Not All Memories Are Equal: Hierarchical Collaborative Memory for Validity-Aware Retrieval in LLM Agents

Anonymous ACL submission

Abstract

In team collaboration scenarios, memory is heterogeneous and continually evolving. Team memories capture collective decisions, protocols, and current consensus, while individual memories preserve member-specific observations, execution traces, and intermediate progress. Existing memory-augmented systems typically retrieve from all stored memories as a flat pool, ranking them by semantic relevance, importance, or recency without modeling hierarchical structure or evolving validity. As a result, they often surface semantically relevant but outdated or conflicting memories, especially individual memories that no longer align with current team consensus, instead of prioritizing currently valid memories. This is particularly problematic when collaborative LLM agents answer user questions, since their responses should be grounded in valid memories. We propose HiCoMER, a framework for hierarchical collaborative memory management and validity-aware retrieval in LLM agents. HiCoMER first maintains the validity of team and individual memories and then retrieves memories that remain valid, rather than retrieving directly from all stored memories. It consists of three components: a Hierarchical Memory Conflict Updater, a Validity-Aware Memory Retriever, and a Memory-Grounded Answer Generator. To evaluate HiCoMER, we construct two new datasets for memory-grounded question answering in collaborative settings. Experiments on both datasets show that HiCoMER consistently outperforms strong baselines by reducing outdated retrieval, preserving current team consensus, and improving downstream QA quality[1].

1 Introduction

Large language model (LLM) agents are increasingly moving beyond single-turn assistance toward long-horizon collaboration in shared environments (Boiko et al., 2023; Gao et al., 2024). In these settings, memory is heterogeneous and hierarchical: team memories record collective decisions and protocols, while individual memories capture member-specific observations and execution traces (Shuster et al., 2022). Agents may retrieve semantically relevant but invalid memories, overlooking current team consensus. For example, when asked "How should we synthesize Compound A?", an agent may select a past individual experiment log while ignoring a team-level update banning Compound A due to toxicity. Existing memory-augmented systems (Lewis et al., 2020; Packer et al., 2023) retrieve from a flat pool based on semantic similarity or recency, failing to capture hierarchical conflicts or evolving validity, as shown in Figure 1.

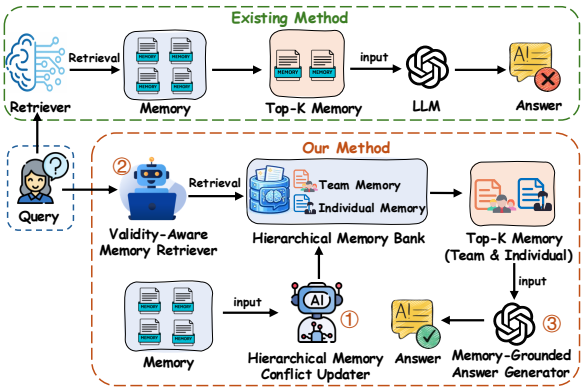

Figure 1: Illustration comparing existing methods and our proposed HiCoMER.

Moreover, collaborative memories are often produced in temporally aligned groups. A single meeting or experiment cycle may generate both team-level decisions and multiple individual records, some of which may conflict with the current consensus. Memory management must therefore account for both within-group hierarchical conflicts and cross-time validity changes.

[1]All data and code will be made publicly available. An anonymized copy is included with this submission for review.

To address these challenges, we propose HiCoMER, a hierarchical collaborative memory framework. HiCoMER maintains memory validity and performs validity-aware retrieval through three components: a Hierarchical Memory Conflict Updater, a Validity-Aware Memory Retriever, and a Memory-Grounded Answer Generator. This ensures agents prioritize currently valid memories. We construct two new datasets for memory-grounded question answering in collaborative settings. Experiments show that HiCoMER consistently reduces outdated retrieval, preserves team consensus, and improves downstream QA quality.

Our contributions are as follows:

- We formalize the problem of hierarchical memory conflict in collaborative LLM settings, where both team and individual memories may evolve over time, and semantically relevant memories can become invalid under the current consensus.
- We propose HiCoMER, a hierarchical collaborative memory framework that maintains memory validity and performs conflict-aware, validity-aware retrieval. HiCoMER consists of three components: a Hierarchical Memory Conflict Updater, a Validity-Aware Memory Retriever, and a Memory-Grounded Answer Generator, which together enable agents to prioritize currently valid memories.
- We construct two new datasets for memory-grounded question answering in collaborative environments. Experiments demonstrate that HiCoMER consistently improves retrieval safety, consensus preservation, and downstream QA performance over strong baselines.

2 Related Work

2.1 Long-Term Memory Retrieval

Retrieval-augmented generation (RAG) grounds LLM outputs on retrieved evidence and is widely used in knowledge-intensive NLP (Lewis et al., 2020). Many works improve the retrieval backbone, including dense retrievers for open-domain QA such as DPR (Karpukhin et al., 2020), latent-retrieval pretraining like REALM (Guu et al., 2020), and large-scale retrieval-augmented pretraining as in RETRO (Borgeaud et al., 2022). On the reader side, fusion-based architectures such as FiD (Izacard and Grave, 2021) and end-to-end retrieval-augmented models like Atlas (Izacard and Grave, 2022) enhance robustness and integration. Complementary IR advances include late interaction models like ColBERT (Khattab and Zaharia, 2020), sparse expansion models like SPLADE (Formal et al., 2021), and query-side augmentation such as HyDE (Gao et al., 2022). Self-reflective pipelines like Self-RAG critique retrieved evidence and generations to improve faithfulness (Asai et al., 2023). Agent-oriented systems treat memory as a persistent, growing corpus with write/read policies, e.g., MemGPT (Packer et al., 2023) and interactive generative agents (Park et al., 2023). Despite these advances, most approaches optimize flat relevance signals and do not explicitly model hierarchical validity or authority, which is central to our setting.

