ROXY: Generative Indexing and Conflict-Aware Reranking for Long-Horizon Conversational Memory

Anonymous ACL submission

Abstract

Long-horizon conversational assistants must answer questions grounded in multi-session dialogue, yet external memory faces two key challenges: (1) retrieval mismatch—queries often lack cues to surface relevant memories, and (2) conflict resolution—retrieved evidence contains fragmented information, negations, or overrides that similarity- and rule-based methods cannot reliably handle. We propose **ROXY**, a retrieval-oriented memory framework that integrates **Generative Indexing** and **Conflict-Aware Reranking** to address both challenges. For retrieval, ROXY performs *Generative Indexing* (GI): an LLM generates anticipatory cue questions for each memory chunk and indexes them alongside the original content. For conflicts, a **conflict-aware LLM judge** reasons over high-recall candidates conditioned on the query, selecting logically coherent evidence without hand-crafted rules—whereas similarity-based methods fail to distinguish semantically similar but contradictory memories. On the **LoCoMo** benchmark under a MemInsight-aligned evaluation protocol with the same embedding and answer-generation backbone, ROXY achieves **89.3** Recall@5 and **41.2** F1, outperforming MemInsight by **+28.8** and **+11.1** points respectively, with strong gains on single-hop, temporal, multi-hop, and adversarial questions. These results show that anticipatory indexing combined with adaptive conflict reasoning offers a scalable solution for memory-grounded conversational agents. Code and prompts will be publicly available.

1 Introduction

Long-horizon conversational assistants are increasingly expected to operate over weeks of interaction: users return to past decisions, revisit earlier plans, and ask questions whose answers are scattered across many sessions. Yet even strong LLMs remain bounded by a fixed context window, and performance degrades when required evidence lies far outside the prompt. This challenge is made explicit by LoCoMo, a benchmark of very long-term, multi-session conversations where questions demand single-hop recall, multi-hop reasoning across sessions, temporal grounding, open-domain contextual understanding, and robustness to adversarial/unanswerable queries (Maharana et al., 2024). In such settings, improving *retrieval from conversational memory* is often the decisive factor: when the right experiences are not surfaced, answer generation becomes unreliable regardless of the generator's strength. A common approach attaches an external memory store and performs retrieval-augmented generation (RAG) at inference time (Lewis et al., 2020). Recent systems explore long-term memory management and compression (e.g., MemoryBank, MemGPT) (Zhong et al., 2024; Packer et al., 2023), as well as reading agents that maintain gist-like representations for extremely long contexts (Lee et al., 2024). Despite meaningful progress, many memory pipelines still depend on matching a runtime query to *statement-like* stored memories (turns, summaries, or metadata), leading to a persistent cue mismatch: users typically pose *questions* with new wording, intent, and perspective, while memories were stored as *past utterances*. As a result, relevant information can be

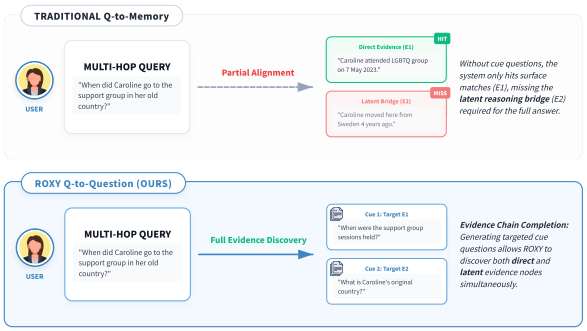

Figure 1: Overview of Generative Indexing. The model generates anticipatory queries for memory chunks to bridge the gap between retrieval cues and stored content.

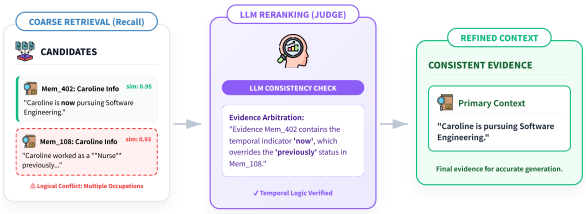

Figure 2: Overview of Conflict-Aware Reranking. A set-conditioned LLM judge evaluates retrieved candidates to resolve contradictions and select consistent evidence.

present in the memory bank but remain effectively unreachable under future query forms.

Motivated by the *Encoding Specificity Principle* (Tulving and Thomson, 1973; Morris et al., 1977), ROXY addresses this mismatch by *proactively injecting future-oriented retrieval cues at indexing time* (Figure 1). For each memory chunk, an LLM generates plausible questions that users might ask, incorporating these cues into the index—akin to document expansion (Nogueira et al., 2019a) but specialized to conversational memory.

