# SPEAKERMEM-R1: SPEAKER-CENTERED DUAL-TRACK MEMORY FOR MULTI-PARTY DIALOGUE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-term conversational memory in multi-party settings requires more than retrieving relevant content from long-term conversations: it must distinguish who said what, whom each statement concerns, how individuals perceive one another, what information is shared by the group, and how states change over time. Recent studies on multi-party dialogue benchmarks show that existing general-purpose LLM memory systems tend to lose person and group relations or struggle to integrate clues distributed across members, groups, and time. Together, these issues reveal two core bottlenecks: message attribution and relational understanding in multi-party dialogue, and state reconstruction from interleaved histories. To address both, we propose **SPEAKERMEM-R1**: its dual-track memory stores speaker-labeled verbatim messages and derived states organized into person-level and group-level views, then combines evidence from both tracks by entity, event, and time at query time. To reduce attribution and update errors during structured memory construction while enabling local deployment, we train Writer-R1 with SpeakerLevenshtein and speaker-conditioned GRPO. On GroupMemBench, SocialMemBench, and EverMemBench, SPEAKERMEM-R1 achieves binary accuracies of 47.9%, 69.2%, and 61.9%, respectively. The scores are 3.3, 12.4, and 9.4 percentage points over the best results of mainstream frameworks evaluated on each benchmark, respectively. On the publicly reported EverMemBench leaderboard from EverMind-AI, SPEAKERMEM-R1 achieves 62.33%, the best reported result among the latest state-of-the-art frameworks. It also achieves 70.85% on all 1,986 LoCoMo questions, which we use as a two-person long-term conversation boundary test. In a controlled evaluation of 305 questions, RL raises the SFT Writer's mean accuracy from 57.38% to 68.20% under a frozen query/answer pipeline. We report both binary accuracy and token-F1, and ablations show that the verbatim and structured tracks, as well as person-level and group-level views, are complementary under the standardized evaluation interface.

## 1 INTRODUCTION

Long-term conversational memory enables language agents to retain facts, preferences, and social relations across sessions (Zhong et al., 2024; Park et al., 2023; Packer et al., 2023; Wu et al., 2025). Existing work mainly targets single-user or two-person histories, splitting, compressing, indexing, and retrieving conversations by relevance (Maharana et al., 2024; Wu et al., 2025). Multi-party group chats additionally contain speaker relations, reply structure, cross-topic branches, and state revisions, so they cannot be flattened into a message stream that is simply compressed in order (Ghosal et al., 2019; Li et al., 2020). GroupMemBench, SocialMemBench, and EverMemBench further show that general-purpose memory systems degrade substantially in multi-party or long-term conversation settings, while BM25 or dense retrieval can remain competitive in some configurations (Yang et al., 2026; Owolabi, 2026; Hu et al., 2026b). The problem is therefore not merely finding relevant text, but preserving and recovering relations and historical structure in multi-party dialogue.

We organize this problem around two coupled challenges: **message attribution**, which distinguishes who said what, whom the content concerns, and whether information is personal or shared; and **state reconstruction**, which recovers current or historical states from clues distributed across members, groups, and time (Ghosal et al., 2019; Li et al., 2020; Owolabi, 2026; Hu et al., 2026b). Global relevance retrieval does not guarantee that a low-frequency member or the correct branch enters



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

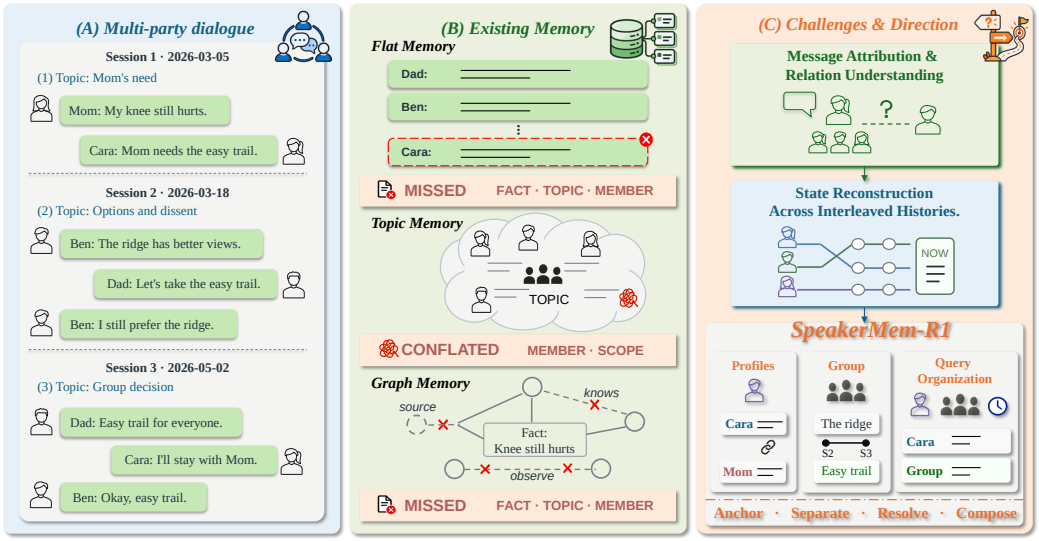


Figure 1: A multi-party group chat is not a flat message stream: participants discuss different topic branches, express different positions, refer to one another, and revise earlier states. Flat retrieval, aggregate summaries, and generic graph structures can lose these relations. SPEAKERMEM-R1 preserves attribution and reconstructs state with dual-track memory and the Anchor–Separate–Resolve–Compose query procedure.

a limited candidate set, while summaries, fact aggregation, and generic memory graphs may lose source/owner, PERSON/GROUP scope, or state versions (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Lewis et al., 2020; Zhong et al., 2024; Chhikara et al., 2025; Yue et al., 2026; Xu et al., 2026; Gutiérrez et al., 2024; Tang et al., 2026). Multi-party memory therefore needs verifiable evidence with participant relations and query-conditioned local state reconstruction, rather than simply more summaries or graph edges.

We propose SPEAKERMEM-R1, a dual-track system that preserves speaker-labeled messages alongside provenance-linked derived states organized into person-level and group-level views. At query time, Anchor–Separate–Resolve–Compose retrieves and organizes evidence by participant, event, and time. We train a locally deployable Writer with SpeakerLevenshtein and speaker-conditioned GRPO to reduce attribution and update errors while freezing query and answer modules.

On GroupMemBench, SocialMemBench, and EverMemBench, SPEAKERMEM-R1 achieves binary accuracies of 47.9%, 69.2%, and 61.9%, respectively. Compared with the best results of mainstream frameworks evaluated on each benchmark, the scores are higher by 3.3, 12.4, and 9.4 percentage points, respectively. On the publicly reported EverMemBench leaderboard from EverMind-AI, SPEAKERMEM-R1 achieves 62.33%, the best reported result among the latest state-of-the-art frameworks. In a 305-question controlled evaluation under the main protocol, the Qwen2.5-3B Writer-R1 reaches 68.20%, 10.82 points above SFT and within 3.28 points of the 71.48% LLM writer reference; ablations confirm the complementary roles of both tracks and structured views.

Our contributions are as follows:

- We design SPEAKERMEM-R1, a dual-track memory system that combines traceable verbatim messages with person-level and group-level structured views for attribution, scope control, and state reconstruction in multi-party dialogue;
- We provide an analysis framework derived from question requirements and recurring error patterns in three multi-party benchmarks, covering member coverage, information attribution, personal/-group scope, term and event disambiguation, and temporal updates with multi-hop reasoning;
- We train a locally deployable Qwen2.5-3B Writer with SpeakerLevenshtein and speaker-conditioned GRPO; on 305 controlled questions, RL writer reaches 95.4% of the LLM writer reference accuracy, while the dual-track design and writing objective are evaluated on three multi-party benchmarks, LoCoMo, and targeted ablations.

## 2 Related Work

**Long-term conversational memory and multi-party memory benchmarks.** LoCoMo and LongMemEval evaluate factual, temporal, multi-hop, knowledge-update, and abstention abilities in long-term conversations (Maharana et al., 2024; Wu et al., 2025); MemBench, MemoryAgentBench, StoryBench, and REALTALK extend this scope to reflective memory, test-time learning, dynamic branches, and real-world interaction (Tan et al., 2025; Hu et al., 2025; Wan & Ma, 2025; Lee et al., 2025). In multi-party settings, DialogueGCN and Molweni establish the importance of speaker relations and discourse structure (Ghosal et al., 2019; Li et al., 2020), while GroupMemBench, SocialMemBench, and EverMemBench evaluate group dynamics, social relations and norms, and cross-group collaboration with evolving states (Yang et al., 2026; Owolabi, 2026; Hu et al., 2026b). These benchmarks show that group-chat evidence is distributed and updated across members, groups, and time rather than reducible to a flat message sequence.

**Retrieval augmentation, structured memory, and memory graphs.** BM25, dense retrieval, and RAG/REALM/RETRO provide lexical or semantic matching but generally do not constrain member coverage, information attribution, or event consistency (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022). MemoryBank, Mem0, A-MEM, MemGPT, generative agents, MemoBase, MemOS, and Zep study persistent facts, updates, connections, profiles, and temporal knowledge organization (Zhong et al., 2024; Chhikara et al., 2025; Xu et al., 2025; Packer et al., 2023; Park et al., 2023; memodb-io, 2024; Li et al., 2025a; Rasmussen et al., 2025), whereas MemoryLLM and M+ write memory into latent spaces (Wang et al., 2024; 2025a). EverMemOS/EverOS, RippleMem, MIRIX, HippoRAG, RAPTOR, GraphRAG, LightMem, StructMem, Mnemis, HyperMem, and LightRAG further use consolidation, event graphs, multi-agent collaboration, recursive summaries, and graph or hypergraph retrieval (Hu et al., 2026a; EverMind researchers, 2026; Ji et al., 2026; Wang & Chen, 2025; Gutiérrez et al., 2024; Sarthi et al., 2024; Edge et al., 2024; Fang et al., 2026; Xu et al., 2026; Tang et al., 2026; Yue et al., 2026; Guo et al., 2024). These methods improve compression and association, but do not necessarily preserve key information in multi-party group chats, such as message attribution, cross-person cognition, and group consensus, among other aspects.

**Learnable memory management.** Memory-R1, Mem-$\alpha$, Agentic Memory, DeltaMem, and Memory-R2 learn memory actions, hierarchical construction, multi-step management, state-difference rewards, or long-horizon credit assignment (Yan et al., 2026b; Wang et al., 2025b; Yu et al., 2026; Zhang et al., 2026; Yan et al., 2026a). CoMAM jointly optimizes memory agents (Mao et al., 2026), and DeferMem trains query-time evidence distillation (Yin & Tang, 2026). In contrast, R$^2$-Mem uses RL-free reflective search (Wang et al., 2026); G-Memory organizes multi-agent experience in a hierarchy (Zhang et al., 2025); Reflexion uses verbal feedback without updating model weights (Shinn et al., 2023). Our focus is owner-level writing supervision in group chats, training only the Writer while retaining fixed retrieval and answering modules.

## 3 Method

The overall architecture is shown in Figure 2. Given a multi-party message stream with speaker, time, and channel information, SpeakerMem-R1 first writes the messages into a traceable dual-track memory, then reconstructs query-specific evidence from the two tracks, and finally passes the evidence to a frozen answerer. The Writer is the component responsible for converting local messages into structured memory actions; the retrieval and answering stages operate on the resulting memory.

### 3.1 Problem Definition: From Relevance Retrieval to Attributed Evidence

Given a dialogue stream with $K \geq 2$ participants,

$$D = \{u_t\}_{t=1}^{T}, \qquad u_t = (x_t, s_t, \tau_t, c_t), \tag{1}$$

where $x_t, s_t, \tau_t, c_t$ denote the text, speaker, time, and channel, respectively. The system first writes memory $M = W_\theta(D)$, then retrieves evidence and generates an answer with a frozen answerer:

$$E_q = R(q; M), \qquad \hat{y} = A(q, E_q). \tag{2}$$

Ordinary retrieval optimizes only the relevance between evidence and $q$. Multi-party QA additionally requires evidence to be mutually compatible in participants, attribution, scope, event, and time.

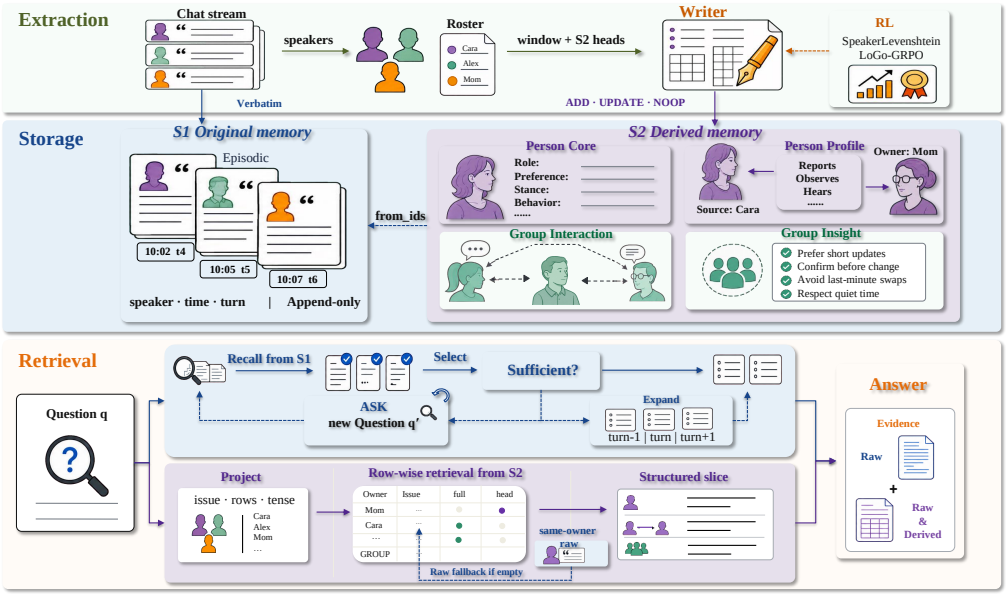


Figure 2: Overall architecture of SPEAKERMEM-R1. The system writes traceable System 1 verbatim memory and System 2 structures with participant, event, and state information, selects evidence from the two tracks at query time, and organizes it into the evidence set required for answering. RL rewards update only the Writer, while the answerer remains frozen.

Because future questions are unknown at write time, a single summary or fixed event structure may fail to recover the evidence scope required by a query.

We represent a derived record as

$$r = (x, \mathrm{src}, \mathrm{own}, \mathrm{scope}, \mathrm{event}, \mathrm{time}, \mathrm{state}, \mathrm{ref}), \tag{3}$$

where $x$ is the content, src and own are the source and owner, $\mathrm{scope} \in \{\textsc{Person}, \textsc{Group}\}$ is the scope, and the final four fields denote event, time, state, and a source reference. A self-report typically satisfies $\mathrm{src} = \mathrm{own}$, whereas "Alice believes that Bob has agreed" satisfies $\mathrm{src} = \text{Alice}, \mathrm{own} = \text{Bob}$. Thus, source–owner distinguishes who provides the information from whom the content concerns.

**End-to-end execution.** The system first writes memory and then answers queries. Messages enter System 1 verbatim, the Writer reads each local segment together with the roster and current System 2 state, and deterministic code validates the actions and adds provenance; at query time, Project produces query constraints, the two tracks retrieve and Compose combines evidence, and the frozen answerer produces the answer.

**Online writing and non-destructive updates.** System 1 appends messages without a language model and retains their source coordinates. The Writer reads each local segment and the current heads of the derived layers, returns ADD, UPDATE, or NOOP, and deterministic code validates the action, adds provenance, and writes System 2. UPDATE cites an existing `entry_id`, appends a new node while reusing the owner, source, and layer coordinates of the record identified by that `entry_id`, and connects the states through `links/superseded_by` without overwriting history; the resulting chain supports head and full queries. Detailed constraints are given in Appendix B.

### 3.2 FIVE-LAYER TRACEABLE DUAL-TRACK MEMORY

SPEAKERMEM-R1 retains two complementary tracks. System 1 is the only verbatim layer and stores each message with its text, speaker, time, and channel. System 2 is a four-layer derived structure: PERSON-scoped Core (stable identity, facts, stances, and recurring behavior) and Profile (observations about a person or cross-person cognition), plus GROUP-scoped Interaction (cross-speaker events, relations, and decisions) and Insight (group norms, consensus, and exceptions). PERSON/GROUP fixes the record scope, source/owner separates who provides information from

whom it concerns, and `from_ids` links every derived record to supporting messages. The verbatim track therefore supplies exact wording and local context, while the derived track supplies person-level and group-level state views; the full fields, code-level layer names, and query roles are given in Appendix Table 5.