2.2 Hierarchical Consistency under Conflicts

Conflict resolution is often studied through contradiction detection, entailment reasoning, and factual verification. NLI datasets like SNLI and MultiNLI provide supervision for entailment and contradiction (Bowman et al., 2015; Williams et al., 2018), and adversarial benchmarks such as ANLI stress-test robustness (Nie et al., 2020). Fact verification datasets like FEVER formalize consistency as verifying claims against evidence (Thorne et al., 2018). For generation, approaches include FactCC (Kryściński et al., 2020), QAGS-style QA checks (Wang et al., 2020), and broader evaluations like TruthfulQA (Lin et al., 2022) and TRUE (Honovich et al., 2022). Post-hoc detection and correction methods include SelfCheckGPT (Manakul et al., 2023), iterative retrieval-revision pipelines like RARR (Ram et al., 2023), and fine-grained factual scoring such as FActScore (Min et al., 2023). However, these works mostly resolve conflicts at generation or evaluation time, rather than as write-time operations on evolving memories. In collaborative environments, conflicts are frequently asymmetric: team decisions and SOP updates can invalidate many individual logs even if semantically similar to future queries. HiCoMER addresses this by organizing collaborative memories as time-aligned groups, performing trainable maintenance over hierarchical and temporal conflicts, and learning a validity-aware retrieval function over the maintained memory bank.

3 Methodology

3.1 Framework Overview

HiCoMER is a collaborative memory framework for long-horizon LLM agents operating in team

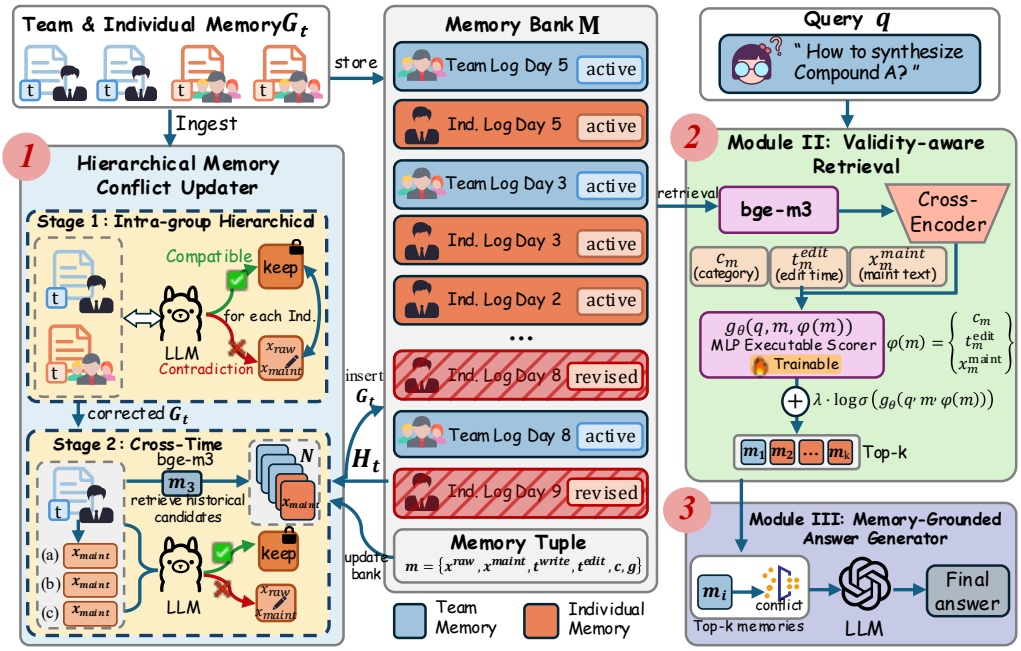

Figure 2: Overall framework of HiCoMER. The framework consists of three modules: Hierarchical Memory Conflict Updater, Validity-Aware Memory Retriever, and Memory-Grounded Answer Generator.

environments. In collaborative settings, memory is inherently heterogeneous: Team memories capture shared decisions, protocols, and current coordination state, while Individual memories record local observations, execution traces, and intermediate progress. Memory retrieval should therefore not rely solely on semantic relevance, because semantically relevant memories may already be outdated or conflict with other memories under the current collaborative state. This issue becomes particularly severe when agents retrieve directly from raw stored memories without explicitly modeling the distinct functions of Team and Individual memories or maintaining the memory bank to resolve outdated and conflicting memories. To address this problem, HiCoMER explicitly models collaborative memory at both the Team and Individual levels and maintains the memory bank to preserve a valid collaborative memory state for downstream question answering.

As illustrated in Figure 2, HiCoMER consists of three modules. The first module, Hierarchical Memory Conflict Updater, maintains the memory bank under two types of conflict: same-time hierarchical conflict between Team and Individual memories in the current group, and cross-time conflict between the current maintained Team/Individual memories and previously stored memories. The second module, Validity-Aware Memory Retriever, ranks candidate memories by combining semantic relevance with maintained memory-state signals, so that retrieval is guided not only by textual relevance but also by whether a memory remains valid under the current collaborative state. The third module, Memory-Grounded Answer Generator, produces the final response based on the retrieved memories after a lightweight conflict-resolution step.

Unlike conventional memory-augmented systems that retrieve directly from raw stored memories, HiCoMER explicitly models Team and Individual memories and maintains their evolving validity over time. This design allows the framework to reduce the retrieval of conflicting or outdated memories and ultimately generate answers grounded in currently valid memories rather than raw historical accumulation.

3.2 Hierarchical Memory Conflict Updater

This module maintains the collaborative memory bank by resolving outdated and conflicting memories under the hierarchical structure of Team and Individual memories. Its goal is to preserve a valid collaborative memory state for downstream retrieval and answer generation.

