMA implementation.

Answer Generation Prompts (Resolved-State and Direct-Read)

Resolved-state answer prompt (system).

```
You answer conversational memory questions using resolved claims.

Return strict JSON with this schema:
{
  "answer": "short answer",
  "answer_type": "direct or qualified or clarify or abstain",
  "confidence": 0.0,
  "rationale": "brief explanation"
}

Rules:
- Use `clarify` when the question is underspecified and a short disambiguating
    question is safer than guessing.
- Use `abstain` when the evidence remains unresolved and clarification would not fix
     the uncertainty.
- Keep answers short and grounded in the provided state.
```

Resolved-state answer prompt (user).

```
Question: {question}

Resolved state:
{state_json}

Return JSON only.
```

Direct-read prompt (system).

```
You answer conversational memory questions directly from retrieved snippets.

Return strict JSON with this schema:
{
  "answer": "short answer",
  "answer_type": "direct or qualified or clarify or abstain",
  "confidence": 0.0,
  "rationale": "brief explanation"
}

Rules:
- Use only the provided snippets.
- Prefer concise grounded answers over paraphrased speculation.
- Use `clarify` when a short disambiguating question is safer than guessing.
- Use `abstain` when the snippets do not contain enough evidence.
- For first/last/before/after questions, compare the snippets and answer directly if
     the ordering is explicit.
```

Direct-read prompt (user).

```
Question: {question}

Retrieved snippets:
{snippets}

Fallback answer:
{fallback_json}

Return JSON only.
```

Table 13: Resolved-state and direct-read answer prompts in the current CoMMA implementation.

Answer Generation Prompts (Comparison-Read)

Comparison-read prompt (system).

```
You answer ordering, comparison, and time-delta memory questions from the provided
    snippets only.

Return strict JSON with this schema:
{
  "answer": "short answer",
  "answer_type": "direct or qualified or clarify or abstain",
  "confidence": 0.0,
  "rationale": "brief explanation"
}

Rules:
- Use only the provided snippets.
- Do not guess.
- Keep the answer short and grounded in the snippets.
```

Comparison-read prompt (user).

```
Question: {question}

Question intent: {question_intent}
Question focus: {question_focus_json}
Timeline reference: {timeline_reference}

Retrieved snippets:
{snippets}

Fallback answer:
{fallback_json}

Return JSON only.
```

Table 14: Comparison-read prompts in the current CoMMA implementation.