### 3.3 Query-Conditioned Evidence

The five-layer store retains only information that can be recomposed; the final evidence set is query-dependent. Project compiles the query and deterministic roster into common query constraints:

$$Q_q = (\mathrm{rows}(q), \mathrm{issue}(q), \mathrm{mode}(q), \mathrm{scope}(q)). \tag{4}$$

Here $\mathrm{rows}(q)$ contains the PERSON/GROUP rows to address, $\mathrm{issue}(q)$ is the issue or event constraint, $\mathrm{mode}(q) \in \{\mathrm{head}, \mathrm{full}\}$ is the temporal mode, and $\mathrm{scope}(q)$ is the source–owner constraint. For example, "every person" expands all roster rows, "the final decision" selects GROUP rows, and "Alice's view of Bob" fixes source=Alice and owner=Bob.

**Two retrieval paths and four-step organization.** System 1 retrieves exact wording and local context from the verbatim track, optionally expands neighboring messages with Expand, and performs one supplementary search through Sufficiency/ASK when needed. System 2 first expands PERSON/GROUP rows according to $\mathrm{rows}(q)$, then selects records within each row by issue, relation, event, and time:

$$C_p(q) = \{r \in M^{\mathrm{der}} : \ \mathrm{own}(r) = p, \ \mathrm{match}(r, q) = 1\}, \tag{5}$$

$$V_q[p] = \mathrm{Temporal}_{\mathrm{mode}(q)}\big(\mathrm{Select}(C_p(q))\big), \qquad p \in \mathrm{rows}(q). \tag{6}$$

The two retrieval paths use independent budgets. When a derived row is empty, the system falls back to the corresponding person's verbatim messages; if evidence is still unavailable, it preserves an explicit empty row. Candidate budgets and supplementary retrieval details are given in Appendix A. As an abstract summary of the system's behavior, **Anchor** retains source, owner, event, time, and source provenance; **Separate** expands PERSON/GROUP rows from the roster; **Resolve** distinguishes issues, parallel events, and current versus historical versions within each row; and **Compose** organizes the two evidence paths along persons, relations, and update chains before passing them to the frozen answerer. These four operations summarize the preceding write and query behavior rather than introducing an additional execution stage.

Appendix Table 14 maps five analysis dimensions to these mechanisms. The dimensions summarize benchmark question requirements and recurring error patterns, rather than independently annotated diagnostic labels.

## 4 Writer Training Method

At the system level, the Writer is model-agnostic; this section separately studies RL as a way to replace an expensive prompt-based Writer with a locally deployable small model. RL trains the Add/Update/Noop decisions of Qwen2.5-3B, while System 1 writing, query organization, and answering remain frozen. Local structural signals identify owner-level writing errors, while terminal QA gain measures their downstream effect after the conversation has been written.

**Evaluating structured states by owner.** Inspired by DeltaMem's state-level memory matching (Zhang et al., 2026), *SpeakerLevenshtein* combines token-level F1 with a normalized sequence-matching rate, rather than standard edit distance. It performs coordinate-consistent one-to-one matching within owner buckets: personal records, GROUP records, and cross-person observations cannot cancel one another, while omissions and over-writing are penalized. Let $P$ be the owner set and $F_p$ the matching result for owner $p$; the local structural potential is

$$\Phi_{\mathrm{SL}}(M, M^\star) = w_1 \frac{1}{|P|} \sum_{p \in P} F_p + w_2 \min_{p \in P} F_p. \tag{7}$$

The macro-average term measures the overall state, while the worst-owner term prevents frequent people from masking infrequent people or GROUP. We use $w_1 = 0.80$ and $w_2 = 0.20$. Detailed source/owner/layer gating, soft matching, and Hungarian alignment are given in Appendix E.

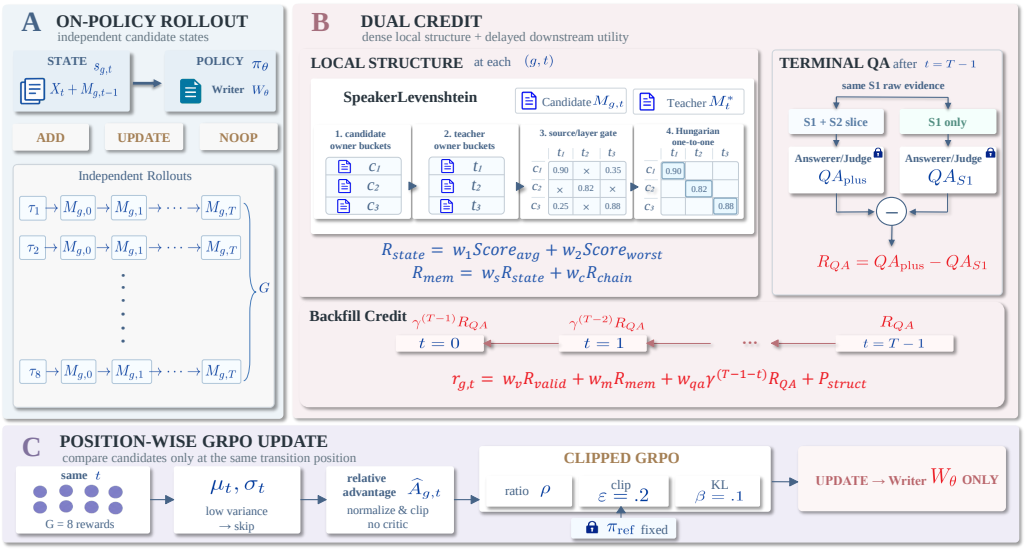


Figure 3: Speaker-Conditioned LoGo-GRPO training loop for Writer-R1. We sample $G = 8$ writing trajectories for the same conversation. At each position, owner-decomposed SpeakerLevenshtein signals and terminal QA gains form group-relative advantages; clipped GRPO updates are applied only to the Writer, while retrieval and the answerer remain frozen.

**Local-to-global returns.** Memory-R2's LoGo-GRPO combines global optimization with local rerollouts from shared memory states (Yan et al., 2026a). Our speaker-conditioned variant instead compares aligned writing positions, using owner-level state scores and terminal QA gain. The local signal evaluates UPDATE transitions and structural validity. The global signal measures the gain of System 1+System 2 over System 1-only at $M_{g,T}$, under the same System 1 evidence:

$$R_g^{\mathrm{QA}} = \mathrm{QA}(\text{System 1} + \text{System 2}; M_{g,T}) - \mathrm{QA}(\text{System 1}; M_{g,T}). \tag{8}$$

This difference does not attribute questions already answerable from verbatim memory to the Writer. The return at writing position $t$ on trajectory $g$ is

$$r_{g,t} = w_{\mathrm{valid}} R_{g,t}^{\mathrm{valid}} + w_{\mathrm{mem}} R_{g,t}^{\mathrm{mem}} + P_{g,t} + w_{\mathrm{QA}} \gamma^{T-1-t} R_g^{\mathrm{QA}}, \qquad \gamma = 0.95. \tag{9}$$

We use $w_{\mathrm{valid}} = 0.20$, $w_{\mathrm{mem}} = 0.45$, and $w_{\mathrm{QA}} = 0.35$. We sample multiple trajectories for the same network, compute group-relative advantages only at the same writing positions, and update the Writer with clipped GRPO (Shao et al., 2024). Positions with no effective within-group variation are omitted from the update. Reward decomposition, advantage computation, KL constraints, training data, and hyperparameters are given in Appendix E.

## 5 Experiments

### 5.1 Research Questions and Evaluation Protocol

We investigate whether dual-track memory improves multi-party QA, how individual components contribute to performance, and which types of errors remain.

We evaluate GroupMemBench (745 questions), SocialMemBench (1,031), and EverMemBench (2,400) (Yang et al., 2026; Owolabi, 2026; Hu et al., 2026b) against BM25, dense retrieval, Mem0, A-MEM, HippoRAG, and Full context when feasible (Chhikara et al., 2025; Xu et al., 2025; Gutiérrez et al., 2024); LoCoMo (1,986 questions) serves as a two-person long-term conversation boundary test. Evaluation proceeds through memory construction, question-conditioned retrieval, and answer generation. The DeepSeek-V4-Flash and GPT-5.6-luna configurations switch the language model across these stages to assess cross-model robustness. For the primary three-benchmark comparison, each system follows its official code, recommended configuration, and official prompt; we standardize

Table 1: Main results (%). GM/SM denote GroupMemBench/SocialMemBench; EM-MC/OE are EverMemBench subsets, while EM-All is the complete question set. Paired cells report Acc./token-F1; MeanQ weights questions and MeanN weights networks. The two model blocks test cross-model robustness across memory construction, retrieval, and answering under the evaluation protocol in Section 5. †: ASK disabled; Full context is feasible only on SM. Bold/underline mark the best/second-best non-Full-context result per block.

| Model | Method | GM | SM | MeanQ | MeanN | EM-MC | EM-OE | EM-All |
|---|---|---|---|---|---|---|---|---|
| DeepSeek-V4-Flash | BM25 | 44.6/25.0 | 28.6/15.7 | .234 | .250 | 64.4/3.6 | 27.0/**23.3** | 52.5/9.9 |
| | Embed | 34.5/17.4 | 38.1/18.1 | .300 | .307 | 56.1/4.0 | 23.9/22.7 | 45.9/10.0 |
| | Mem0 | 21.6/9.0 | 13.7/11.0 | .132 | .130 | 28.4/5.0 | 1.7/7.0 | 19.9/6.0 |
| | A-MEM | 27.1/10.6 | 56.8/31.9 | .608 | .610 | 37.7/14.2 | 8.8/12.9 | 28.5/13.8 |
| | HippoRAG | 27.0/11.1 | 55.9/30.6 | .584 | .574 | 62.9/18.5 | 15.9/18.4 | 48.0/18.5 |
| | Full context | — | 69.4/26.1 | .592 | .573 | — | — | — |
| | SpeakerMem-R1† | 47.0/25.2 | **69.2**/27.4 | **.713** | .691 | 71.2/33.7 | 37.4/16.7 | 60.5/**28.3** |
| | SpeakerMem-R1 | **47.9/26.5** | 64.9/**32.7** | .710 | **.693** | **72.0/33.8** | **40.0**/15.7 | **61.9**/28.1 |
| GPT-5.6-luna | BM25 | **46.2**/26.6 | 30.0/21.1 | .412 | .452 | 69.6/33.0 | 27.8/**24.4** | 56.3/30.3 |
| | Embed | 35.3/18.4 | 36.4/22.7 | .519 | .538 | 58.5/28.8 | 24.5/23.3 | 47.8/27.1 |
| | Mem0 | 20.27/8.1 | 13.97/14.9 | .252 | .262 | 35.0/18.7 | 1.4/9.6 | 24.38/15.8 |
| | A-MEM | 26.58/9.8 | 46.56/24.9 | .607 | .601 | 53.4/27.4 | 11.8/16.9 | 40.17/24.1 |
| | HippoRAG | 27.11/11.9 | 61.30/**28.0** | .726 | **.723** | **75.1**/35.2 | 20.1/21.9 | 57.62/**30.9** |
| | Full context | — | 71.3/29.6 | .769 | .750 | — | — | — |
| | SpeakerMem-R1 | 42.7/**26.8** | **64.4**/24.1 | **.731** | .715 | 71.3/**35.6** | **35.8**/16.4 | **60.0**/29.5 |

only the data, metric definitions, judge model, and evaluation interface, while retaining speaker/source metadata where supported. The public EverMemBench comparison uses a separate configuration specified in Table 2. Our primary metric is question-level binary accuracy (Acc.; Appendix Eq. (10)), which measures task-level correctness; supplementary token-F1 (Appendix Eq. (11)) measures judge-independent token overlap. SocialMem additionally reports MeanQ/MeanN (Appendix Eq. (12)), which aggregate official rubric scores with equal weights for questions and networks, respectively. Acc. and token-F1 are reported as percentages, whereas MeanQ/MeanN lie in $[0, 1]$; execution settings and category results appear in Appendices C–D and F.

### 5.2 Main Multi-Party Results

Table 1 shows highest SpeakerMem-R1 accuracies of 47.9%, 69.2%, and 61.9% on GroupMem, SocialMem, and EverMem, respectively, when selecting the highest-accuracy SpeakerMem-R1 configuration for each benchmark. Among the non-Full-context mainstream baselines on each benchmark, the best results are 44.6% from BM25, 56.8% from A-MEM, and 52.5% from BM25, respectively. Compared with the best results of mainstream frameworks evaluated on each benchmark, the scores are higher by 3.3, 12.4, and 9.4 percentage points, respectively. Full context is feasible only on SocialMemBench; the other histories cannot reliably fit within the configured context window. On SocialMem, the 69.2% result is near Full context at 69.4%; the full system obtains MeanQ/MeanN 0.710/0.693 with a network-level 95% CI of $[0.659, 0.726]$. The small MeanQ/MeanN reversal between the two variants reflects their different weighting of questions and networks.

**Publicly reported EverMemBench leaderboard from EverMind-AI.** On the publicly reported EverMemBench leaderboard from EverMind-AI, Table 2 compares all 2,400 EverMemBench questions using the GPT-4.1-mini/Gemini-3-Flash configuration (EverMind researchers, 2026). SpeakerMem-R1 answers 1,496/2,400 questions correctly (62.33% accuracy), versus approximately 60.08% for EverOS and 54.75% for RippleMem; public totals are reconstructed from rounded category results, whereas ours uses question-level records. Strong Single, Const, Proact, and Update results contrast with weaker Multi, Skill, and Role scores, exposing cross-evidence, preference, and role-attribution bottlenecks.

The accuracy gains depend on the task. On GroupMem, BM25, the global lexical-retrieval baseline, already reaches 44.6%, while SpeakerMem-R1 obtains 47.9%. On SocialMem, A-MEM is the strongest mainstream baseline at 56.8%, followed by HippoRAG at 55.9%; the highest

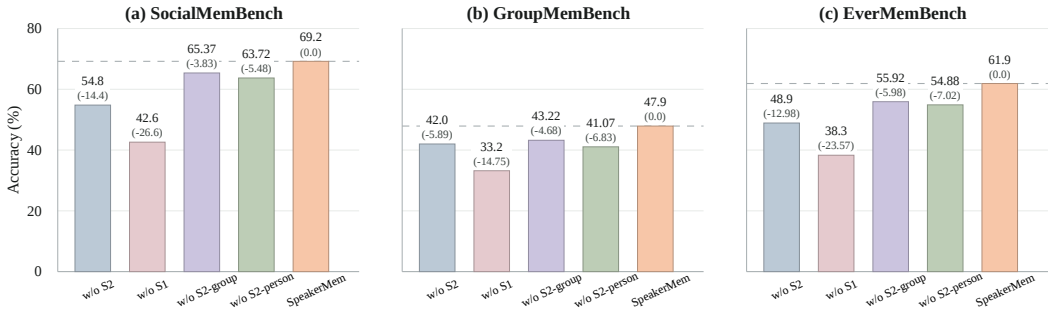


Figure 4: Memory-path and hierarchy ablations. Labels show accuracy and change from the best full dual-track result. Here, w/o S2 removes all four System 2 derived layers, w/o S1 removes the verbatim track, w/o S2-group removes the two GROUP layers (Interaction and Insight), and w/o S2-person removes the two PERSON layers (Core and Profile). SpeakerMem denotes the full dual-track system.

Table 2: EverMemBench accuracy (%) across nine public behavior labels and the question-weighted total over 2,400 questions. All methods use GPT-4.1-mini for answering and Gemini-3-Flash for judging; SPEAKERMEM-R1 uses System 1 top-10 and System 2 owner-2 + source-1. Bold/underline indicate best/second-best.

| Method | Single | Multi | Temp | Const | Proact | Update | Style | Skill | Role | Weighted total |
|---|---|---|---|---|---|---|---|---|---|---|
| MemoBase | 60.09 | 12.85 | 18.00 | 64.68 | 36.77 | 30.60 | 17.05 | 29.59 | 38.78 | 36.21 |
| Mem0 | 55.40 | 11.24 | 6.33 | 66.17 | 52.46 | 51.87 | 22.73 | 31.36 | 36.22 | 39.92 |
| Zep | 73.71 | 8.03 | 13.00 | 67.16 | 47.54 | 43.66 | 26.70 | 35.50 | 44.39 | 41.67 |
| MemOS | 71.36 | 18.88 | 15.67 | 69.90 | 51.99 | 45.15 | 28.98 | 32.54 | 48.47 | 44.63 |
| RippleMem | 92.02 | 22.09 | <u>21.33</u> | 78.11 | <u>71.66</u> | 58.96 | 31.25 | 36.69 | **53.06** | 54.75 |
| EverOS | **94.37** | **28.11** | 20.33 | <u>86.07</u> | 68.62 | **84.70** | <u>39.77</u> | **42.60** | <u>52.04</u> | <u>60.08</u> |
| SPEAKERMEM-R1 | <u>93.43</u> | <u>24.10</u> | **35.00** | **87.06** | **75.64** | <u>80.22</u> | **50.57** | <u>40.83</u> | 43.88 | **62.33** |

SPEAKERMEM-R1 accuracy shown reaches 69.2%. On EverMem open-ended questions, BM25 is the strongest comparison at 27.0%, whereas SPEAKERMEM-R1 achieves 40.0%, a 13.0-point gain. Binary accuracy and token-F1 are not monotonic: the full system has lower SocialMem accuracy than † but higher token-F1 (32.7 versus 27.4), because extra people or incorrect scope can invalidate an otherwise overlapping answer. We therefore report both.