At each time step t, HiCoMER receives a time-aligned memory group

$$G_t = M_t^{\text{Team}} \cup M_t^{\text{Individual}}, \tag{1}$$

where M_t^{Team} contains collective records such as

meeting resolutions, protocol updates, and supervisor instructions, while $M_t^{\text{Individual}}$ contains individual records such as execution logs, observations, failed attempts, and intermediate progress. During maintenance, each memory is associated with both its original content and its maintained content, so that HiCoMER preserves the historical record while updating the currently valid memory view used by downstream modules.

HiCoMER first identifies a bounded set of relevant historical memories, because the current group is unlikely to affect the entire historical memory bank and exhaustively comparing against all past memories would be computationally inefficient. Specifically, it constructs a textual query q_t^{hist} from the current Team and Individual memories and retrieves the Top-K most relevant memories from the previously stored bank:

$$H_t = \text{TopK}_{m \in \mathcal{M}_{<t}} \; s_\phi\left(q_t^{\text{hist}}, m\right), \qquad (2)$$

where $\mathcal{M}_{<t}$ denotes all Team and Individual memories stored before time t, q_t^{hist} is constructed from the current group G_t, and $s_\phi(\cdot,\cdot)$ is a dense semantic retrieval model that computes the relevance between the current-group query and each historical memory. H_t contains the Top-K historical Team/Individual memories with the highest retrieval scores.

Given the current group G_t and the retrieved historical candidates H_t, the updater performs maintenance at two levels. The first level updates the current group itself. Within G_t, Team and Individual memories may express inconsistent states. HiCoMER therefore performs an intra-group hierarchical update that revises the current Individual memories under the coordination signal provided by the current Team memories, yielding a maintained current-group state:

$$\widetilde{G}_t = M_t^{\text{Team}} \cup \widetilde{M}_t^{\text{Individual}}, \qquad (3)$$

where $\widetilde{G}_t$ denotes the maintained version of the current group after conflict resolution. In this stage, the Team memories are treated as the authoritative coordination state at time t, and the updater revises Individual memories when they are incompatible with that state.

The second level updates the retrieved historical candidates. HiCoMER uses the maintained current-group state $\widetilde{G}_t$, which consists of the current Team memories and the updated current Individual memories, to perform an inter-group hierarchical update over the historical candidates in H_t. This step revises previously stored Team and Individual memories whose maintained content has become outdated or conflicting under the current state.

We instantiate the updater with a lightweight instruction-tuned LLM and train it to generate structured maintenance decisions rather than free-form outputs. For each target memory, the model predicts three elements: a relation label, an action label, and a revised maintained text when revision is required. The relation label is drawn from $\{\text{compatible}, \text{neutral}, \text{contradiction}\}$, and the action label is drawn from $\{\text{keep}, \text{revise}\}$. When the predicted action is "revise", HiCoMER updates the maintained content.

We train the updater in two stages. We first perform supervised fine-tuning on structured maintenance supervision, so that the model learns to produce the desired decision format and content. We then further optimize it using Group Relative Policy Optimization (GRPO). Given an input context u, the policy samples a set of candidate structured outputs

$$\mathcal{Y}(u) = \{\tilde{y}_1, \ldots, \tilde{y}_G\}, \qquad (4)$$

which are scored by a deterministic reward

$$R(\tilde{y}; u) = \lambda_1 R_{\text{cons}} + \lambda_2 R_{\text{rev}} + \lambda_3 R_{\text{fmt}}, \qquad (5)$$

where R_{cons} measures consistency with the gold maintained state, R_{rev} encourages necessary but minimal revision, and R_{fmt} enforces schema-valid output. In this way, GRPO encourages the updater to generate maintenance decisions that are accurate, concise, and structurally valid.

3.3 Validity-Aware Memory Retriever

This module retrieves memories that are not only semantically relevant to the query but also valid under the current maintained collaborative state. Its goal is to reduce the retrieval of outdated or conflicting memories while preserving evidence that remains useful for downstream answer generation.

Given a user query q and the maintained memory bank $\mathcal{M}_{\le t}$ after processing groups up to time t, HiCoMER first retrieves a high-recall candidate set

$$R_t(q) = \text{TopN}_{m \in \mathcal{M}_{\le t}} \; r_\psi(q, m), \qquad (6)$$

where $r_\psi(\cdot,\cdot)$ is a dense retrieval model that measures the semantic relevance between the query and each maintained memory, and $R_t(q)$ contains the top-N candidate memories returned by the first-stage retriever.

HiCoMER then reranks each candidate memory $m \in R_t(q)$ using two complementary signals.

The first is a semantic relevance score

$$S_{\text{sem}}(q, m) = f_\omega(q, m), \tag{7}$$

where $f_\omega(\cdot, \cdot)$ is a cross-encoder reranker that jointly encodes the query and the candidate memory for fine-grained relevance scoring.

The second is a validity-aware score that reflects whether a candidate memory remains usable under the current maintained state. For each candidate memory, HiCoMER considers its maintained content, its source type (Team or Individual), and its most recent maintenance time. Let h_q and h_m^{maint} denote the dense representations of the query and the maintained content of memory m, respectively. HiCoMER then constructs a feature vector

$$v(q, m) = [h_m^{\text{maint}}; h_q \odot h_m^{\text{maint}}; e(c_m); \tau_m], \tag{8}$$

where $e(c_m)$ is a trainable embedding of the memory type $c_m \in \{\text{Team}, \text{Individual}\}$, and τ_m is a normalized timestamp feature derived from the latest maintenance time of m. A lightweight MLP produces the validity-aware score

$$S_{\text{val}}(q, m) = g_\theta(v(q, m)). \tag{9}$$

HiCoMER combines these two signals into a unified retrieval score:

$$S_{\text{final}}(q, m) = S_{\text{sem}}(q, m) + \lambda \cdot \log \sigma(S_{\text{val}}(q, m)), \tag{10}$$

where λ controls the contribution of the validity-aware component. The final retrieved evidence is obtained by ranking candidate memories in $R_t(q)$ according to $S_{\text{final}}(q, m)$.

We train the retriever after obtaining maintained memory states from Section 3.2. For each query q, we construct pairwise ranking instances consisting of a positive memory m^+ and a negative memory m^-, where m^+ is a gold supporting memory that remains valid under the maintained state, while m^- is an outdated, conflicting, or semantically similar but non-valid competitor. The retriever is optimized with a pairwise ranking objective

$$\mathcal{L}_{\text{retr}} = -\log \left(\frac{\exp(s^+)}{\exp(s^+) + \exp(s^-)} \right). \tag{11}$$

so that the validity-aware scorer learns to rank currently valid memories above outdated or conflicting alternatives.

At inference time, HiCoMER first retrieves a candidate set using the dense retriever and then reranks the candidates using the unified score $S_{\text{final}}(q, m)$. The resulting ranked memories are passed to the downstream answer generator.

3.4 Memory-Grounded Answer Generator

This module generates the final answer from the retrieved memories. Its goal is to produce responses grounded in the retrieved evidence while suppressing residual conflicts that may still remain among the top-ranked candidates.

Given a user query q and the ranked memory set returned by the retriever, a lightweight conflict-resolution step is applied over the retrieved evidence. Although Module I and Module II already reduce outdated and conflicting memories, residual inconsistency may still remain when multiple memories are semantically relevant to the same query but reflect different collaborative states.

To address this issue, each retrieved memory is assigned a priority score based on its source type, maintained validity, and maintenance time:

$$\begin{aligned} P(m) = \alpha \cdot \mathbb{I}[c_m = \text{Team}] \\ + \beta \cdot \mathbb{I}[\text{Valid}(m)] \\ + \gamma \cdot \text{NormTime}(t_m^{edit}), \end{aligned} \tag{12}$$

where α, β, and γ are positive weights, c_m denotes the memory type, and $\text{NormTime}(t_m^{edit})$ is the normalized maintenance-time score. If two retrieved memories have contradictory maintained contents, the memory with higher priority is retained. Detailed conflict-resolution rules are provided in Appendix A. The resulting evidence set is then passed to the answer generator.

The generator receives the query together with the resolved evidence set and produces the final response based solely on this evidence.

4 Experiments

4.1 Dataset

Existing datasets lack explicit hierarchical conflicts and temporally-aligned team versus individual memories, which are crucial for evaluating memory maintenance and validity-aware retrieval in collaborative long-horizon tasks. To address this gap, we construct two new datasets corresponding to two biomedical collaboration settings: Acute Lung Injury (ALI) and PROTAC platform (PROTAC).

Table 1: Detailed statistics.

Statistic	ALI	PROTAC	Total
Candidate papers	30	30	60
Retained papers	27	26	53
Valid groups	908	873	1,781
Memory documents	8,917	8,801	17,718
Injected conflicts	317	304	621
Same-time conflicts	189	181	370
Cross-time conflicts	128	123	251
Train papers	19	18	37
Validation papers	3	3	6
Test papers	5	5	10

The ALI dataset focuses on sepsis-related therapeutic development, where conflicts are primarily driven by safety-sensitive protocol updates, treatment discontinuation, and endpoint revisions. The PROTAC dataset models linker-free PROTAC platform development under N-end rule constraints, with conflicts arising more often from target-strategy revisions, assay replacements, and platform-level design invalidations.

The detailed dataset statistics are summarized in Table 1. The full construction process is provided in Appendix B, and illustrative examples are provided in Appendix C.

4.2 Baselines and Evaluation Metrics

We compare HiCoMER against a diverse set of baselines covering standard semantic retrieval pipelines such as flat RAG, hybrid RAG, and rerank RAG; recency-based memory control methods; agentic and hierarchical memory frameworks including MemGPT-style agents (Packer et al., 2023) and G-Memory (Zhang et al., 2025); and reflection- or production-oriented systems that rely on generation-time reasoning without explicit memory-bank maintenance, such as Self-RAG (Asai et al., 2023) and Mem0 (Chhikara et al., 2025). These baselines are chosen to test whether hierarchical conflict resolution can be addressed by stronger semantic matching alone, by recency heuristics, by existing long-horizon memory mechanisms, or by generation-time reflection without explicit memory maintenance. We also include ablated variants of HiCoMER to isolate the contributions of the Hierarchical Memory Conflict Updater and the Validity-Aware Memory Retriever. For fairness, all methods are evaluated on the same grouped memory stream and query interface, and system-level baselines are adapted to the same retrieval settings. Full baseline definitions are provided in Appendix D.

Performance is evaluated at three levels. For the Hierarchical Memory Conflict Updater, we measure structured maintenance quality using Conflict F1 and Maintenance Accuracy. For upstream maintained-state retrieval, we use Outdated Retrieval Rate at 5 (ORR@5) to quantify the reduction of outdated memory retrieval, Consensus Retention Rate at 5 (CRR@5) to evaluate preservation of authoritative Team consensus, and NDCG at 10 (NDCG@10) to measure overall executable ranking quality. Finally, for the full HiCoMER pipeline, we report ROUGE-L and Decision F1 to evaluate whether improvements in maintained-state retrieval translate into more accurate and executable memory-grounded answers. Detailed metric definitions are provided in Appendix D.

4.3 Implementation Details

We evaluate HiCoMER on the ALI and PROTAC datasets under two backbone settings. In the strong-backbone setting, both the Hierarchical Memory Conflict Updater and Memory-Grounded Answer Generator use Llama-3.1-8B. In the lightweight setting, both use Qwen2.5-3B-Instruct. The Validity-Aware Memory Retriever is shared across settings. All experiments use the same grouped memory streams, query interface, and downstream prompt templates to ensure fair comparison.