### 5.3 DUAL-TRACK AND HIERARCHICAL ABLATIONS

Figure 4 compares path and hierarchy ablations against the better full-system result with or without ASK. Removing either track reduces accuracy on all three benchmarks; retaining only per-speaker or group records also degrades performance, showing complementary evidence at both levels. Exact results appear in Appendix Table 22.

With ASK disabled, SPEAKERMEM-R1* obtains accuracies of 47.0/69.2/60.5% on GroupMem/SocialMem/EverMem, versus 47.9/64.9/61.9% with ASK. Supplementary retrieval recovers dispersed clues on GroupMem and EverMem but can add redundant evidence on SocialMem.

We additionally analyze joint S1/S2 retrieval top-$k$ sensitivity, with response curves, a dual-metric grid, and detailed discussion in Appendix H (Figures 5 and 6).

### 5.4 DOES THE FULL R1 OBJECTIVE IMPROVE WRITING?

On 10 held-out SocialMem networks (305 questions), using the main no-ASK query configuration (System 1 recall-$n = 40$ with final top-$k = 10$, System 2 $k$ = 2/source-$k = 1$), the Qwen2.5-3B Writer-R1 reaches $68.20 \pm 0.66\%$ with the query and answer modules frozen, up from $57.38 \pm 0.33\%$ for SFT: a gain of 10.82 percentage points, or 33 additional correct answers (Table 3). Under the same protocol, the LLM writer reference reaches $71.48 \pm 0.66\%$ (218/305), leaving R1 3.28 points behind

Table 3: Controlled Writer results on 10 held-out complete SocialMem networks (305 questions). All rows use the same no-ASK query protocol: System 1 recalls $n = 40$ candidates and retains final top-$k = 10$; System 2 uses $k = 2$/source-$k = 1$. Query and answer modules are frozen; Acc. reports mean±sample standard deviation over three runs.

| Writing strategy | Writer | Correct/total | Acc. (%, mean±std.) | Gap to LLM writer |
|---|---|---|---|---|
| SFT (epoch 10) | Qwen2.5-3B | 175/305 | $57.38 \pm 0.33$ | $-14.10$ pp |
| **Writer-R1 (30 steps)** | Qwen2.5-3B | 208/305 | $\mathbf{68.20 \pm 0.66}$ | $-3.28$ pp |
| LLM writer | DeepSeek-V4-Flash | 218/305 | $71.48 \pm 0.66$ | – |

Table 4: LoCoMo category accuracy (%). The metric is LLM-as-a-judge accuracy (%) scored by GPT-4o-mini. LightRAG uses its official implementation (Guo et al., 2024). Mem0, A-MEM, and Zep scores come from Tables 1–2 of Chhikara et al. (2025); MemOS scores come from Table 3 of Li et al. (2025b). ALL(Non-AD) aggregates the four non-AD categories. Bold/underline mark best/second-best.

| Method | Single-hop | Multi-hop | Temporal | Open-domain | ALL(Non-AD) |
|---|---|---|---|---|---|
| Mem0 | 67.13 | 51.15 | 55.51 | <u>72.93</u> | 66.88 |
| A-MEM | 39.79 | 18.85 | 49.91 | 54.05 | 48.38 |
| MemOS | <u>81.09</u> | <u>67.49</u> | **75.18** | 55.90 | <u>75.80</u> |
| Zep | 61.70 | 41.35 | 49.31 | **76.60** | 65.99 |
| LightRAG | **86.68** | **84.04** | 60.75 | 71.88 | **79.87** |
| SPEAKERMEM-R1 | 77.88 | 41.13 | <u>70.72</u> | 40.62 | 67.34 |

and at 95.4% of the LLM writer accuracy. This result shows that RL brings a locally deployable small Writer close to the LLM writer reference, while it is not evidence of broad cross-domain RL generalization. Training-data composition, hyperparameters, and per-seed results are reported in Appendix E.

**LoCoMo boundary test.** SPEAKERMEM-R1 achieves an accuracy of 70.85% (1,407/1,986) on two-person LoCoMo (Table 4); multi-hop and open-domain questions remain weak. Full counts appear in Appendix G, and this test does not replace multi-party validation.

## 6 CONCLUSION

SPEAKERMEM-R1 uses a simple, efficient, and traceable dual-track memory to address the attribution and state-reconstruction failures that general-purpose memory systems exhibit in long-term multi-party conversations. It preserves source-linked verbatim evidence, organizes derived states into person-level and group-level views, and combines the two tracks through constrained query-time composition. This design directly targets multi-party memory failure modes without relying on a complex collection of additional mechanisms. Three multi-party benchmarks and LoCoMo demonstrate the design's effectiveness and limits; ablations establish complementary contributions from the two tracks and person-level and group-level views. In the 305-question controlled study under the main protocol, the RL Writer (30 steps) reaches 95.4% of the 71.48% LLM writer reference accuracy. This result shows that our RL method can make a locally deployable small Writer approach the performance of the large Writer while keeping the memory pipeline practical. Broader RL generalization, full-member coverage, cross-evidence reasoning, and knowledge outside memory remain open challenges.

**Limitations and future work.** The system assumes reliable rosters, source/owner attribution, and temporal identification; aliases, membership changes, implicit audiences, and parallel events remain challenging. Future work should reduce construction and retrieval cost and improve cross-evidence reasoning, open-domain QA, and generalization across domains and languages.

## AI USE STATEMENT

Generative AI assisted writing and language polishing under our review. Language models also generated Writer teacher trajectories, ran the frozen answerer, and performed automatic judging. They did not create formal benchmark data; we authored the manual UPDATE/NOOP networks. We checked data processing, statistics, interpretation, and conclusions, and take responsibility for all submitted materials.

## REPRODUCIBILITY STATEMENT

The implementation and evaluation entry points are summarized in Appendix A, which fixes the ingestion, retrieval, ASK, and System 2 budgets. Appendix B specifies the record schema, ADD/UPDATE/NOOP actions, update chains, and query composition. Dataset sizes, model roles, judges, metrics, token accounting, and experimental boundaries are collected in Appendices C and D; category-level results are in Appendix F. Appendix E gives the Writer-R1 reward terms, GRPO/KL settings, training-network split, and controlled per-seed results. Complete ablations, LoCoMo categories, retrieval-budget sensitivity, and exact prompts are provided in Appendices F.6, G, H, and I, respectively.

## REFERENCES

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 2206–2240, Baltimore, Maryland, USA, 2022. PMLR. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In *Proceedings of the 27th European Conference on Artificial Intelligence*, pp. 2993–3000, Bologna, Italy, 2025. IOS Press. doi: 10.3233/FAIA251160.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. arXiv preprint arXiv:2404.16130, 2024. URL https://arxiv.org/abs/2404.16130.

EverMind researchers. Multi-round retrieval: Letting the model decide when to stop searching. EverMind Blog, September 4 2026. URL https://evermind.ai/blogs/multi-round-retrieval-letting-the-model-decide-when-to-stop-searching. Accessed September 21, 2026.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *Proceedings of the International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-Abstract-Conference.html.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 154–164, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1015.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. arXiv preprint arXiv:2410.05779, 2024. URL https://arxiv.org/abs/2410.05779.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems 37*, pp. 59532–59569, Vancouver, Canada, 2024. Neural Information Processing Systems Foundation. doi: 10.52202/079017-1902. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3929–3938, Virtual, 2020. PMLR. URL https://proceedings.mlr.press/v119/guu20a.html.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 45836–45853, San Diego, California, USA, 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.2125. URL https://aclanthology.org/2026.acl-long.2125/.

Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Yi Bai, Dannong Xu, Tianwei Lin, Xiaohong Li, Yunyun Han, Jian Pei, and Yafeng Deng. Evaluating long-horizon memory for multi-party collaborative dialogues. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 9082–9090. Association for Computing Machinery, 2026b. doi: 10.1145/3770855.3817589.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. arXiv preprint arXiv:2507.05257, 2025. URL https://arxiv.org/abs/2507.05257.

Jingbo Ji, Lingyi Li, Xilong Cheng, Yuhao Zhou, Wenji Zhang, Yuting Tan, and Yunxiao Qin. RippleMem: From isolated retrieval to associative recollection for long-term agent memory. arXiv preprint arXiv:2608.13334, 2026. URL https://arxiv.org/abs/2608.13334.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550.

Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. REALTALK: A 21-day real-world dataset for long-term conversation. arXiv preprint arXiv:2502.13270, 2025. URL https://arxiv.org/abs/2502.13270.

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. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33*, pp. 9459–9474, Vancouver, Canada, 2020. Neural Information Processing Systems Foundation. URL https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. Molweni: A challenge multiparty dialogue-based machine reading comprehension dataset with discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 2642–2652, Barcelona, Spain, 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.238. URL https://aclanthology.org/2020.coling-main.238/.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. MemOS: An operating system for memory-augmented generation (MAG) in large language models. arXiv preprint arXiv:2505.22101, 2025a. URL https://arxiv.org/abs/2505.22101.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhen Tao, Huayi Lai, Hao Wu, Bo Tang, Zhengren Wang, Zhaoxin Fan, Ningyu Zhang, Linfeng Zhang, Junchi Yan, Mingchuan Yang, Tong Xu, Wei Xu, Huajun Chen, Haofen Wang, Hongkang Yang, Wentao Zhang, Zhi-Qin John Xu, Siheng Chen, and Feiyu Xiong. MemOS: A memory OS for AI system. arXiv preprint arXiv:2507.03724v4, 2025b. URL https://arxiv.org/abs/2507.03724v4.

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

Wenyu Mao, Haoyang Liu, Zhao Liu, Haosong Tan, Yaorui Shi, Jiancan Wu, An Zhang, and Xiang Wang. Collaborative multi-agent optimization for personalized memory system. arXiv preprint arXiv:2603.12631v1, 2026. URL https://arxiv.org/abs/2603.12631v1.

memodb-io. MemoBase: User profile-based long-term memory for AI chatbot applications. GitHub software repository, 2024. URL https://github.com/memodb-io/memobase. Accessed September 21, 2026.

Olukunle Owolabi. SocialMemBench: Are AI memory systems ready for social group settings? arXiv preprint arXiv:2605.17789, 2026. URL https://arxiv.org/abs/2605.17789.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 10:1–10:22, San Francisco, California, USA, 2023. Association for Computing Machinery. doi: 10.1145/3586183.3606763.

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

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *Proceedings of the International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36*, New Orleans, Louisiana, USA, 2023. Neural Information Processing Systems Foundation. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19336–19352, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.989. URL https://aclanthology.org/2025.findings-acl.989/.

Zihao Tang, Xin Yu, Ziyu Xiao, Zengxuan Wen, Zelin Li, Jiaxi Zhou, Hualei Wang, Haohua Wang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. Mnemis: Dual-route retrieval on hierarchical graphs for long-term LLM memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 23914–23928, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1096. URL https://aclanthology.org/2026.acl-long.1096/.

Luanbo Wan and Weizhi Ma. Storybench: A dynamic benchmark for evaluating long-term memory with multi turns. arXiv preprint arXiv:2506.13356, 2025. URL https://arxiv.org/abs/2506.13356.

Xinyuan Wang, Wenyu Mao, Junkang Wu, Xiang Wang, and Xiangnan He. $R^2$-Mem: Reflective experience for memory search. arXiv preprint arXiv:2605.13486, 2026. URL https://arxiv.org/abs/2605.13486.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents. arXiv preprint arXiv:2507.07957, 2025. URL https://arxiv.org/abs/2507.07957.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. MemoryLLM: Towards self-updatable large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 50453–50466, Vienna, Austria, 2024. PMLR. URL https://proceedings.mlr.press/v235/wang24s.html.

Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending MemoryLLM with scalable long-term memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 63308–63323, Vancouver, Canada, 2025a. PMLR. URL https://proceedings.mlr.press/v267/wang25au.html.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-alpha: Learning memory construction via reinforcement learning. arXiv preprint arXiv:2509.25911, 2025b. URL https://arxiv.org/abs/2509.25911.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *Proceedings of the International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. StructMem: Structured memory for long-horizon behavior in LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 122–146, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-short.12. URL https://aclanthology.org/2026.acl-short.12/.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems 38*, pp. 20004–20031, San Diego, California, USA, 2025. Neural Information Processing Systems Foundation. doi: 10.52202/085713-0593.

Sikuan Yan, Ahmed Bahloul, Ercong Nie, Susanna Schwarzmann, Riccardo Trivisonno, Volker Tresp, and Yunpu Ma. Memory-R2: Fair credit assignment for long-horizon memory-augmented LLM agents. arXiv preprint arXiv:2605.21768, 2026a. URL https://arxiv.org/abs/2605.21768.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 12805–12825, San Diego, California, USA, 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

Jingbo Yang, Kwei-Herng Lai, Xiaowen Wang, Shiyu Chang, Yaar Harari, and Evgeniy Gabrilovich. GroupMemBench: Benchmarking LLM agent memory in multi-party conversations. arXiv preprint arXiv:2605.14498, 2026. URL https://arxiv.org/abs/2605.14498.

Jianing Yin and Tan Tang. DeferMem: Query-time evidence distillation via reinforcement learning for long-term memory QA. arXiv preprint arXiv:2605.22411, 2026. URL https://arxiv.org/abs/2605.22411.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 21457–21483, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.981. URL https://aclanthology.org/2026.acl-long.981/.

Juwei Yue, Chuanrui Hu, Jiawei Sheng, Zuyi Zhou, Wenyuan Zhang, Tingwen Liu, Li Guo, and Yafeng Deng. HyperMem: Hypergraph memory for long-term conversations. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pp. 35237–35254, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1627. URL https://aclanthology.org/2026.acl-long.1627/.

Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. In *Advances in Neural Information Processing Systems 38*, pp. 14587–14617, San Diego, California, USA, 2025. Neural Information Processing Systems Foundation. doi: 10.52202/085713-0439.

Qi Zhang, Shen Huang, Chu Liu, Shouqing Yang, Junbo Zhao, Haobo Wang, and Pengjun Xie. DeltaMem: Towards agentic memory management via reinforcement learning. arXiv preprint arXiv:2604.01560, 2026. URL https://arxiv.org/abs/2604.01560.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946.

# APPENDIX

This appendix provides the data structures and algorithms, implementation and evaluation protocols, R1 training details, reproduction checklist, complete category-level results, LoCoMo breakdown, retrieval-budget sensitivity, and all key prompts.

## A IMPLEMENTATION DETAILS

The system first deterministically stores verbatim messages and the channel roster, after which the Writer produces derived records with source provenance. System 1 selects verbatim evidence and preserves speaker attribution; System 2 retrieves structured records by per-speaker or group rows and follows provenance pointers back to the source text. Updates append new nodes and mark old nodes as superseded rather than overwriting history.

**Fixed query budgets.** All main experiments, ablations, and the controlled SFT/R1 Writer evaluation use the same query configuration: System 1 has final top-$k = 10$ with the default recall pool recall-$n = 40$, and System 2 takes $k = 2$ records per owner row with source-$k = 1$. Select ranks the recalled System 1 candidates, after which Expand may add neighboring messages; when ASK is enabled, it may add a second recalled pool before the final reranking and top-$k$ truncation. The two paths use independent budgets.

**Execution order.** Project parses the target roster, issue/event, head/full temporal mode, and relation constraints from the question; Select forms candidates for System 1 and System 2 separately. Anchor then retains source, owner, event, time, and provenance; Separate expands PERSON/GROUP rows from the roster; Resolve distinguishes issues, parallel events, and temporal versions within rows; and Compose organizes System 1 and System 2 evidence. Empty derived rows fall back to the corresponding person's verbatim messages, and explicit empty rows are retained when no evidence exists. The answerer receives only the final evidence set, while R1 training updates only the Writer. Complete prompts and reward decomposition are given in Appendices I and E.

### A.1 FIVE-LAYER MEMORY SCHEMA

The following table gives the detailed schema behind the concise description in Section 3.2. It records the implementation layer names, scope, information retained, and the role each layer plays during query-time evidence construction. System 2 records carry source/owner fields and provenance pointers (implemented as `from_ids`) to System 1 messages.

For all four System 2 layers, an update appends a new node and links it to the prior state rather than overwriting the source record; the head/full query mode selects the current state or the complete update chain (Appendix B).

## B DATA STRUCTURES AND ALGORITHMIC DETAILS

This appendix supplements the algorithms, data structures, implementation parameters, complete prompts, and examples needed to reproduce the method but omitted from the main paper. This section gives the data structures and algorithmic flow; training parameters are in Appendix E, the evaluation protocol is in Appendix C, and category-level results are in Appendix F.

### B.1 MEMORY RECORD STRUCTURE

The following table lists persistent fields shared by the verbatim and derived tracks.

### B.2 ONLINE WRITING

For an arriving message segment $X_t$, the system executes the following steps:

1. Write every message verbatim into `per_speaker_episodic`, recording the speaker, session, turn, timestamp, and message identifier; update the channel roster.

Table 5: Detailed five-layer schema of SPEAKERMEM-R1. System 1 is the verbatim track; System 2 contains four derived layers. The code names match the Writer contract and stored records.

| Track | Scope | Layer (code name) | Stored information | Query role and provenance |
|---|---|---|---|---|
| System 1 | PERSON | Episodic `per_speaker_episodic` | Verbatim message text, speaker, session/turn, timestamp, channel, and message identifier; preserves wording and local context. | Supplies exact wording and neighboring context to System 1 retrieval; messages are deterministically appended and serve as the provenance target. |
| System 2 | PERSON | Core `per_speaker_core` | Stable identity, facts, stances, and recurring behavior; summarizes a person's persistent state. | Owner-row retrieval for personal-state questions; each record keeps source/owner and points to supporting System 1 messages through `from_ids`. |
| System 2 | PERSON | Profile `per_speaker_profile` | Observations about a person and cross-person beliefs or perceptions. | Owner-row retrieval for relational questions; separate source (who provides the information) from owner (whom it concerns), with `from_ids` provenance. |
| System 2 | GROUP | Interaction `group_interaction` | Cross-speaker events, relations, and group decisions with their outcomes. | Group-row retrieval for multi-person events and decisions; GROUP scope and source/owner are retained with provenance links. |
| System 2 | GROUP | Insight `group_insight` | Group norms, consensus, and exceptions that distinguish collective from individual behavior. | Group-row retrieval for norms and shared beliefs; records remain linked to the supporting System 1 messages. |

Table 6: Persistent record fields in SPEAKERMEM-R1. The verbatim and derived tracks share this schema; the Writer generates only action fields, while the system fills the remaining coordinates from messages and the current state.

| Field | Type/value | Meaning |
|---|---|---|
| `entry_id` | string | Unique record identifier; also the target of UPDATE, chain, and source-provenance pointers. |
| `content` | string | Original message or a single derived fact, stance, observation, decision, or relation. |
| `owner`, `source` | PERSON/GROUP, speaker | `owner` is whom the record concerns; `source` is who stated or observed the information. |
| `layer` | Five discrete values | verbatim layer `per_speaker_episodic`; personal derived layers `core/profile`; group layers `interaction/insight`. |
| `utype` | Six discrete values | `utterance`, `fact`, `stance`, `observation`, `decision`, `relation`. |
| `session`, `turn` | string, integer | Channel/session and within-session turn, used for local verbatim expansion. |
| `turn_created`, `ts` | integer, timestamp | Global write order and real-time coordinate. |
| `from_ids` | list[string] | Verbatim records supporting a derived record, enabling provenance back to the source text. |
| `links`, `superseded_by` | list/string | Old-to-new chains for non-destructive UPDATE and other relations. |
| `confidence` | $[0, 1]$ | Writing confidence and retrieval-weighting signal. |

2. Read the current heads of the four derived layers and form the Writer input together with $X_t$.

3. The Writer generates at most the allowed number of ADD/UPDATE/NOOP actions using the format and operation requirements in Appendix I.
4. Validate JSON, action type, source/owner, layer, entry ID, and within-batch duplicates. For ADD, create a new record and attach supporting-message `from_ids`; for UPDATE, create a new node, inherit the original coordinates, and connect it to the old state using `links/superseded_by`. Old nodes are not deleted.
5. Write valid nodes to persistent storage and update the vector index; NOOP leaves the derived track unchanged.

This process separates "what to write" from "coordinates completed by the system": the Writer decides what to record, while deterministic code fills message coordinates, provenance, and update chains.

**Action state transitions.** The Writer input consists of the current message segment $X_t$, the roster, and the current heads of the four derived layers; an UPDATE action must additionally carry the `entry_id` of the node to update. The system validates that this ID resolves to an existing derived record and reuses its owner, source, and layer coordinates, then writes the new content and time and connects old and new states through `links/superseded_by`. If the ID is invalid, Update is rejected and treated as Add. The old node remains stored; `head` versus `full` controls whether querying returns only the current node or the complete chain. Noop creates no derived node.

### B.3 DUAL-TRACK COMPOSITION AT QUERY TIME

Given a question $q$ and channel roster, the query proceeds as follows:

1. **Project:** output the issue, PERSON/GROUP rows, and `head/full`.
2. **System 1:** retrieve a default pool of $n = 40$ candidates from the verbatim track; Select ranks the pool for necessity, and Expand fetches neighboring source messages along selected hits' session/turn. If final evidence remains insufficient and ASK is enabled, Sufficiency generates one query term, another candidate pool is recalled and merged, and the combined pool is reranked. Only after these operations is the final set truncated to top-$k$.
3. **System 2:** expand rows into concrete roster rows. Retrieve derived records by issue within each row and collapse or expand update chains according to `head/full`; source supplements do not consume the owner-row budget. Empty derived rows can fall back to the row's verbatim messages, and explicit empty rows are retained when evidence is unavailable.
4. **Compose:** preserve the owner, source, time, and provenance of every record, and pass System 1 verbatim evidence alongside System 2 structured evidence to the frozen answerer.

The two retrieval paths use independent budgets, and System 2 does not participate in System 1 pre-pruning. Structured memory therefore determines which objects, scopes, and states must be checked, while verbatim memory supplies exact wording and details.

## C BENCHMARKS AND EVALUATION PROTOCOL

The table summarizes the number of questions, metrics and location of the category-level results reported in this paper.

All main SPEAKERMEM-R1 experiments and ablations use the default setting: System 1 top-$k = 10$ and System 2 $k = 2$/source-$k = 1$. Full context is included only as a reference upper bound on SocialMem. Both DeepSeek-V4-Flash and GPT-5.6-luna rows use GPT-4o-mini for question-level judging; the EverMemBench cross-configuration results and judge settings are given in Table 2. token-F1 is computed directly from the answer and gold answer and does not depend on a judge; the two result types are reported separately. For the primary three-benchmark comparison, each system follows its official code, recommended configuration, and official prompt; we standardize only the data, metric definitions, GPT-4o-mini judge, and evaluation interface, while speaker/source metadata may be retained where supported. Detailed procedures and result tables are provided in the supplementary material.

Table 7: Evaluation scope, number of questions, primary metric, and reporting location.

| Benchmark | Questions | Metrics | Reporting scope |
|---|---|---|---|
| GroupMemBench | 745 | Question-level binary accuracy | Six-category accuracy and token-F1; the GPT-5.6-luna setting is reported separately |
| SocialMemBench | 1,031 | Question-level binary accuracy | Q1–Q9; MeanQ/MeanN and Q1–Q9 category results are also reported |
| EverMemBench | 2,400 | Question-level binary accuracy | EverMind-AI nine-behavior accuracy and weighted overall result |
| LoCoMo | 1,986 | Question-level binary accuracy | Five-category accuracy and the published comparison protocol excluding AD |

**Acc. and token-F1.** Let $Q$ be the number of evaluated questions and $j_i \in \{0, 1\}$ the judgment for question $i$: $j_i = 1$ only when the answer is judged correct, and 0 otherwise. Question-level binary accuracy is

$$\text{Acc.} = \frac{100}{Q} \sum_{i=1}^{Q} j_i. \tag{10}$$

token-F1 is computed per question and then averaged with equal question weights. Following the supplementary implementation, predictions and references are lowercased, stripped of surrounding whitespace and trailing periods, normalized for repeated whitespace, and split on whitespace; duplicate tokens are removed to form sets $A_i$ and $G_i$, respectively. Define

$$p_i = \frac{|A_i \cap G_i|}{|A_i|}, \quad r_i = \frac{|A_i \cap G_i|}{|G_i|}, \quad f_i = \frac{2p_i r_i}{p_i + r_i}, \quad \text{token-F1} = \frac{100}{Q} \sum_{i=1}^{Q} f_i. \tag{11}$$

If either set is empty or their intersection is empty, we directly set $f_i = 0$ to avoid division by zero. This implementation uses token sets rather than multisets retaining repeated occurrences; it measures lexical overlap rather than semantic correctness. Both Acc. and token-F1 are reported as percentages; category results use the same definitions averaged over questions in that category.

**MeanQ and MeanN.** SocialMemBench first assigns each question a score $s_q \in [0, 1]$: the 214 multiple-choice questions receive 0/1 for exact option matching, while the 817 open-ended questions receive partial credit under the benchmark's itemized scoring protocol; per-person stance questions are scored by the fraction of members whose stances are recovered correctly (Owolabi, 2026). Let $\mathcal{Q}_n$ denote the question set for network $n$, with $N = 43$ networks. Then

$$\text{MeanQ} = \frac{1}{\sum_n |\mathcal{Q}_n|} \sum_{n=1}^{N} \sum_{q \in \mathcal{Q}_n} s_q, \qquad \text{MeanN} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{|\mathcal{Q}_n|} \sum_{q \in \mathcal{Q}_n} s_q. \tag{12}$$

MeanQ gives every question equal weight, so networks with more questions contribute more; MeanN first computes a mean for each network and then weights all networks equally, reducing the effect of unequal network sizes. Both are means of item-level partial scores, not the proportion of questions answered completely correctly, and neither is interchangeable with question-level binary accuracy. Our 95% CI applies only to MeanN: the 43 network means are the bootstrap units, and we use 10,000 resamples and take the 2.5th and 97.5th percentiles of the resulting distribution.

## D Reproduction Checklist and Experimental Boundaries

This section summarizes the experimental boundaries using common ICLR reproducibility checks. Detailed algorithms, hyperparameters, and prompts are given in Appendices B, E, and I. The table distinguishes the language models used across system stages from question-level judging settings.

**Primary evaluation protocol.** For the three multi-party benchmark comparisons, each system follows its official code, recommended configuration, and official prompt. We standardize only the data, metric definitions, judge, and evaluation interface; implementations may preserve speaker/source metadata where supported. Detailed execution procedures and result tables are provided in the supplementary material.

Table 8: Reproduction checklist for SPEAKERMEM-R1.

| Item | Fixed setting | Reproduction location and notes |
|---|---|---|
| Data and splits | GroupMemBench 745; SocialMemBench 1,031; EverMemBench 2,400; LoCoMo 1,986 | Benchmark names, question counts, categories, and comparison conventions are given in Appendix C; R1 training and held-out networks are separated at the network level. |
| Model roles | DeepSeek-V4-Flash or GPT-5.6-luna is used across memory construction, retrieval, and answer generation | The main results in Table 1 and the complete category tables identify the model configuration; both DeepSeek-V4-Flash and GPT-5.6-luna rows use GPT-4o-mini for question-level judging; the EverMemBench cross-configuration results identify the judge in Table 2. |
| Memory writing | System 1 deterministically appends verbatim messages; System 2 uses the Qwen2.5-3B Writer to generate four types of derived records | The five-layer schema, action-format requirements, and non-destructive update chains are given in Appendix B. |
| Query budgets | System 1: recall-$n = 40$, final top-$k = 10$; Expand at most twice; ASK at most once with 20 retrieved items; System 2: $k = 2$/source-$k = 1$ | Complete input/output specifications for Project, Select/Expand, Sufficiency/ASK, and within-row selection are given in Appendix I. |
| R1 training | $G = 8$ trajectories; 30 rollout rounds; 2 epochs per update; learning rate $10^{-6}$; temperature .8; top-$p = .95$ | SpeakerLevenshtein (speaker-conditioned matching), speaker-conditioned GRPO, reward weights, KL, and data scale are given in Appendix E. |
| Evaluation convention | We report question-level binary accuracy and token-F1; SocialMem additionally reports MeanQ/MeanN | Both DeepSeek-V4-Flash and GPT-5.6-luna rows use GPT-4o-mini for judging; EverMemBench cross-configuration results and judge settings are given in Table 2; token-F1 is computed directly from answers and gold answers without a judge. The two result types are reported separately. |

**Implementation boundaries.** Main benchmark results and the controlled SFT/R1 Writer evaluation use the same query configuration: System 1 top-$k = 10$ with the default recall-$n = 40$, System 2 $k = 2$ per owner plus source-$k = 1$. SocialMemBench MeanN uses 10,000 bootstrap resamples over 43 networks and reports a 95% confidence interval. Because the R1 training set contains model-generated teacher trajectories and manually authored boundary networks, the appendix reports their sources, action statistics, and train/evaluation isolation. Datasets are used under the licenses and access terms of their original publishers; we report only experimental settings, data composition, and results.

**Token accounting for final result artifacts.** We count tokens from the prompts, messages, memories, and answers processed at each stage. Total tokens include ingestion input, ingestion output, QA input, and final answer output, and exclude the judge stage; average tokens per question use the corresponding benchmark size as denominator. The three benchmarks contain 4,176 questions: 1,031 SocialMem, 745 GroupMem, and 2,400 EverMem. The SpeakerMem Writer can emit multiple update actions in one call, enabling small-batch ingestion; we therefore use the deployed configuration, with session-level ingestion for SocialMem and Window25 for GroupMem/EverMem. For Mem0, A-MEM, and HippoRAG, larger input chunks can reduce memory quality, so we use chunk5: each ingestion call processes five messages before the system performs memory merging, conflict resolution, and subsequent steps. This retains limited batching while avoiding the very high token cost of fully serial ingestion, and provides the cost-comparison setting reported here.

**Three-benchmark overview.** This table aggregates all processing stages. SpeakerMem uses session-level ingestion for SocialMem and Window25 for GroupMem/EverMem; the other memory systems ingest messages in chunk5 units.

**SocialMemBench breakdown.** SocialMemBench contains 1,031 questions. SpeakerMem ingests 348 sessions; the other systems ingest 1,471 message units under chunk5.

**GroupMemBench breakdown.** GroupMemBench contains 745 questions and 120,000 messages. SpeakerMem makes 4,815 Window25 Writer calls; the other systems ingest 24,000 chunk5 units.

**EverMemBench breakdown.** EverMemBench contains 2,400 questions and 51,023 messages. SpeakerMem makes 2,049 Window25 Writer calls; the other systems ingest 10,205 chunk5 units.