Training proceeds sequentially. First, the updater is trained on structured maintenance instances to learn hierarchical memory conflict resolution. The trained updater generates maintained memory states for the training split, which are then used to train the Validity-Aware Memory Retriever with a pairwise ranking objective. At inference, maintained memories are retrieved and reranked for executability, and the top memories are passed to the answer generator.

4.4 Results

We first present the main results on the ALI dataset, our primary in-domain evaluation setting. To assess effectiveness under a stronger backbone, we evaluate HiCoMER with Llama-3.1-8B for both the Hierarchical Memory Conflict Updater and the Memory-Grounded Answer Generator. We also evaluate a lighter deployment-oriented setting using Qwen2.5-3B-Instruct to test the robustness of the framework under constrained compute and memory budgets. Tables 2 and 3 report the end-to-

Table 2: End-to-end evaluation of the full HiCoMER pipeline on the ALI dataset using Llama-3.1-8B as the backbone for the Hierarchical Memory Conflict Updater and the Memory-Grounded Answer Generator. Retrieval metrics ORR@5, CRR@5, and NDCG@10 measure maintained-state retrieval quality, while answer-level metrics ROUGE-L and Decision F1 assess how well the pipeline converts maintained and retrieved memories into final memory-grounded responses.

Method	**ORR@5 (↓)**	**CRR@5 (↑)**	**Decision F1 (↑)**	**ROUGE-L (↑)**	**NDCG@10 (↑)**
Standard & Semantic Retrieval					
Flat RAG	45.83	51.94	41.67	35.38	42.76
Hybrid RAG	41.92	55.68	44.27	37.14	45.31
Rerank RAG	37.57	61.83	48.88	40.47	51.86
Multi-Agent & Hierarchical Memory					
G-Memory	29.74	68.91	53.79	44.82	57.63
Agentic Memory (Park)	32.81	64.66	52.14	43.58	54.77
Reflection & Production Systems					
Self-RAG	35.12	62.57	55.23	45.97	53.64
MemGPT-style Window	23.79	49.08	46.97	41.73	48.88
Mem0	28.63	70.84	57.36	47.48	59.27
HiCoMER (Llama-3.1-8B)	**14.18**	**84.87**	**64.61**	**52.93**	**68.74**

end performance of HiCoMER. Retrieval-oriented metrics ORR@5, CRR@5, and NDCG@10 primarily measure maintained-state retrieval quality, capturing the joint effect of the Hierarchical Memory Conflict Updater and the Validity-Aware Memory Retriever. Answer-level metrics ROUGE-L and Decision F1 further evaluate how improvements in retrieval translate into higher-quality memory-grounded responses. Using Qwen2.5-3B-Instruct, HiCoMER maintains strong performance, demonstrating that the maintenance-and-retrieval design is robust to smaller backbone models. Gains are observed in both maintained-state retrieval and final response quality.

Overall, HiCoMER substantially reduces outdated or non-validity-aware retrievals while better preserving authoritative Team consensus. This leads to improved executable ranking and stronger downstream responses. The consistent gains across both Llama-3.1-8B and Qwen2.5-3B-Instruct suggest that improvements stem from the framework design rather than backbone scale. These results highlight that semantic relevance alone is insufficient in hierarchical collaborative memory settings: neither stronger semantic matching, recency heuristics, nor generation-time reflection can fully replace explicit memory maintenance and validity-aware retrieval. Further component-level evaluation of the Hierarchical Memory Conflict Updater on held-out structured maintenance instances is provided in Appendix E.

Additional results on the PROTAC dataset and extended per-baseline analyses are provided in Appendix F.

4.5 Ablation Study

To quantify the contribution of each major module in HiCoMER, we conduct a full-pipeline ablation study on the ALI dataset. We remove each of the three core modules individually: the Hierarchical Memory Conflict Updater, the Validity-Aware Memory Retriever, and the Memory-Grounded Answer Generator. Table 4 reports both retrieval-oriented metrics and answer-level metrics to assess how degradations in upstream modules propagate to final response quality.

Removing either the updater or the validity-aware retriever leads to substantial drops in both maintained-state retrieval metrics and final response quality, confirming that these upstream modules are critical for end-to-end performance. Removing the answer generator also degrades final response metrics, highlighting its role in converting retrieved memories into high-quality answers. Additional ablation analyses on the PROTAC dataset are provided in Appendix G.

4.6 Human Evaluation

We conduct a human evaluation to assess the practical utility of system outputs in realistic scenario-based tasks. Seventeen participants with research experience, including Ph.D. students, master's students, and junior research assistants, evaluated 10 report-writing tasks each, yielding a total of 170

Table 3: End-to-end evaluation of HiCoMER on the ALI dataset using Qwen2.5-3B-Instruct for both the Hierarchical Memory Conflict Updater and the Memory-Grounded Answer Generator. Retrieval metrics ORR@5, CRR@5, and NDCG@10 assess maintained-state ranking, while answer-level metrics ROUGE-L and Decision F1 measure the quality of memory-grounded responses under a smaller backbone.

Method	**ORR@5 (↓)**	**CRR@5 (↑)**	**Decision F1 (↑)**	**ROUGE-L (↑)**	**NDCG@10 (↑)**
Standard & Semantic Retrieval					
Flat RAG	46.57	50.86	39.43	33.57	41.93
Hybrid RAG	43.04	54.88	42.16	35.61	44.36
Rerank RAG	38.76	60.43	46.48	38.92	50.37
Multi-Agent & Hierarchical Memory					
G-Memory	31.03	67.12	50.84	42.69	55.87
Agentic Memory (Park)	34.08	63.04	49.58	41.76	53.41
Reflection & Production Systems					
Self-RAG	36.77	60.93	52.09	43.46	51.94
MemGPT-style Window	24.46	47.18	44.19	39.54	46.72
Mem0	30.18	68.37	53.88	44.87	57.12
HiCoMER (Qwen2.5-3B-Instruct)	**16.07**	**82.34**	**60.28**	**49.81**	**66.08**

Table 4: Ablation study on the ALI dataset. ORR@5, CRR@5, and NDCG@10 measure maintained-state retrieval quality, while ROUGE-L and Decision F1 assess memory-grounded response quality. Each row shows the effect of removing a single module from the full HiCoMER pipeline.