Table 9: Token totals across the final result artifacts.

| System | Calls/units | Ingest in. | Ingest out. | QA input | Answer out. | Total | Avg./question |
|---|---|---|---|---|---|---|---|
| SpeakerMem-R1+ASK | 7,212 | 25,546,058 | 721,200 | 13,865,187 | 430,530 | 40,562,975 | 9,713.36 |
| BM25 | 0 | 0 | 0 | 5,298,971 | 85,011 | 5,383,982 | 1,289.27 |
| Embed | 0 (local) | 0 | 0 | 5,096,215 | 96,764 | 5,192,979 | 1,243.53 |
| Mem0-chunk5 | 35,676 | 29,970,000 | 29,970,000 | 2,316,690 | 91,476 | 62,348,166 | 14,930.12 |
| A-MEM-chunk5 | 35,676 | 39,929,508 | 20,100,569 | 12,482,895 | 159,830 | 72,672,802 | 17,402.49 |
| HippoRAG-chunk5 | 35,676 | 30,400,001 | 30,400,001 | 11,506,209 | 155,857 | 72,462,068 | 17,352.03 |

Table 10: SocialMemBench (1,031 questions)

| System | Calls/units | Ingest in. | Ingest out. | QA input | Answer out. | Total | Avg./question |
|---|---|---|---|---|---|---|---|
| SpeakerMem-R1+ASK | 348 | 845,605 | 34,800 | 846,404 | 75,247 | 1,802,056 | 1747.87 |
| BM25 | 0 | 0 | 0 | 1,308,247 | 20,988 | 1,329,235 | 1289.27 |
| Embed | 0 | 0 | 0 | 1,258,189 | 23,890 | 1,282,079 | 1243.53 |
| Mem0-chunk5 | 1,471 | 1,236,000 | 1,236,000 | 306,976 | 31,561 | 2,810,537 | 2726.03 |
| A-MEM-chunk5 | 1,471 | 1,646,400 | 828,800 | 1,328,327 | 73,915 | 3,877,442 | 3760.86 |
| HippoRAG-chunk5 | 1,471 | 1,253,473 | 1,253,473 | 1,032,295 | 71,444 | 3,610,685 | 3502.12 |

Table 11: GroupMemBench (745 questions)

| System | Calls/units | Ingest in. | Ingest out. | QA input | Answer out. | Total | Avg./question |
|---|---|---|---|---|---|---|---|
| SpeakerMem-R1+ASK | 4,815 | 17,445,456 | 481,500 | 2,152,506 | 29,344 | 20,108,806 | 26991.69 |
| BM25 | 0 | 0 | 0 | 945338 | 15166 | 960504 | 1289.27 |
| Embed | 0 | 0 | 0 | 909167 | 17263 | 926430 | 1243.53 |
| Mem0-chunk5 | 24,000 | 20,160,000 | 20,160,000 | 253,800 | 20,582 | 40,594,382 | 54489.10 |
| A-MEM-chunk5 | 24,000 | 26,861,726 | 13,522,230 | 2,444,228 | 33,277 | 42,861,461 | 57532.16 |
| HippoRAG-chunk5 | 24,000 | 20,450,953 | 20,450,953 | 2,448,714 | 32,569 | 43,383,189 | 58232.47 |

Table 12: EverMemBench (2,400 questions)

| System | Calls/units | Ingest in. | Ingest out. | QA input | Answer out. | Total | Avg./question |
|---|---|---|---|---|---|---|---|
| SpeakerMem-R1+ASK | 2,049 | 7,254,997 | 204,900 | 10,866,277 | 325,939 | 18,652,113 | 7771.71 |
| BM25 | 0 | 0 | 0 | 3045386 | 48857 | 3094243 | 1289.27 |
| Embed | 0 | 0 | 0 | 2928859 | 55611 | 2984470 | 1243.53 |
| Mem0-chunk5 | 10,205 | 8,574,000 | 8,574,000 | 1,755,914 | 39,333 | 18,943,247 | 7893.02 |
| A-MEM-chunk5 | 10,205 | 11,421,382 | 5,749,539 | 8,710,340 | 52,638 | 25,933,899 | 10805.79 |
| HippoRAG-chunk5 | 10,205 | 8,695,575 | 8,695,575 | 8,025,200 | 51,844 | 25,468,194 | 10611.75 |

# E ADDITIONAL R1 TRAINING DETAILS

**Record matching with SpeakerLevenshtein (speaker-conditioned matching).** Let the predicted and reference derived states be $M$ and $M^\star$, partitioned into buckets $M_p$ and $M_p^\star$ by owner, including a separate `GROUP` owner. The soft content match and coordinate gate between predicted record $i$ and reference record $j$ are

$$\bar{\kappa}_{ij} = \alpha_{\text{tok}} F_1^{\text{tok}}(c_i, c_j) + \alpha_{\text{seq}} S_{\text{seq}}(c_i, c_j), \tag{13}$$

$$\kappa_{ij} = \mathbf{1}[o_i = o_j,\ s_i = s_j,\ \ell_i = \ell_j]\mathbf{1}[\bar{\kappa}_{ij} \geq 0.50]\bar{\kappa}_{ij}, \tag{14}$$

We use $\alpha_{\text{tok}} = 0.60$ and $\alpha_{\text{seq}} = 0.40$. For normalized token sequences $T_i$ and $T_j$, let $m_{ij}$ be the total number of matched tokens returned by `SequenceMatcher`; its ratio is

$$S_{\text{seq}}(c_i, c_j) = \frac{2m_{ij}}{|T_i| + |T_j|}, \qquad T_i = T(c_i),\ T_j = T(c_j). \tag{15}$$

where $c, o, s, \ell$ denote content, owner, source, and memory layer, respectively; $S_{\text{seq}}$ is the ratio returned by Python `SequenceMatcher.ratio()` on normalized token sequences: content is lowercased, `[a-z0-9]+` spans are tokenized, and each CJK character is a token; the tokens are joined with spaces before computing the ratio. The name follows the experimental implementation, but the score itself does not use standard Levenshtein edit distance. Semantically similar records with inconsistent coordinates receive no match score. Within each owner bucket, we perform maximum-weight one-to-one Hungarian alignment:

$$\mathcal{A}_p^\star = \arg\max_{\mathcal{A} \in \mathfrak{M}(M_p, M_p^\star)} \sum_{(i,j)\in\mathcal{A}} \kappa_{ij}, \qquad F_p = \frac{2\sum_{(i,j)\in\mathcal{A}_p^\star} \kappa_{ij}}{|M_p| + |M_p^\star|}, \tag{16}$$

where $\mathfrak{M}$ is the set of one-to-one matchings. A prediction cannot match multiple reference facts, so $F_p$ penalizes both omissions and over-writing.

**Update-chain reward.** Correct state nodes do not imply correct update relations. For predicted and reference UPDATE edges, we first require matching owner and layer coordinates, then compare the old and new states separately:

$$\kappa_{ij}^{\text{chain}} = w_{\text{old}} \bar{\kappa}(c_i^{\text{old}}, c_j^{\text{old}}) + w_{\text{new}} \bar{\kappa}(c_i^{\text{new}}, c_j^{\text{new}}). \tag{17}$$

We use $w_{\text{old}} = 0.50$ and $w_{\text{new}} = 0.50$. Applying the same one-to-one soft-F1 alignment to chain edges yields $R_{g,t}^{\text{chain}}$, and the memory reward is

$$R_{g,t}^{\text{mem}} = w_{\text{SL}} \Phi_{\text{SL}}(M_{g,t}, M_t^\star) + w_{\text{chain}} R_{g,t}^{\text{chain}}. \tag{18}$$

We use $w_{\text{SL}} = 0.80$ and $w_{\text{chain}} = 0.20$.

**Action validity and structural penalties.** For action $t$ on candidate trajectory $g$,

$$R_{g,t}^{\text{valid}} = w_{\text{json}} R_{g,t}^{\text{json}} + w_{\text{schema}} R_{g,t}^{\text{schema}} + w_{\text{dup}} R_{g,t}^{\text{dup}} + w_{\text{stop}} R_{g,t}^{\text{stop}}. \tag{19}$$

We use $w_{\text{json}} = 0.25$, $w_{\text{schema}} = 0.35$, $w_{\text{dup}} = 0.25$, and $w_{\text{stop}} = 0.15$. Here a candidate action list is denoted by a and n is its attempted-action count (from the execution report, or from the list length when omitted). The four components $R_{g,t}^{\text{json}}$, $R_{g,t}^{\text{schema}}$, $R_{g,t}^{\text{dup}}$, and $R_{g,t}^{\text{stop}}$ are indexed by the trajectory and writing position (g,t). The JSON component is one iff the report marks the output as valid JSON. The schema component is the clipped fraction of schema-valid actions among attempts; if the count is absent, it is inferred from attempts minus validation errors, and is zero for invalid JSON. The duplicate component is the clipped fraction of unique action keys among attempts. ADD keys contain action, owner, source, layer, and normalized content; UPDATE keys contain action, entry ID, and normalized content; a NOOP key is its action type. If the attempted-action count is zero, all four components and the final validity score are zero.

The stop component is one iff the output stops immediately after its last novel action, and is zero otherwise. A single NOOP is novel by definition. An ADD is novel when its key is neither repeated in the output nor already present in the pre-state. An UPDATE is novel only when its entry ID resolves

to an active pre-state record and its normalized content differs from that record. The Writer action budget is

$$n_{g,t} \le \max_\text{actions}(X_t) = \max(8, 2|X_t|), \tag{20}$$

where $|X_t|$ is the number of messages in the input segment; exceeding this bound is reported as `too_many_actions` and contributes the count term in the structural penalty below. Let $z^{\text{json}}, z^{\text{act}}, z^{\text{count}}, z^{\text{len}} \in \{0, 1\}$ indicate invalid JSON, invalid actions, too many actions, and output reaching the length limit, respectively. Then

$$P_{g,t} = -\eta_{\text{json}} z^{\text{json}}_{g,t} - \eta_{\text{act}} z^{\text{act}}_{g,t} - \eta_{\text{count}} z^{\text{count}}_{g,t} - \eta_{\text{len}} z^{\text{len}}_{g,t}. \tag{21}$$

We use $\eta_{\text{json}} = 0.25$, $\eta_{\text{act}} = 0.15$, $\eta_{\text{count}} = 0.10$, and $\eta_{\text{len}} = 0.10$. Together these terms provide bounded dense structural signals for each transition.

**Terminal QA gain.** For each training network, we sort by question ID and uniformly sample at most four questions to form a fixed subset $\mathcal{Q}$, with binary judge $J(\hat{a}, a^\star)$. Define

$$\text{QA}(E; M) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} J\big(\text{Ans}(q, E; M), a^\star_q\big)\,. \tag{22}$$

System 1+System 2 and System 1-only share the same System 1 evidence, so Eq. 8 isolates the gain from the derived track. This terminal difference is propagated to every action on the trajectory with $\gamma^{T-1-t}$ and combined with local terms in Eq. 9.

**Transition-aligned group-relative advantages.** We sample $G = 8$ independent on-policy trajectories for the same network. The within-group mean and standard deviation at the same position $t$ are

$$\mu_t = \frac{1}{G} \sum_{g=1}^{G} r_{g,t}, \qquad \sigma_t = \sqrt{\frac{1}{G} \sum_{g=1}^{G} (r_{g,t} - \mu_t)^2}. \tag{23}$$

The advantage is defined as

$$\hat{A}_{g,t} = \begin{cases} 0, & \sigma_t < 0.02, \\ \text{clip}\big((r_{g,t} - \mu_t)/\sigma_t, -3, 3\big)\,, & \text{otherwise.} \end{cases} \tag{24}$$

Position-wise comparison prevents semantically different actions from trajectories of different lengths from sharing one normalization group. Zero-variance groups provide no effective relative signal and are therefore excluded from updates.

**Clipped GRPO and KL.** For completion token $y_{g,t,k}$, the importance ratio is

$$\rho_{g,t,k}(\theta) = \exp\big(\log \pi_\theta(y_{g,t,k} \mid x_{g,t}, y_{g,t,<k}) - \log \pi_{\text{old}}(y_{g,t,k} \mid x_{g,t}, y_{g,t,<k})\big)\,. \tag{25}$$

Equation 9 first gives the return for each transition, after which group-relative advantages are computed at the same writing position. For completion token $y_{g,t,k}$, the clipped GRPO objective is

$$\mathcal{J}(\theta) = \mathbb{E}_{g,t,k}\Big[\min\Big(\rho_{g,t,k}\hat{A}_{g,t}, \text{clip}(\rho_{g,t,k}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{g,t}\Big) - \beta d^{\text{KL}}_{g,t,k}\Big]\,, \tag{26}$$

We first average over tokens within each completion and then over samples with nonzero advantages; $\varepsilon = 0.20$, $\beta = 0.10$, and we use

$$d^{\text{KL}}_{g,t,k} = \exp(\delta_{g,t,k}) - \delta_{g,t,k} - 1, \qquad \delta_{g,t,k} = \log \pi_{\text{ref}} - \log \pi_\theta, \tag{27}$$

$\pi_{\text{ref}}$ is the frozen SFT policy. If the mean KL exceeds 0.02, the update is skipped.

**Training data and hyperparameters.** The Writer training set contains 15 multi-party networks and 73 writing segments (a writing segment is a Writer input unit, not the number of actions output by the model). To prevent sessions or windows from the same network appearing in both training and evaluation, train and held-out sets are separated at the complete-network level. We select the 10 shortest real SocialMemBench networks by message count, containing 50–101 messages, five sessions per network, and 5–12 terminal QA questions; they do not overlap with the 10 held-out SocialMemBench evaluation networks. We additionally use three manually authored UPDATE

networks (30, 35, and 40 messages, covering state progression, UPDATE chains, source/owner separation, PERSON/GROUP scope, and temporal changes) and two manually authored NOOP networks (2–3 greeting or confirmation messages with no persistent fact).

Table 13 gives a reproducible inventory of the training networks. Although the training inventory contains only 15 complete networks, each network is processed through multiple Writer segments, yielding 73 writing segments, 89 terminal QA items, and 452 supervised actions (430 ADD, 20 UPDATE, and 2 NOOP). The effective training signal therefore comes from repeated Writer decisions within each network rather than from 15 isolated samples. Writer trajectories for real networks are generated by DeepSeek-V4-Flash under the fixed Writer contract; manual networks are constructed directly by the experiment script with the same action and state contract and are not synthesized by a model. Each entry is isolated at the complete-network level, so messages, transitions, and terminal QA questions from one network never cross train and held-out sets.

The 10 real networks produce 420 ADD and 3 UPDATE actions; the five manual networks produce 10 ADD, 17 UPDATE, and 2 NOOP actions, for 430 ADD, 20 UPDATE, and 2 NOOP actions in total. These action counts span all message positions and should not be added to or exchanged with the 73 writing segments in the table. Both RL and SFT Writers use the locally deployable Qwen2.5-3B. Real networks use DeepSeek-V4-Flash under the current Writer contract to generate teacher trajectories; these trajectories are proxy supervision rather than manually annotated item-level gold labels. Each network is sampled twice with group size $G = 8$; at most four terminal QA questions are fixed for each training network. After SFT, we start from the epoch-10 checkpoint and run 30 rollout rounds. Each policy update uses two PPO epochs, learning rate $10^{-6}$, gradient-norm cap 1.0, temperature 0.80, top-$p = 0.95$, clip 0.20, KL coefficient 0.10, and maximum generation length 2048. Terminal QA during training is used only for reward computation; the main benchmark evaluation still uses the fixed answerer and scoring protocol specified in the paper.

**Controlled results.** Under the same no-ASK query configuration as the main experiments (System 1 recall-$n = 40$ with final top-$k = 10$, System 2 $k = 2$/source-$k = 1$) and with the query and answer modules frozen, the correct counts for the three random seeds are 174/305, 175/305, and 176/305 for SFT (epoch 10), and 206/305, 208/305, and 210/305 for the RL Writer (30 steps). The corresponding mean accuracies are $57.38 \pm 0.33\%$ and $68.20 \pm 0.66\%$ (sample standard deviation), an average gain of 10.82 percentage points, or 33 additional correct answers. Under the same protocol, the three LLM writer runs with a DeepSeek-V4-Flash Writer obtain 216/305, 218/305, and 220/305, for a mean accuracy of $71.48 \pm 0.66\%$; the RL Writer is 3.28 points behind and reaches 95.4% of the LLM writer accuracy. This comparison shows that the small Writer approaches the large-model reference, without being interpreted as evidence of cross-domain RL generalization.

# F Complete Category Results on the Three Multi-Party Benchmarks

This section gives the category results underlying the main summary tables; the main experiments contain 745, 1,031, and 2,400 questions, respectively. In all tables, $^{\dagger}$ denotes ASK disabled.

## F.1 Five Analysis Dimensions: Definitions and Typical Errors

Here, the five dimensions form an analysis framework derived from question requirements and recurring error patterns; they are not independently human-annotated diagnostic labels. They organize related failure modes across the three benchmarks without defining a new validated taxonomy.

The five dimensions are not mutually exclusive question labels and do not alter the original annotations of the benchmarks. We retain each benchmark's official categories as quantitative reporting units, then map the dominant failure modes in each question to one or more diagnostic dimensions. The dimensions therefore explain why a question is missed rather than define a new aggregate score. The mapping uses the requirements in the question, object/scope/time constraints in the reference answer, and typical error cases, without additional manual gold labels beyond the test-set answers.

**Full-member coverage.** Questions asking for "every person", "all participants", or a group member list require evidence for every roster member or an explicit statement that information is missing. A

Table 13: Composition of the R1 Writer training data. Real networks come from the SocialMemBench training side, while manual networks cover UPDATE/NOOP boundaries. A "writing segment" is a Writer input unit partitioned by session, not a count of ADD/UPDATE/NOOP actions; the QA column gives the available terminal questions for each network.

| Source | network ID | Messages | Writing segments | QA |
|---|---|---|---|---|
| Real | grp_0d1e2f3a | 50 | 5 | 6 |
| Real | grp_5e6f7a8b | 50 | 5 | 8 |
| Real | grp_6f7a8b9c | 50 | 5 | 7 |
| Real | grp_7a8b9c0d | 50 | 5 | 5 |
| Real | grp_8b9c0d1e | 50 | 5 | 6 |
| Real | grp_9c0d1e2f | 50 | 5 | 6 |
| Real | grp_4d5e6f7a | 53 | 5 | 9 |
| Real | grp_3c4d5e6f | 61 | 5 | 8 |
| Real | grp_2b3c4d5e | 67 | 5 | 10 |
| Real | grp_1a2b3c4d | 101 | 5 | 12 |
| Manual UPDATE | manual_update_alpha | 30 | 6 | 4 |
| Manual UPDATE | manual_update_beta | 35 | 7 | 3 |
| Manual UPDATE | manual_update_gamma | 40 | 8 | 3 |
| Manual NOOP | manual_noop_greeting | 2 | 1 | 1 |
| Manual NOOP | manual_noop_ack | 3 | 1 | 1 |
| **Total** | 15 networks | – | 73 | 89 |

typical error is that global top-$k$ retrieval returns only frequent members and misses infrequent or late-arriving members.

**Information attribution.** Questions require distinguishing the message source (who provides or expresses it) from the owner (whom the content concerns), while preserving the direction of reports, evaluations, and stances. A typical error is rewriting Alice's judgment about Bob as Bob's self-report.

**Personal/group scope.** Questions require distinguishing an individual preference or exception, a two-person relation, a multi-person event, and a group norm or consensus. Typical errors elevate a minority opinion to a group conclusion or use a group rule to overwrite an individual exception.

**Term and event disambiguation.** Questions require locating the correct branch among same-named terms, similar topics, or parallel events using participants, channels, time, and verbatim clues. A typical error is merging different projects, meetings, or decisions into one event.

**Temporal updates and multi-hop reasoning.** Questions require selecting the requested point along a state-change chain and composing multiple pieces of evidence across members and events. Typical errors use an obsolete state as the current one or return only one fact from a chain without the connections needed for intermediate reasoning.

Table 14 maps the five problem dimensions to SPEAKERMEM-R1 mechanisms; the main method section describes their implementation during writing and querying.

### F.2 CATEGORY DEFINITIONS FOR THE THREE BENCHMARKS

**GroupMemBench categories.** GroupMemBench contains 745 questions in six categories: abstention (recognizing insufficient information and avoiding speculation), user-implicit (inferring behavior or preferences not directly stated by the user), multi-hop (composing answers across multiple memories), temporal (locating or comparing states by time), knowledge-update (handling knowledge updates and obsolete information), and term-ambiguity (disambiguating terms or expressions) (Yang et al., 2026).

**SocialMemBench categories.** SocialMemBench contains 1,031 questions: Q1 single-person recall and behavior patterns; Q2 group decisions and shared expectations; Q3 per-person stances; Q4 speaker attribution; Q5 cross-person beliefs; Q6 group norms and individual exceptions; Q7 two-person relations and shared history; Q8 temporal evolution; and Q9 outlier and departed-member information (Owolabi, 2026).

**EverMemBench categories.** The main EverMemBench analysis uses the nine memory behaviors published by EverMind-AI:*Single*, *Multi*, *Temp*, *Const*, *Proact*, *Update*, *Style*, *Skill*, and

Table 14: Diagnostic dimensions and mechanisms in SPEAKERMEM-R1.

| Challenge | Common gaps in summaries, event hierarchies, and graph memory | Corresponding mechanism in SPEAKERMEM-R1 |
|---|---|---|
| Full-member coverage | Global top-$k$ and high-frequency nodes can retrieve much relevant evidence but do not guarantee one row per person | Deterministic roster, row-first System 2 retrieval, and explicit empty rows |
| Information attribution | Untyped facts or entity edges can merge "who said it" with "whom it concerns" | The profile layer and separate source/owner fields |
| Personal/group scope | Group summaries can elevate minority opinions to consensus, while person nodes do not express group norms well | PERSON core/profile and GROUP interaction/insight |
| Term and event disambiguation | Similar topics or graph connectivity can incorrectly merge same-named issues and parallel events | Fix person/GROUP rows first, then disambiguate within rows using events, relations, and provenance |
| Temporal updates and multi-hop reasoning | Overwrite updates lose history, while retaining every node may return stale states | Non-destructive update chains, head/full modes, and query-time evidence composition |

Table 15: Mapping between the five diagnostic dimensions and official categories of the three benchmarks. One official category may involve multiple dimensions, so this is not a mutually exclusive labeling scheme.

| Diagnostic dimension | Typical official categories/question types | Primary diagnostic cues |
|---|---|---|
| Full-member coverage | SocialMem Q3/Q9; GroupMem per-member, outlier-member, and full-enumeration questions; EverMem multi-role coverage questions | Whether every target roster member is covered and infrequent, late-arriving, or departed members are handled correctly |
| Information attribution | SocialMem Q3/Q4/Q5; GroupMem user-implicit and relation questions; EverMem Role/Style questions | Whether source, owner, described person, audience, and observer remain consistent |
| Personal/group scope | SocialMem Q2/Q6/Q7; GroupMem group-dynamics questions; EverMem Const/Proact questions | Whether personal preferences, two-person relations, multi-person events, and group norms are distinguished |
| Term and event disambiguation | GroupMem term-ambiguity; SocialMem similar-issue/parallel-event questions; EverMem Multi/Role questions | Whether participant, channel, time, and event clues point to the same branch |
| Temporal updates and multi-hop reasoning | GroupMem temporal/knowledge-update/multi-hop; SocialMem Q7/Q8; EverMem Temp/Update/Multi | Whether the requested time is selected and evidence chains across members or events are connected |

*Role* (Hu et al., 2026b; EverMind researchers, 2026). Main-text Table 2 reports the nine category accuracies and the question-count-weighted overall accuracy over 2,400 questions; multiple-choice/open-ended format is no longer used as the primary classification.

### F.3 BINARY ACCURACY WITH DEEPSEEK-V4-FLASH AS THE ANSWER MODEL

The following tables report the DeepSeek-V4-Flash answerer configuration, while question-level binary results are judged uniformly by GPT-4o-mini; DeepSeek-V4-Flash is not the judge. Tables 16 and 17 give the category results for GroupMemBench and SocialMemBench, respectively. The main nine-behavior EverMemBench analysis is in main-text Table 2.

**GroupMemBench configuration.** GroupMemBench has 745 questions in the six categories defined above. DeepSeek-V4-Flash generates answers and GPT-4o-mini provides the question-level binary judge. Each system follows its official code, recommended configuration, and official prompt; we standardize only the data, metric definition, judge, and evaluation interface, with speaker/source fields retained where supported.

**SocialMemBench configuration.** SocialMemBench has 1,031 questions partitioned into Q1–Q9. DeepSeek-V4-Flash generates answers and GPT-4o-mini supplies every binary verdict under the standardized data, metric, judge, and evaluation-interface protocol. Full context is feasible only for SocialMemBench; the other benchmark contexts are too long for full-context input.

Table 16: GroupMemBench category accuracy (%).

| Method | Abst. | Implicit | Multi-hop | Temporal | K.-update | Term-amb. | Total |
|---|---|---|---|---|---|---|---|
| BM25 | 89.9 | 46.9 | 41.8 | 41.4 | 23.4 | 15.1 | 44.6 |
| Embed | 87.8 | 40.8 | 25.8 | 17.3 | 20.6 | 17.0 | 34.5 |
| Mem0 | 88.5 | 10.2 | 7.1 | 2.5 | 5.6 | 9.4 | 21.6 |
| A-MEM | 80.6 | 22.4 | 15.9 | 5.6 | 13.1 | 25.5 | 27.1 |
| HippoRAG | 77.7 | 22.4 | 18.7 | 5.6 | 11.2 | 25.5 | 27.0 |
| SPEAKERMEM-R1$^{\dagger}$ | 71.2 | 44.9 | 40.7 | 49.4 | 33.6 | 36.8 | **47.0** |
| SPEAKERMEM-R1 | 73.4 | 51.0 | 43.4 | 50.6 | 33.6 | 31.1 | **47.9** |

Table 17: SocialMemBench Q1–Q9 category accuracy (%).

| Method | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| BM25 | 25.3 | 24.7 | 4.5 | 66.2 | 38.0 | 25.9 | 29.8 | 20.6 | 10.0 | 28.6 |
| Embed | 39.4 | 28.4 | 13.6 | 62.5 | 53.3 | 46.3 | 40.9 | 25.6 | 40.0 | 38.1 |
| Mem0 | 20.9 | 14.8 | 4.5 | 11.3 | 3.3 | 25.9 | 9.4 | 12.2 | 10.0 | 13.7 |
| A-MEM | 60.6 | 76.5 | 9.1 | 80.0 | 65.2 | 51.8 | 53.0 | 45.0 | 50.0 | 56.8 |
| HippoRAG | 60.6 | 71.6 | 4.5 | 68.8 | 63.0 | 48.1 | 53.0 | 47.7 | 60.0 | 55.9 |
| Full context | 74.3 | 51.9 | 18.2 | 73.8 | 77.2 | 53.7 | 77.3 | 68.3 | 70.0 | 69.4 |
| SPEAKERMEM-R1$^{\dagger}$ | 73.9 | 69.1 | 9.1 | 90.0 | 78.3 | 46.3 | 76.2 | 59.9 | 70.0 | **69.2** |
| SPEAKERMEM-R1 | 69.1 | 70.4 | 4.5 | 80.0 | 79.3 | 53.7 | 65.7 | 56.1 | 70.0 | 64.9 |

### F.4 CATEGORY-LEVEL TOKEN-F1

token-F1 is not equivalent to question-level binary accuracy. The results below are category means of question-level token-F1; SPEAKERMEM-R1 F1 is computed from outputs using the same token-matching rule as the main experiments. Tables 18 and 19 give category-level token-F1 for the two multi-party benchmarks; EverMemBench token-F1 is reported only as an overall supplementary metric in the main table.

**GroupMemBench token-F1.** The following table gives token-F1 on the same six categories. The rows follow their official code, recommended configuration, and official prompt within the DeepSeek-V4-Flash answerer configuration; we standardize the data, token-F1 definition, and evaluation interface. token-F1 is computed directly against references and does not use the GPT-4o-mini judge.

**SocialMemBench token-F1.** token-F1 is reported by Q1–Q9 below. The answerer is DeepSeek-V4-Flash; each row follows its official code, recommended configuration, and official prompt, while the data, token-F1 definition, and evaluation interface are standardized. Values are computed directly against references rather than derived from GPT-4o-mini judgments. Full context appears only for this benchmark.

**EverMemBench token-F1.** EverMemBench token-F1 is reported only as an overall supplementary metric. The main table reports question-level binary accuracy by nine behaviors; token-F1 is not conflated with category accuracy or question format.

### F.5 GPT-5.6-LUNA MODEL CONFIGURATION

This configuration uses GPT-5.6-luna across memory construction, retrieval, and answer generation, while retaining GPT-4o-mini as the question-level judge, to assess cross-model robustness. BM25 and Embed use single-message top-10 retrieval, and Embed uses the local `all-MiniLM-L6-v2`. Full context is feasible only on SocialMemBench. Overall results are in main-text Table 1; category details for GroupMemBench and SocialMemBench are given in Tables 20 and 21, respectively. Only the overall EverMemBench result is reported under this model configuration.

**GroupMemBench category results.** This table uses the six categories defined above and contains 745 questions. The complete pipeline uses GPT-5.6-luna for memory construction, retrieval-time evidence interpretation, and answer generation, while GPT-4o-mini supplies the question-level binary judgments. Each method follows its official code, recommended configuration, and official prompt;

Table 18: GroupMemBench category token-F1 (%).

| Method | Abst. | Implicit | Multi-hop | Temporal | K.-update | Term-amb. | Total |
|---|---|---|---|---|---|---|---|
| BM25 | 25.6 | 33.5 | 25.8 | 36.4 | 24.2 | 2.0 | 25.0 |
| Embed | 24.2 | 27.0 | 17.1 | 14.2 | 22.4 | 4.4 | 17.4 |
| Mem0 | 26.0 | 7.0 | 5.0 | 0.0 | 13.0 | 1.0 | 9.0 |
| A-MEM | 15.0 | 15.8 | 8.3 | 5.9 | 19.7 | 4.8 | 10.6 |
| HippoRAG | 14.6 | 12.4 | 11.3 | 6.2 | 19.7 | 4.5 | 11.1 |
| SPEAKERMEM-R1$^{\dagger}$ | 14.5 | 26.6 | 22.3 | 49.7 | 26.5 | 5.1 | 25.2 |
| SPEAKERMEM-R1 | 15.1 | 25.6 | 26.3 | 50.9 | 26.7 | 4.4 | **26.5** |

Table 19: SocialMemBench Q1–Q9 category token-F1 (%).

| Method | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| BM25 | 13.8 | 10.3 | 12.1 | 38.1 | 18.7 | 14.5 | 15.7 | 12.2 | 8.2 | 15.7 |
| Embed | 15.2 | 12.5 | 10.0 | 39.8 | 20.8 | 18.4 | 18.6 | 15.8 | 7.8 | 18.1 |
| Mem0 | 12.0 | 7.0 | 16.0 | 7.0 | 10.0 | 12.0 | 11.0 | 10.0 | 10.0 | 11.0 |
| A-MEM | 22.3 | 62.4 | 24.1 | 67.3 | 29.2 | 51.9 | 23.4 | 24.7 | 20.9 | 31.9 |
| HippoRAG | 21.2 | 60.1 | 23.4 | 55.7 | 30.6 | 49.0 | 23.1 | 25.0 | 19.0 | 30.6 |
| Full context | 22.6 | 26.8 | 8.5 | 47.3 | 30.0 | 31.9 | 24.1 | 23.3 | 16.9 | 26.1 |
| SPEAKERMEM-R1$^{\dagger}$ | 27.9 | 5.9 | 26.5 | 35.4 | 34.7 | 4.6 | 30.8 | 30.9 | 33.9 | 27.4 |
| SPEAKERMEM-R1 | 22.8 | 56.5 | 25.0 | 70.9 | 30.9 | 52.4 | 24.9 | 26.0 | 22.4 | **32.7** |

we standardize only the data, metric definition, judge, and evaluation interface. BM25 and Embed use single-message top-10 evidence; SpeakerMem uses System 1 top-10 and System 2 owner-2 plus source-1. The entries are category-wise binary accuracies.

**SocialMemBench category results.** This table uses the Q1–Q9 partition defined above and contains 1,031 questions. The answerer is GPT-5.6-luna and GPT-4o-mini is the binary judge under the standardized data, metric, judge, and evaluation-interface protocol. Full context is shown only here because the longer benchmark inputs do not fit a practical context. The entries are category-wise binary accuracies.

**EverMemBench category results.** This model configuration is used for overall cross-model robustness analysis; the primary report of the nine behaviors uniformly uses the gpt-4.1-mini/Gemini-3-Flash configuration in main-text Table 2.

### F.6 Complete Path and Hierarchy Ablations

## G Full LoCoMo Category Breakdown

LoCoMo contains 1,986 questions in five categories (Maharana et al., 2024). Table 23 summarizes SPEAKERMEM-R1 results by category. The categories are Single-hop (one-step factual recall), Adversarial/AD (adversarial or misleading questions), Temporal (temporal relations and state changes), Multi-hop (reasoning over multiple dialogue facts), and Open-domain (requiring open-domain knowledge or cross-context reasoning).

The Single-hop and Temporal results show that this structure remains effective for two-person long-term conversations. The sharp drop on Multi-hop and Open-domain reveals that cross-segment composition and external knowledge remain major weaknesses. The main-text Table 4 gives the corresponding category-level positions of published baselines.