Variant	ORR@5	CRR@5	NDCG@10	ROUGE-L	Decision F1
Full HiCoMER (Llama-3.1-8B)	14.18	84.87	68.74	52.93	64.61
w/o Hierarchical Memory Conflict Updater	30.42	71.63	58.91	46.86	55.94
w/o Validity-Aware Memory Retriever	29.16	70.31	57.12	45.97	54.83
w/o Memory-Grounded Answer Generator	14.18	84.87	68.74	47.62	56.48

Table 5: Human evaluation results on simulated system utility, based on 170 evaluation instances. Memory Precision measures the relevance and accuracy of retrieved context, and Intervention Usefulness measures the actionability of the system's suggestions.

Method	Memory Precision	Intervention Usefulness
Mem0	3.73	3.68
HiCoMER	4.11	4.04

evaluation instances. In each task, participants were shown outputs generated under the same scenario by HiCoMER and Mem0 in randomized order without method identifiers.

Participants rated each output on two 5-point scales. Memory Precision measures the relevance and accuracy of retrieved context, while Intervention Usefulness measures the actionability of the system's suggestions. Comparative results are reported in Table 5.

Participants consistently preferred HiCoMER over Mem0 on both memory precision and intervention usefulness, indicating that the improvements observed in automatic metrics translate into more practically useful assistance in realistic scientific collaboration scenarios.

5 Conclusion

We present HiCoMER, a hierarchical collaborative memory framework for long-horizon LLM agents. HiCoMER models team-level and individual-level information, maintains memory consistency with a trainable hierarchical conflict updater, retrieves currently valid memories via a validity-aware scorer, and generates final answers grounded in the maintained evidence.

Experimental results on the datasets constructed in this work show that HiCoMER substantially reduces outdated retrieval while preserving valid memory states, improving downstream tasks and yielding higher-quality, memory-grounded answers and positive human evaluation outcomes.

Future work includes richer conflict structures, multi-level supersession, and robustness under noisier or asynchronous memory streams. We hope HiCoMER encourages a shift from volume-oriented retrieval toward validity-aware memory management in long-term LLM agents.

Limitations

While HiCoMER demonstrates consistent improvements in hierarchical memory maintenance and validity-aware retrieval, a few limitations should be noted. First, noisy, incomplete, or misaligned Team or Individual records could affect maintenance accuracy and downstream answer quality. Finally, our current evaluation focuses on biomedical collaboration scenarios. Future work will extend testing to additional domains.

Ethical Statement

HiCoMER is designed for research and simulation of collaborative memory management in LLM agents and does not directly interact with human subjects or sensitive patient data in deployed settings. All datasets used in this work are derived from publicly available scientific publications and are reverse-simulated to avoid exposing identifiable human data.

For the human evaluation, participants were recruited voluntarily from research-experienced individuals and performed scenario-based tasks on simulated data only. No private or sensitive information was used, and all participant data were anonymized. Participants provided informed consent and were debriefed after the evaluation.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Daniil A. Boiko, Robert MacKnight, Gabe Gomes, and 1 others. 2023. Autonomous chemical research with large language models. *Nature*. ArXiv:2304.05332 (extended version).

Sebastien Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George B. Heigold, Chris J. Maddison, Tianqi Liu, Michael M. Chen, , and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Prateek Chhikara, Dev Khant, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Shashank Gao and 1 others. 2024. Empowering biomedical discovery with AI agents. *Cell*. Perspective article.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Or Honovich, Tomer Wolfson, Avia Efrat, Ori Ram, and Omer Levy. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of NAACL*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Gautier Izacard and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, 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)*.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of EMNLP*.

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 (NeurIPS)*. ArXiv:2005.11401.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of ACL*.

Potsawee Manakul, Krisztian Balog, and Takahiro Kamigaito. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of EMNLP*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Hannaneh Hajishirzi, Luke Zettlemoyer, and Danqi Chen. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of EMNLP*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of ACL*.

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

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*.

Ori Ram, Itai Gat, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2023. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Kurt Shuster, Stephen Roller, Naman Goyal, and 1 others. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: A large-scale dataset for fact extraction and verification. In *Proceedings of NAACL-HLT*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of ACL*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025. G-Memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*.

A Detailed Conflict Resolution in Answer Generation

Given two retrieved memories m_i and m_j, HiCoMER checks if their maintained contents are contradictory:

$$\mathrm{Contradict}(x_{m_i}^{\mathrm{maint}}, x_{m_j}^{\mathrm{maint}}) = 1. \quad (13)$$

If no contradiction is detected, both memories are retained. If a contradiction exists, the memory with the higher priority is retained:

$$m_i \succ m_j \quad \text{if} \quad \mathrm{Contradict}(x_{m_i}^{\mathrm{maint}}, x_{m_j}^{\mathrm{maint}}) = 1 \\ \wedge\ P(m_i) > P(m_j). \quad (14)$$

The final evidence set for answer generation is then defined as

$$\mathcal{C}^* = \{m \in R_t(q) \mid \nexists\, m' \in R_t(q) \text{ such that } \\ m' \succ m\}. \quad (15)$$

This procedure yields the following preference order in common cases:

- Team memories are preferred over Individual memories when their maintained contents conflict.
- Among memories of the same type, those that remain valid under the maintained state are preferred over invalid or outdated ones.
- If both memories have the same type and validity, the memory with the more recent maintenance time is preferred.

After conflict resolution, each memory in $\mathcal{C}^*$ is serialized with its source type, maintenance time, and maintained content, and the resulting structured evidence set is passed to the instruction-following LLM for final answer generation.

B ALI and PROTAC Dataset Construction Details

This appendix provides full details on the construction of the ALI and PROTAC datasets. For each domain, we start with a candidate pool of 30 PubMed papers. After paper-level eligibility screening, 27 ALI papers and 26 PROTAC papers are retained. Each retained paper is reverse-simulated into a 12-week collaborative lifecycle, divided into event

windows spanning 48–72 hours. Each event window corresponds to a grouped memory unit containing Team memories, which encode collective decisions, protocol-level constraints, and authoritative consensus, and Individual memories, which capture local execution traces, observations, and intermediate progress.

Hierarchical conflicts are explicitly injected, including same-time conflicts within a single event window and cross-time conflicts spanning multiple windows. ALI contains 317 injected conflicts (189 same-time, 128 cross-time), and PROTAC contains 304 injected conflicts (181 same-time, 123 cross-time). To reduce document-level leakage, train/validation/test splits are organized by source paper: ALI uses 19/3/5 and PROTAC uses 18/3/5. Both datasets support six query types: fact lookup, procedure validation, conflict diagnosis, risk warning, next-step decision, and protocol consistency checking. GPT-5 is used for reverse simulation, query-answer generation, and evidence-trace annotation. Across valid groups, each contains on average 2.1 Team memories and 7.8 Individual memories, totaling 9.9 documents per group.

C Illustrative examples

Illustrative examples for the ALI and PROTAC datasets are provided in Table 6.

D Baseline Definitions and Metric Details

To evaluate HiCoMER's handling of hierarchical memory conflicts, we compare it against several baseline categories. All baselines are evaluated on the same grouped memory stream and query interface to ensure differences reflect memory maintenance and retrieval strategies rather than input format or generation capacity.

- **Standard retrieval and semantic matching:** Flat RAG ranks candidates by dense semantic similarity. Hybrid RAG combines dense and sparse (BM25) retrieval. Rerank RAG applies a cross-encoder to rerank Hybrid RAG candidates. These provide strong semantic matching baselines.

- **Recency-based memory control:** Recency RAG applies temporal decay to older memories. MemGPT-style Window retains only the most recent memories within a fixed active window, simulating recency-focused memory management.

- **Agentic and hierarchical memory systems:** Agentic Memory scores candidates by relevance, recency, and static importance. G-Memory organizes memories hierarchically and performs structured retrieval over the shared memory pool.

- **Reflection and production memory frameworks:** Self-RAG applies generation-time reflection to assess whether retrieved memories remain valid. Mem0 is a production memory infrastructure adapted to the shared-memory benchmark.

- **Internal HiCoMER variants:** Rule-Maintained HiCoMER replaces the trainable updater with rule-based maintenance. HiCoMER without validity-aware retrieval retains the maintenance module but ranks memories solely by semantic similarity.

- **Updater-only baselines:** Rule-based Updater applies deterministic revise rules. Qwen2.5-3B Zero-shot uses direct structured prompting without training. Qwen2.5-3B + SFT applies supervised fine-tuning. Qwen2.5-3B + SFT + GRPO is the full trainable updater. Llama-3.1-8B Zero-shot/SFT/SFT+GRPO provide larger-backbone variants for robustness evaluation.

Metrics are defined to assess multiple dimensions of performance:

- **Updater-specific metrics:** Conflict F1 measures accuracy in predicting conflict relations. Maintenance Accuracy measures exact-match correctness of relation and action labels.

- **Retrieval-oriented metrics:** ORR@K (Outdated Retrieval Rate) is the fraction of retrieved memories labeled outdated or non-executable. CRR@K (Consensus Retention Rate) is the recall of authoritative Team memories. NDCG@10 evaluates ranking quality with higher gain assigned to valid supporting memories.

- **Answer-level full-pipeline metrics:** ROUGE-L measures textual overlap between generated and gold answers. Decision F1 evaluates correctness on decision-oriented queries, including risk warnings, next-step decisions, and protocol consistency checks.

Table 6: Illustrative examples.

Dataset	Team Memory	Conflicting Individual/ Historical Memory	Query
ALI	Week 4 team decision: discontinue Compound X due to elevated toxicity in the pilot cohort.	Week 4 lab log: prepared Compound X for follow-up efficacy testing; Week 2 protocol note: expand Compound X testing if preliminary efficacy remains stable.	Should the team proceed with the planned Compound X follow-up experiment?
PROTAC	Week 6 platform update: replace the original degradation assay with the N-end-rule-based validation workflow for downstream screening.	Week 6 experiment log: continued using the previous degradation readout; Week 4 design note: prioritize the earlier target-screening route.	Is the current validation result directly usable under the latest platform protocol?

Table 7: Component-level evaluation of the Hierarchical Memory Conflict Updater on held-out structured maintenance instances from ALI. Conflict F1 measures the accuracy of predicting conflict relations, while Maintenance Accuracy measures the correctness of structured memory updates.

Model	Conflict F1	Maintenance Accuracy
Rule-based Updater	46.13	53.38
Qwen2.5-3B Zero-shot	57.84	60.57
Qwen2.5-3B with SFT	77.18	81.47
Qwen2.5-3B with SFT and GRPO	83.57	87.31
Llama-3.1-8B Zero-shot	61.68	64.93
Llama-3.1-8B with SFT	81.76	85.36
Llama-3.1-8B with SFT and GRPO	87.88	91.17

Conflict F1 and Maintenance Accuracy primarily assess the updater, ORR@5, CRR@5, and NDCG@10 reflect the combined effect of the updater and the validity-aware memory retriever, and ROUGE-L and Decision F1 evaluate the end-to-end quality of memory-grounded answer generation.