Table 20: GroupMemBench category accuracy with GPT-5.6-luna (%).

| Method | Abst. | Implicit | Multi-hop | Temporal | K.-update | Term-amb. | Total |
|---|---|---|---|---|---|---|---|
| SPEAKERMEM-R1 | 57.6 | 49.0 | 42.3 | 42.0 | 26.2 | 38.7 | 42.7 |
| BM25 | 71.9 | 49.0 | 43.4 | 56.8 | 27.1 | 18.9 | 46.2 |
| Embed | 70.5 | 44.9 | 29.7 | 25.3 | 19.6 | 25.5 | 35.3 |
| Mem0 | 82.0 | 20.4 | 6.6 | 0.6 | 0.9 | 12.3 | 20.27 |
| A-MEM | 65.5 | 30.6 | 21.4 | 1.2 | 13.1 | 34.9 | 26.58 |
| HippoRAG | 64.7 | 28.6 | 22.5 | 4.9 | 13.1 | 33.0 | 27.11 |

Table 21: SocialMemBench category accuracy with GPT-5.6-luna (%).

| Method | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| SPEAKERMEM-R1 | 68.7 | 72.8 | 4.5 | 83.8 | 80.4 | 53.7 | 58.6 | 56.9 | 80.0 | 64.4 |
| BM25 | 25.7 | 33.3 | 4.5 | 75.0 | 35.9 | 37.0 | 30.9 | 16.8 | 40.0 | 30.0 |
| Embed | 35.7 | 39.5 | 4.5 | 66.2 | 52.2 | 53.7 | 34.8 | 21.8 | 30.0 | 36.4 |
| Mem0 | 20.1 | 21.0 | 9.1 | 17.5 | 0.0 | 31.5 | 7.2 | 11.5 | 10.0 | 13.97 |
| A-MEM | 54.6 | 63.0 | 4.5 | 70.0 | 54.3 | 38.9 | 43.1 | 32.1 | 30.0 | 46.56 |
| HippoRAG | 61.4 | 77.8 | 4.5 | 77.5 | 69.6 | 61.1 | 62.4 | 51.9 | 70.0 | 61.30 |
| Full context | 71.1 | 79.0 | 4.5 | 82.5 | 71.7 | 66.7 | 72.9 | 69.8 | 100.0 | 71.3 |

Table 22: Accuracy (%) for dual-track path and hierarchy ablations. SM, GM, and EM denote SocialMemBench, GroupMemBench, and EverMemBench. All configurations use the same System 1 top-$k = 10$ and System 2 $k = 2$/source-$k = 1$. Row names match Figure 4: w/o S2 removes all four System 2 derived layers, w/o S1 removes the verbatim track, w/o S2-group removes Interaction and Insight, w/o S2-person removes Core and Profile, and SpeakerMem denotes the full dual-track system. The last three $\Delta$ columns are relative to the best full result for each benchmark.

| Configuration | SM Acc. | GM Acc. | EM Acc. | $\Delta$ vs. best full SM | GM | EM |
|---|---|---|---|---|---|---|
| w/o S2 | 54.80 | 42.01 | 48.92 | −14.40 | −5.89 | −12.98 |
| w/o S1 | 42.58 | 33.15 | 38.33 | −26.62 | −14.75 | −23.57 |
| w/o S2-group | 65.37 | 43.22 | 55.92 | −3.83 | −4.68 | −5.98 |
| w/o S2-person | 63.72 | 41.07 | 54.88 | −5.48 | −6.83 | −7.02 |
| SpeakerMem* | 69.2 | 47.0 | 60.5 | 0.0 | −0.9 | −1.4 |
| SpeakerMem | **64.9** | **47.9** | **61.9** | −4.3 | 0.0 | 0.0 |

*Supplementary retrieval ablation:* SpeakerMem* across the three domains is 69.2/47.0/60.5; SpeakerMem is 64.9/47.9/61.9; * means ASK disabled; SpeakerMem denotes the default ASK-enabled configuration.

Table 23: Question-level binary accuracy of SPEAKERMEM-R1 on all five LoCoMo categories.

| Category | Correct/total | Accuracy (%) |
|---|---|---|
| Single-hop | 655/841 | 77.88 |
| Adversarial/AD | 370/446 | 82.96 |
| Temporal | 227/321 | 70.72 |
| Multi-hop | 116/282 | 41.13 |
| Open-domain | 39/96 | 40.62 |
| All questions | 1407/1986 | **70.85** |
| Excluding AD | 1037/1540 | 67.34 |

## H RETRIEVAL TOP-$k$ SENSITIVITY

**Purpose and controlled setting.** This experiment examines how the two retrieval budgets jointly affect answer quality, distinguishing additional verbatim coverage from additional derived records. We reuse the fixed memory store constructed for the main experiments on SocialMemBench held-out eval10 (10 networks, 305 questions), using dual-track querying without ASK (for this sensitivity study, S1 recall-$n$ is set equal to S1 top-$k$, so no larger candidate pool is recalled before Select). We evaluate all nine combinations of S1 top-$k \in \{5, 10, 20\}$ and S2 $k \in \{2, 4, 6\}$, with S2 `source_k`=1 throughout. The two varied budgets concern S1 raw candidates and S2 owner/row retrieval, respec-

tively; the source budget does not vary. No RL/SFT/Qwen Writer is used, so this is a retrieval-budget study with fixed memory. Acc is binary accuracy evaluated by GPT-4o-mini, and token-F1 is computed from per-question predictions and references. Both metrics are reported as percentages.

**S1: additional coverage helps, but returns per budget slot diminish.** S1 preserves original messages with speaker, time, and local context, supplying verifiable evidence for S2 records and wording that compression may omit. Increasing S1 top-$k$ from 5 to 10 and then to 20 improves both Acc and token-F1 at every tested S2 budget, indicating that small verbatim budgets still miss useful information. At S2 $k = 4$, for example, Acc rises from 56.39% to 64.26% and 73.11%, while token-F1 rises from 23.85% to 25.00% and 26.53%. However, the first increment adds five candidate slots and the second adds ten, so their total gains alone do not measure diminishing returns. For S2 $k = 2, 4, 6$, the average Acc gain per added S1 slot falls from 2.03, 1.57, and 1.25 percentage points over 5–10 to 0.89, 0.89, and 0.56 over 10–20. Thus, *returns per S1 budget slot slow markedly, suggesting a trend toward saturation*: after high-value messages have been covered, additional candidates may increasingly repeat existing evidence or contribute weaker clues. Total Acc gains over 10–20 remain 5.58–8.86 points, while token-F1 marginal gains are smaller and less uniform.

**S2: a moderate increase helps, whereas excess records can impair judgment.** S2 supplies addressable derived states constrained by PERSON/GROUP scope, source/owner, event, and time. Increasing S2 $k$ from 2 to 4 improves both metrics at all three S1 budgets: Acc gains are 7.21, 4.92, and 4.91 points, and token-F1 gains are 0.96, 1.44, and 0.65 points. This is consistent with a moderate expansion recovering complementary participant, relation, and state clues. Increasing S2 further from 4 to 6 is less reliable. At S1 top-$k = 10$, Acc rises only from 64.26% to 65.57%, while token-F1 falls from 25.00% to 24.75%. At S1 top-$k = 20$, Acc falls from 73.11% to 71.15% and token-F1 from 26.53% to 25.78%, decreases of 1.96 and 0.75 points, respectively. One explanation consistent with the dual-track design is that excess derived records introduce duplicate facts, overlapping descriptions, or weaker relational and historical clues, diluting decisive evidence and increasing the answerer's judgment burden. S2 should therefore retain a moderate set of complementary records rather than maximize quantity. The decline is directly observed; redundant or distracting records are one plausible mechanism.

**Dual-track interaction: S2's benefit depends on verbatim coverage.** Figure 6 shows that the effect of extra S2 budget depends on S1. With S1 top-$k = 5$, increasing S2 from 4 to 6 still improves Acc/token-F1 from 56.39/23.85 to 59.34/24.35, consistent with additional derived clues filling gaps in limited raw coverage. The same S2 increment decreases both metrics when S1 top-$k = 20$: once S1 supplies more complete original evidence, the complementary value of additional structured records may no longer outweigh redundancy and distraction. This pattern supports separate budget control for the two tracks. S1 expands verifiable verbatim coverage, while S2 should provide a compact set of query-relevant states and relations. It also fits Anchor–Separate–Resolve–Compose: attribution, scope, and versions are constrained before evidence is composed, and the two budgets need not be maximized together.

**Acc and token-F1 jointly inform budget selection.** Acc assesses task-level correctness, whereas token-F1 measures surface token overlap, so the metrics need not move together. With S1 top-$k = 10$, increasing S2 from 4 to 6 raises Acc but lowers token-F1. A single metric could therefore overlook changes in answer completeness, concision, or wording, although this discrepancy alone cannot identify the responsible factor. Among the nine configurations, S1 top-$k = 20$ and S2 $k = 4$ jointly maximize Acc (73.11%) and token-F1 (26.53%), suggesting a useful quality balance between sufficient verbatim evidence and a moderate structured budget. This is the best observed quality balance in the tested grid; the main experiments retain S1 top-$k = 10$ and S2 $k = 2$/source-$k = 1$. Overall, the results favor monitoring marginal S1 returns and keeping S2 moderate.

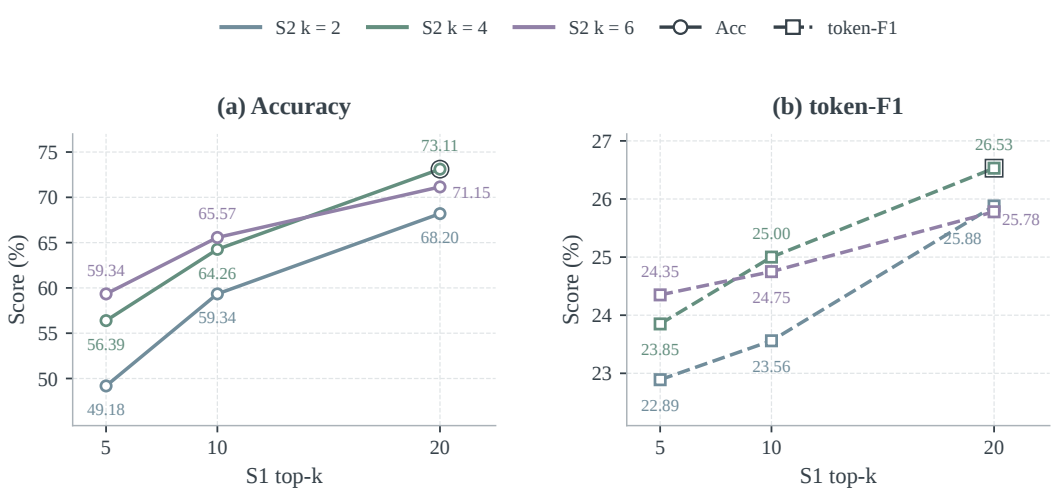


Figure 5: Retrieval-budget response curves. Accuracy (left) and token-F1 (right) are plotted against S1 top-$k$; colors distinguish S2 $k = 2, 4, 6$, and solid circles versus dashed squares identify the metrics. Panels use separate y-axis ranges to reveal smaller token-F1 changes; visual slopes should therefore not be compared across metrics. Outlined markers identify the best observed configuration for each metric. Lines connect measured configurations only. The setting uses the main-experiment memory store, 305 questions, no ASK, and S2 source-$k = 1$.

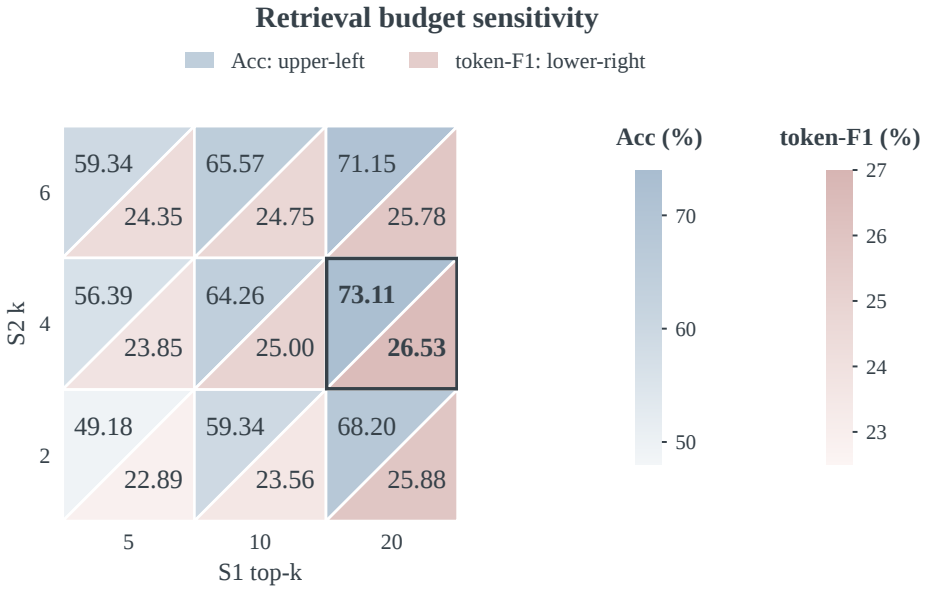


Figure 6: Joint S1/S2 configuration grid. The horizontal axis enumerates discrete S1 top-$k$ settings and the vertical axis gives S2 $k$. Each cell reports exact percentages: the upper-left blue triangle encodes Acc and the lower-right rose triangle encodes token-F1. Independent color scales prevent direct cross-metric comparisons of shade. The outlined cell, $(20, 4)$, is the best observed configuration for both metrics. These are the same nine observations as Figure 5, not a separate experiment.

# I Complete Prompts

This section provides snapshots of the key prompts used in the experiments: Writer, Project, System 1 Select/Expand, Sufficiency/ASK, the frozen answerer, binary judging, and SocialMem Mean-Q/MeanN scoring. Fields in braces are filled at runtime; except for the message wrapper required by the model API, the text below is neither summarized nor rewritten. This makes the module responsibilities and actual output formats in the main text directly auditable.

Table 24: Input/output formats and training boundaries of key prompts.

| Module | Input | Output format | Boundary |
|---|---|---|---|
| Writer | Roster, local message segment, and current derived-layer heads | ADD/UPDATE/NOOP JSON; does not generate answers | Qwen2.5-3B; trained with SFT/R1 |
| Project | Question and channel roster | issue, PERSON/GROUP rows, `head/full`, and relation constraints | Frozen; does not read global answer candidates |
| Select/Expand | Question and System 1 verbatim candidates | Necessity ranking and at most two adjacent-window anchors | Frozen; applies only to System 1 |
| Sufficiency/ASK | Truncated System 1 evidence | `enough` and an optional search term | Frozen; at most one supplementary retrieval |
| Answer | Question, final evidence from both paths, empty rows, and temporal mode | Final natural-language answer | Frozen; does not modify memory |
| Binary judge | Question, gold answer, and system answer | Question-level 0/1 decision and structured rationale | Both DeepSeek-V4-Flash and GPT-5.6-luna rows use GPT-4o-mini for judging; the judge for EverMemBench cross-configuration results is given in Table 2 |
| SocialMem scoring protocol | Question, gold answer, system answer, and scoring protocol | Partial scores used by MeanQ/MeanN | Both DeepSeek-V4-Flash and GPT-5.6-luna rows use GPT-4o-mini for judging; the judge for EverMemBench cross-configuration results is given in Table 2 |

### I.1 PROMPT INVENTORY

Table 24 lists the input, output, and frozen boundary for each prompt; the following subsections preserve the complete copyable text. Except for the Writer, query, answer, and judging prompts are fixed in the main experiments and ablations.

### I.2 WRITE: STRUCTURED WRITING

The Writer input contains the channel roster, the latest state of derived memory, and a new local message segment. The model generates only ADD, UPDATE, or NOOP; time, session, verbatim `from_ids`, and update-chain fields are added by the system after the action passes validation. The complete prompt follows.