E Updater Evaluation

We evaluate the Hierarchical Memory Conflict Updater on held-out structured maintenance instances derived from the ALI dataset. This component-level evaluation measures whether the updater can correctly predict conflict relations and structured maintenance actions.

The results show that the trained updater substantially outperforms rule-based maintenance, indicating that hierarchical memory updating cannot be reduced to standard contradiction detection alone. Reward-based GRPO optimization further improves structured executable-state maintenance, yielding consistent gains over zero-shot and SFT-only settings.

F Results on PROTAC Dataset

We present the full main results on the PROTAC dataset in this appendix. As in the main text, the tables report end-to-end evaluations of the complete HiCoMER pipeline. Metrics ORR@5, CRR@5, and NDCG@10 primarily capture the combined effect of the Hierarchical Memory Conflict Updater and Validity-Aware Memory Retriever, while ROUGE-L and Decision F1 assess the final response quality produced by the Memory-Grounded Answer Generator.

The evaluation on the PROTAC dataset demonstrates that HiCoMER consistently reduces outdated retrieval and preserves valid memory states under complex, evolving workflow constraints. Improvements in maintained-state retrieval translate into higher-quality, memory-grounded responses, reflecting the effective coordination of hierarchical memory maintenance and validity-aware retrieval.

Table 8: End-to-end evaluation of the full HiCoMER pipeline on the PROTAC dataset using Llama-3.1-8B as the backbone for the Hierarchical Memory Conflict Updater and the Memory-Grounded Answer Generator. Retrieval metrics ORR@5, CRR@5, and NDCG@10 measure maintained-state retrieval quality, while answer-level metrics ROUGE-L and Decision F1 assess how well the pipeline converts maintained and retrieved memories into final memory-grounded responses.

Method	**ORR@5 (↓)**	**CRR@5 (↑)**	**Decision F1 (↑)**	**ROUGE-L (↑)**	**NDCG@10 (↑)**
Standard & Semantic Retrieval					
Flat RAG	48.12	48.73	38.66	33.81	40.68
Hybrid RAG	44.79	52.14	41.08	35.23	43.37
Rerank RAG	40.66	57.58	45.42	38.17	49.09
Multi-Agent & Hierarchical Memory					
G-Memory	31.84	64.77	50.31	42.38	54.34
Agentic Memory (Park)	34.87	61.92	48.79	41.14	51.83
Reflection & Production Systems					
Self-RAG	37.88	60.71	51.63	43.76	50.42
MemGPT-style Window	25.47	45.63	43.18	38.86	45.74
Mem0	30.69	66.53	53.07	45.12	55.48
HiCoMER (Llama-3.1-8B)	**15.37**	**82.74**	**62.79**	**51.43**	**66.81**

Table 9: End-to-end evaluation of HiCoMER on the PROTAC dataset using Qwen2.5-3B-Instruct for both the Hierarchical Memory Conflict Updater and the Memory-Grounded Answer Generator. Retrieval metrics ORR@5, CRR@5, and NDCG@10 assess maintained-state ranking, while answer-level metrics ROUGE-L and Decision F1 measure the quality of memory-grounded responses under a smaller backbone.

Method	**ORR@5 (↓)**	**CRR@5 (↑)**	**Decision F1 (↑)**	**ROUGE-L (↑)**	**NDCG@10 (↑)**
Standard & Semantic Retrieval					
Flat RAG	48.97	47.63	36.88	31.94	39.47
Hybrid RAG	45.74	50.82	39.42	33.76	42.03
Rerank RAG	41.48	56.07	43.61	36.87	47.58
Multi-Agent & Hierarchical Memory					
G-Memory	33.02	63.09	47.86	40.58	52.46
Agentic Memory (Park)	35.93	59.97	46.82	39.53	50.63
Reflection & Production Systems					
Self-RAG	39.06	58.88	49.18	41.74	48.66
MemGPT-style Window	26.08	43.87	40.76	36.83	44.19
Mem0	31.87	64.66	50.73	42.93	53.88
HiCoMER (Qwen2.5-3B-Instruct)	**17.58**	**79.83**	**58.12**	**48.17**	**63.96**

Table 10: Ablation study on the PROTAC dataset. ORR@5, CRR@5, and NDCG@10 measure maintained-state retrieval quality, while ROUGE-L and Decision F1 assess memory-grounded response quality. Each row shows the effect of removing a single module from the full HiCoMER pipeline.

Variant	ORR@5	CRR@5	NDCG@10	ROUGE-L	Decision F1
Full HiCoMER (Llama-3.1-8B)	15.37	82.74	66.81	51.43	62.79
w/o Hierarchical Memory Conflict Updater	31.26	70.94	57.63	45.98	54.86
w/o Validity-Aware Memory Retriever	30.11	69.82	56.34	45.21	53.91
w/o Memory-Grounded Answer Generator	15.37	82.74	66.81	46.58	54.97

Overall, these results indicate that HiCoMER provides a robust and effective framework for managing dynamic collaborative memories in mechanism-driven platform settings.

G Ablation Study on the PROTAC Dataset

To further assess module contributions across datasets, we conduct an ablation study on the PROTAC dataset. Each of the three core HiCoMER modules is removed individually: the Hierarchical Memory Conflict Updater, the Validity-Aware Memory Retriever, and the Memory-Grounded Answer Generator. Table 10 reports retrieval-oriented and answer-level metrics to evaluate how module removals affect maintained-state retrieval and final response quality.

Removing either the updater or the validity-aware retriever results in significant drops across both retrieval and answer-level metrics, confirming that these upstream modules are essential for end-to-end performance. Removing the answer generator also reduces response quality, demonstrating its role in converting retrieved memories into high-quality answers. These results complement the ALI dataset ablation study and show consistent module contributions across datasets.