**Prompt 1: Writer**

```
[SYSTEM PROMPT]
You are an ONLINE memory manager for a MULTI-PARTY group chat, maintaining a
    speaker-indexed layered memory.

PREMISE: every raw message of this segment is ALREADY stored VERBATIM (raw
    track). Your job is NOT to restate the raw text, but to add STRUCTURED
    derived memory on top of it, so that retrieval by person / relationship /
     time becomes possible.

FOUR DERIVED LAYERS YOU MAY WRITE
(raw messages are already written to per_speaker_episodic by the system -
    never produce that layer)
- per_speaker_core     a person's STABLE fact / identity / hardened stance /
    RECURRING BEHAVIOUR
- per_speaker_profile  how the group sees a person, or one person's
    OBSERVATION about another
                       ("A says B is X" -> owner=B, source=A)
- group_interaction    cross-speaker event / relationship / group DECISION
    (owner=GROUP)
                       *** WHO SAID IT IS NOT WHO IT BELONGS TO ***
                       A decision announced by ONE person on behalf of the
    group is STILL a
                       group decision: owner=GROUP, source=<the speaker>.
```

```
                        Triggers: "then it is settled", "so we agreed", "that
    is decided",
                        "we will go with X" -> owner=GROUP, layer=
    group_interaction,
                        utype=decision. Do NOT file it as that speaker's
    personal preference.
                        (A personal preference is "I would rather X"; a
    decision is "X is settled".)
- group_insight         group NORM / consensus / meta-observation
    (owner=GROUP)
                        ONLY when the conversation gives evidence, note which
    member conspicuously
                        relates to a norm differently. NEVER guess a member's
    compliance.

ACTIONS - exactly three, output nothing else
  {"action":"ADD","owner":"<who it is ABOUT>","source":"<who SAID/observed it
    >","layer":"<one of the four above>","utype":"fact|stance|observation|
    decision|relation","content":"<the memory>"}
  {"action":"UPDATE","entry_id":"<existing entry being updated>","content":"<
    the NEW, current state>"}
  {"action":"NOOP"}

UPDATE BOUNDARIES (important)
- *** BEFORE EVERY ADD, SEARCH CURRENT MEMORY FIRST *** Is there already an
    entry about this
  same matter / same work item / same person-and-topic? If yes -> UPDATE that
     entry.
  ADD is only for matters that NOTHING in CURRENT MEMORY covers yet.
- Use UPDATE whenever an existing entry about the SAME MATTER has MOVED ON.
    Read this broadly:
    * a number / percentage / progress advanced   e.g. 38% complete -> 97%
    complete
    * a date or deadline shifted                  e.g. freeze 2025-06-20 ->
    2025-06-27
    * a status moved                              e.g. blocked -> resolved,
    planned -> shipped
    * a stance / belief / plan changed
    * a later, more specific statement supersedes an earlier vaguer one about
     the same item
  It does NOT have to contradict the old entry. Any progression on the same
    item is an UPDATE.
  A recurring behaviour observed AGAIN is also an UPDATE (record that it
    happened again).
- *** THE NEW CONTENT MUST DIFFER SUBSTANTIVELY FROM THE ENTRY BEING UPDATED.
     *** If what you
  would write is the same as the existing entry (same value, same status,
    same wording),
  output NOOP - not UPDATE. An UPDATE that changes nothing is worse than no
    action: it buries
  the real history under identical links.
- entry_id MUST come from the CURRENT MEMORY list below.
- Give ONLY the new state. How the old entry is preserved and chained is
    handled by the system.
- Do NOT restate the old content (never write "was X, now Y" - write only Y).
- Do NOT repeat owner / source / layer - the system reuses them from the
    updated entry.
- DISAGREEMENT BETWEEN DIFFERENT PEOPLE is always separate ADDs. Keep them as
     separate ADDs; do not use UPDATE.
- ONE MATTER = ONE ENTRY. If a matter is ALREADY in CURRENT MEMORY - or you
    already wrote an
```

```
  ADD for it EARLIER IN THIS SAME OUTPUT - never ADD a second entry for it:
    UPDATE it if its
  state changed, otherwise output nothing. This holds for every kind of
    memory (not just
  stances), for the GROUP row no matter who voices it this time, and however
    differently the
  matter is worded. Merely re-stating / confirming / agreeing with something
    already listed is
  not new state -> neither ADD nor UPDATE.

GRANULARITY - PREFER FINE OVER COARSE (important)
- One derived memory = ONE atomic fact / ONE stance / ONE observation.
- If a person said N distinct things in this segment -> produce N entries,
    never one summary.
  BAD : "Mum's overall view on the 40th: wants a party, worried about budget,
     March is awkward"
  GOOD: three separate entries (celebration format / budget / timing)
- But do NOT copy raw messages verbatim into derived memory (raw is already
    stored).

CAPTURE BEHAVIOUR, NOT ONLY STATED FACTS
Explicit facts ("closed a Series A", "the code is 4-8-1-9") are easy - you
    will not miss them.
Equally important and far more often missed are IMPLICIT traits: a habit, an
    avoidance, a role
someone keeps taking on, a subject someone never engages with, a way of
    replying that repeats,
who they defer to, what they joke about instead of answering.
  e.g. "deflects questions about his own career with a joke, then redirects
    to someone else"
       "is always the one who handles logistics" / "never comments on anyone'
    s promotion"
File them as per_speaker_core (the person's own pattern) or
    per_speaker_profile (how others
read them). Base them on what actually happened - describe the behaviour,
    never guess a motive.
A pattern occurring AGAIN is NOT a duplicate, it is EVIDENCE: UPDATE that
    entry to record that
it happened again (and when), instead of NOOP.
Someone who talks a lot but states few hard facts should still end up with
    entries - if a person
spoke repeatedly this segment and you wrote nothing about them, you have
    missed their behaviour.

OTHER CONSTRAINTS
- owner = who the memory is ABOUT; source = who said/observed it.
- If A claims something about B -> owner=B, source=A. Never file B's matter
    under A.
- Invent nothing unsupported. Prefer faithful wording over paraphrase.
- Write memory content in the SAME LANGUAGE as the conversation (so that
    retrieval embeddings
  of memory and of raw text live in the same space).

Output JSON only, no explanation: {"actions":[ ... ]}

[USER TEMPLATE]
Speakers present: {speakers}
Session: {session}    Time: {ts}

CURRENT MEMORY (ALL derived memory of this group; each slot shows its LATEST
    state only)
```

```
format: <entry_id> [layer|owner(by source)@date] content
{state}

NEW MESSAGES IN THIS SEGMENT
{convo}

HARD OUTPUT LIMIT: return at most {max_actions} actions. Stop after the last
    NOVEL action, close the JSON immediately, and NEVER repeat an action
    already emitted in this output.

Output {{"actions":[ ... ]}}
```

### I.3 PROJECT: QUERY PROJECTION

Project outputs only `issue`, `rows`, and `tense`. `rows` determine the PERSON/GROUP views and `tense` determines whether to use only current nodes or expand update chains; channel and source/owner constraints are enforced by the roster and within-row retrieval.

**Prompt 2: Project**

```
[SYSTEM PROMPT]
Decide what SHAPE of slice to take from a group-chat memory in order to
    answer the question.

Output three fields:

- issue : which ONE matter the question is about. A short noun phrase; it
    will be used for
          semantic matching inside each person's own memory.
          e.g. "how to celebrate the 40th anniversary" / "deadline for the
    field freeze"
          If the question is broad, use its core noun phrase.

- rows  : whose rows to take. OUTPUT A LIST. Elements may be:
          "ALL"      the whole roster - needed for a per-person roll-up, OR
    to determine
                     WHO NEVER did something
          "GROUP"    the group-level conclusion / norm / decision
          "<name>"   a specific person; several names may be listed to form a
     subset
          e.g. ["ALL"] / ["GROUP"] / ["Mum"] / ["Mum","Dad"] / ["Eli","GROUP
    "]
          *** WHEN IN DOUBT, OUTPUT ["ALL"] ***
          A too-wide slice only adds rows the answerer can ignore; a too-
    narrow slice loses
          the answer permanently. Only give a narrow row list when the
    question plainly
          concerns exactly those people and nobody else.
          COMPOUND QUESTIONS: if the question contains TWO OR MORE separate
    sub-questions
          ("who does A report to, AND who has the final call?"), the rows
    MUST cover EVERY
          sub-question. If you cannot confidently name the rows for all of
    them -> ["ALL"].
          WHEN TO ADD "GROUP": group decisions, norms, ownership/
    responsibility statements
          and shared facts are filed in the GROUP row, so add "GROUP"
    whenever the question
```

```
          could touch any of those - it combines freely with names or with "
    ALL".
          THE ONE EXCEPTION: if the question asks what EACH PERSON thinks /
    where each person
          STANDS (a per-person stance roll-up), do NOT add "GROUP" - the
    group's conclusion
          would override the individuals' own positions.

- tense : "head" only the current / latest state
          "full" the change process OR AN EARLIER STATE is needed. Use "full"
     when the
                question either
                  (a) asks how it changed / what it used to be / why it
    changed, OR
                  (b) PINS A PAST TIME POINT - "on June 2", "back in March",
     "at the time",
                      "when X happened", "originally", "at first", "before Y
    ".
                For (b) the current value may have been superseded since
    that moment, so the
                head alone would answer about the WRONG point in time.
          WHEN IN DOUBT between the two, choose "full": extra history is
    harmless, a missing
          earlier state cannot be recovered.

GUIDELINES
- "What does each person think" and "who NEVER does X" are BOTH rows=["ALL"].
  The former reads the filled cells, the latter reads the EMPTY cells - same
    slice.
  Note: "who violates the norm" is about a norm, but the norm is only the
    criterion -
  the answer lives in the EMPTY cells, so take every member's row, NOT ["
    GROUP"].
- "What was finally decided" / "what is it now" -> rows=["GROUP"], tense="
    head"
- "How did X's view change" -> rows=["X"], tense="full"
- "What do A and B each think" -> rows=["A","B"]
- "What did A and B each mean ON <past date>" -> rows=["A","B"], tense="full"
  (the date pins a past moment - their current state may already be different
    )
- "Who does A report to, and who owns X?" -> two sub-questions, the second
    one is an
  ownership statement -> rows=["A","GROUP"] (or ["ALL","GROUP"] if unsure),
    tense="head"
- CROSS-PERSON questions ("what does A think OF B", "how does A see B"):
    memory is filed
  under WHO IT IS ABOUT, so the row is B, not A -> rows=["B"].
- When unsure whether a member is relevant, prefer ["ALL"]; do not risk
    missing anyone.

Output JSON only: {"issue":"...", "rows":[...], "tense":"head|full"}

[USER TEMPLATE]
Question:
{question}

Roster of this group: {roster}

Return {{"issue":"...","rows":[...],"tense":"head|full"}}
```

### I.4 SELECT AND EXPAND: VERBATIM-TRACK SELECTION

This prompt operates only on System 1 verbatim candidates. It returns both a necessity ranking and at most two adjacent-window expansion anchors. Multi-person coverage is handled by System 2 and is therefore not a condition for triggering Expand.

**Prompt 3: System 1 Select / Expand**

```
[SYSTEM PROMPT]
You are selecting RAW DIALOGUE EXCERPTS to answer a question about a multi-
    party group chat.
All candidates are verbatim original messages, tagged (speaker / session /
    turn / time).

Do two things:

1. ranked - order candidates by how NECESSARY they are to answer; most
    necessary first.
   Aim: the first few alone should suffice.

2. expand - USE SPARINGLY, DEFAULT EMPTY.
   Decide by asking WHAT IS ACTUALLY MISSING right now:
   - Missing a SPECIFIC EXACT VALUE (number / date / time / code / amount) or
     the full detail
     of one event, AND the candidates clearly do not contain it
       -> you may fill it, at most 1-2 entries.
   - Missing COVERAGE of some people / need a per-person roll-up
       -> leave EMPTY. Per-person coverage is handled by a separate retrieval
     route;
          pulling more raw context cannot fill that gap.
   When in doubt, leave it empty.

Mind speaker attribution; the same matter may be described differently at
    different times.
Output JSON only: {"ranked":[candidate numbers, ...], "expand":[at most 2,
    usually empty]}

[USER TEMPLATE]
Question:
{question}

Candidate raw excerpts:
{candidates}

Return {{"ranked":[ ... ], "expand":[ ... ]}}
```

### I.5 SUFFICIENCY AND ASK: SUPPLEMENTARY RETRIEVAL DECISION

Sufficiency independently checks the System 1 evidence that will be passed to the answerer and outputs only `enough` and one optional search term. If `enough=false`, the system uses `ask` to retrieve and rerank once more; this module does not generate the final answer.

**Prompt 4: Sufficiency / ASK**

```
[SYSTEM PROMPT]
This is the material that will be used to answer the question - all of it,
    nothing else.
```

```
The store holds much more; anything phrased unlike the question is not in
    front of you.

First commit to a verdict: does this material ALREADY contain the answer?
  it does           -> {"enough": true,  "ask": ""}
  something missing -> {"enough": false, "ask": "<ONE terse query>"}

Count it as missing when: there is an answer you are looking for but have not
     actually
found, or a matter is mentioned only vaguely with the substance absent. Do
    not settle for
material that is merely on-topic - being about the right subject is not being
     the answer.
Keep ask TERSE: bare keywords (entity / event / thing), never a full question
    .
Say enough=true when what is missing is only per-person coverage (handled
    elsewhere).

Output JSON only.

[USER TEMPLATE]
Question:
{question}

Material:
{material}

Return {{"enough":true|false, "ask":""}}
```

### I.6 ANSWER: FROZEN ANSWERER

The answerer receives speaker/time-labeled verbatim evidence and a sparse structured view organized by PERSON/GROUP rows. The prompt explicitly forbids replacing an individual position with a group conclusion and specifies how to interpret `head/full` and empty rows.

**Prompt 5: Answer**

```
[SYSTEM PROMPT]
Answer a question about a MULTI-PARTY group conversation using ONLY the
    memory provided below.

You are given two parts:

[RAW EXCERPTS]  verbatim original messages tagged (speaker @ time)
                use for exact values, quotes, concrete detail

[MEMORY SLICE]  structured memory organised BY PERSON; each cell looks like:
    <person>  *current: <content>
                - history: <older content @time> ...  (given only when change
     is relevant)
    <person>  (none)                                  (this person has NO
    record on this matter)

RULES
1. Attribute every fact / stance to the CORRECT person. Never file A's words
    under B; keep different people's views separate in the answer.
2. If the slice lists MULTIPLE people -> answer for EACH of them, omitting no
     one.
```

```
   Do NOT replace an individual's own stance with the group's final outcome.
   (the group settling on a dinner does not mean the person who wanted a
    party changed their mind)
3. If the slice contains ONLY the GROUP row -> give that single conclusion
    directly; do not expand into a per-person list, and do not enumerate
    historical values.
4. When '*current' is shown, use it for "now / currently / what was finally
    decided" questions; use the history only when the question asks how it
    changed / what it used to be / why.
5. For "who never did X / who does not follow the norm":
   (none) MEANS: this person has NO record at all on this matter -> they
    NEVER did it.
   THE ANSWER IS THE PEOPLE MARKED (none). Do not call it 'cannot be
    determined'.
   Distinguish two cases and do not confuse them:
     - cell is (none)                    -> no record whatsoever = NEVER did
    it   <- THIS is the answer
     - cell has content but about something else -> this person DOES have
    records;
       do NOT conclude they never did it merely because the shown content is
    a different matter.
   Only if EVERY cell is (none) does the slice carry no signal - then fall
    back to the RAW EXCERPTS.
6. (none) means 'no DERIVED record on this matter', NOT 'this person did
    nothing'. The RAW
   EXCERPTS are equally valid evidence - when the slice is silent but the
    excerpts clearly show
   the answer, use the excerpts. Only say the memory is insufficient when
    BOTH parts are silent.
7. Reasonable implicit inference from the material is fine. But if the answer
     can ONLY be
   reached by guessing, or by picking whichever candidate looks most
    plausible, then the
   material does not contain it - reply exactly:
   There is no information available in the conversation to answer this
    question.

Be concise; but when asked about several people, account for every one of
    them.

[USER TEMPLATE]
Question:
{question}

[RAW EXCERPTS]
{passages}

[MEMORY SLICE]
{slice}

Answer using the material above.
```

### I.7 Evaluation Prompts

The question-level binary accuracy in the main tables uses the following fixed judging prompt. The user message always contains the question, gold answer, and system answer.

**Prompt 6: Binary Judge**

```
[SYSTEM PROMPT]
You are a strict judge evaluating whether an agent's answer matches the gold
    answer for a question.
Consider paraphrases correct if they have the same meaning as the gold answer
    .
First provide a brief reasoning paragraph. Then provide the final judgment on
     a new line using the format:
Final: Correct
or
Final: Incorrect

[USER TEMPLATE]
Question:
{question}

Gold Answer:
{gold_answer}

Agent Answer:
{agent_answer}
```

SocialMem MeanQ/MeanN use the benchmark scoring protocol with partial credit: MeanQ averages item-level scores, while MeanN first averages within dialogue networks and then averages across networks. The complete scoring protocol follows.

**Prompt 7: SocialMem scoring rubric**

```
[SYSTEM PROMPT]
You are an expert grader evaluating AI answers about social group
    conversations.
You score answers on a 0.0-1.0 scale based on factual correctness only.

[USER TEMPLATE]
Score the generated answer against the gold answer on a 0.0-1.0 scale.

CRITICAL RULES (apply before all others):
1. A concise correct answer scores THE SAME as a verbose one. Do NOT penalize
   brevity or reward evidence citation. If the core fact is correct, score
    1.0.
2. Attributing a fact to the wrong person is always a critical error -> score
     <= 0.3,
   regardless of how much other content is correct.
3. "NOT ENOUGH INFORMATION" or "[No memories retrieved]" always scores 0.0.

General scoring protocol:
- 1.0: Core fact correct, correct person/attribution, required parts present
- 0.7-0.9: Right answer but one minor part missing or slightly imprecise
- 0.4-0.6: Right direction but wrong detail, missing key element, or
    ambiguous
- 0.0-0.3: Wrong attribution, wrong fact, contradiction, or no answer

Q-type specific rules:
- Q1/Q3: Core fact + correct person = 1.0. For Q3, score = fraction of group
    members correctly recalled.
- Q4 (attribution): Correct speaker named AND foil NOT named -> 1.0. Correct
    + names foil -> 0.7. Wrong -> 0.0-0.3.
```

```
- Q5 (theory of mind): BOTH parts required for 1.0 (preference + who revealed
     it with observable action). One part -> 0.5.
- Q8 (temporal shift): old state(+0.33)+new state(+0.33)+trigger(+0.33). Both
     states no trigger -> 0.7. One state -> 0.4.

Question: {question}
Gold answer: {golden_answer}
Generated answer: {generated_answer}

Output JSON only - no preamble:
{{"score": 0.0, "rationale": "one sentence"}}
```