Long-horizon conversational memory introduces a second difficulty that is often underemphasized: retrieval is not only about semantic similarity, but also about selecting *logically compatible* evidence across distributed sessions, partially matching contexts, and logical overrides. In multi-session conversations, multiple retrieved memories may be individually relevant yet mutually contradictory, which can break multi-hop reasoning or produce inconsistent answers. Prior work commonly addresses this by designing explicit memory augmentation, prioritization, or evolution mechanisms. In particular, our primary comparison target MemInsight proposes autonomous memory augmentation and includes a *Priority* strategy tailored to better handle multi-hop reasoning (Salama et al., 2025). While rule-based strategies can handle specific conflict patterns, enumerating them for open-ended conversation is infeasible. By contrast, utilizing an LLM-based judge leverages generalized semantic reasoning to arbitrate conflicts dynamically, adapting to nuanced contradictions that rigid rules cannot capture. ROXY adopts this *evidence selection* approach: specifically, it uses a set-conditioned conflict-aware judge to rerank candidates (Figure 2), explicitly avoiding hand-crafted evolution heuristics.

We evaluate ROXY on LoCoMo under an experimental protocol aligned with MemInsight (Salama et al., 2025; Maharana et al., 2024). To isolate the effect of Generative Indexing and conflict-aware evidence selection, we keep the embedding and generation configuration identical to MemInsight and reuse the same category-aware generation prompting. Under this controlled setting, ROXY achieves **89.3** Recall@5 and **41.2** overall F1 on LoCoMo, improving over MemInsight's best reported results by **+28.8** Recall@5 and **+11.1** F1, with particularly strong gains on single-hop, temporal, and adversarial questions.

Contributions.

- We propose **ROXY**, a retrieval-oriented conversational memory framework that combines **Generative Indexing** and **Conflict-Aware Reranking** to bridge cue mismatch and resolve contradictions.

- We instantiate Generative Indexing (GI) with two complementary strategies—*cue fusion* (concatenation) and *cue separation* (multi-vector)—both targeting the core cue-mismatch problem by integrating anticipated cue questions into the retrieval index.

- We introduce a **set-conditioned, conflict-aware reranking** mechanism that resolves contradictions among retrieved candidates at retrieval time, avoiding hand-crafted memory evolution rules.

2 Related Work

2.1 Long-Horizon Conversational Memory and Evaluation

LLM agents operating over weeks of interaction require memory systems that persist beyond fixed context windows (Packer et al., 2023; Zhong et al., 2024). Generative agents demonstrate how such memory streams support believable simulacra of human behavior (Park et al., 2023). To evaluate these capabilities, benchmarks like LongBench (Bai et al., 2023) and LongMemEval (Wu et al., 2024) assess long-context understanding, while NoLiMa (Modarressi, 2025) targets non-literal consistency. Most relevant to our work is LoCoMo (Maharana et al., 2024), which specifically targets very long-term, multi-session QA. We adopt LoCoMo's protocol and compare against strong baselines including ReadAgent (Lee et al., 2024) and MemInsight (Salama et al., 2025).

2.2 Addressing Retrieval Mismatch via Generative Indexing

A fundamental challenge in memory retrieval is **retrieval mismatch**: future queries often differ linguistically from stored past utterances. Traditional dense retrieval (Karpukhin et al., 2020; Izacard et al., 2021) and sparse methods (BM25, SPLADE) (Robertson and Zaragoza, 2009; Formal et al., 2021) rely on direct semantic modulation, which can be brittle under diverse query forms. Conversational search attempts to bridge this gap by rewriting queries at runtime (Anantha et al., 2020; Voskarides et al., 2020; Yu et al., 2021), but this adds latency and propagates errors. An alternative from IR is document expansion (e.g., Doc2Query, InPars), which synthesizes potential queries at indexing time (Nogueira et al., 2019b; Bonifacio et al., 2022; Dai et al., 2022; Wang et al., 2023). ROXY adapts this strategy to conversational memory via *Generative Indexing*: by generating anticipatory cue questions for each memory chunk, we enable robust query-to-query matching without runtime rewriting.

2.3 Conflict Resolution and Evidence Selection

Long-term memory inevitably accumulates contradictions due to temporal updates or user corrections. While Model Editing approaches like ROME and MEMIT edit weights directly to resolve conflicts (Meng et al., 2022a,b; De Cao et al., 2021), they are risky for user-specific data. Retrieval-based systems often rely on hand-crafted rules or heuristics, such as MemInsight's priority scoring (Salama et al., 2025). ROXY instead treats conflict resolution as a dynamic evidence selection problem. Inspired by "LLM-as-a-Judge" and self-consistency frameworks (Zheng et al., 2023; Wang et al., 2022), we employ a conflict-aware reranker to arbitrate among retrieved candidates, filtering invalid or contradictory evidence based on the specific query context.

3 Methodology

ROXY consists of two main stages: offline generative indexing to bridge cue mismatch, and online conflict-aware reranking to resolve contradictions.

3.1 Memory Bank

We flatten a multi-session conversation into a memory bank $\mathcal{B} = \{b_1, \ldots, b_N\}$. Each memory chunk stores the utterance content x_i and auxiliary metadata $\mathcal{M}_i$:

$$b_i = (x_i, \mathcal{M}_i). \tag{1}$$

In ROXY, **indexing and retrieval are performed on x_i only**. Metadata $\mathcal{M}_i$ (e.g., timestamps) is preserved for downstream answer generation, especially for temporal questions, but is not used to build the retrieval index.

Memory chunk definition. Each memory chunk x_i corresponds to a *single dialog turn*, maintaining one-to-one alignment with LoCoMo's turn-level evidence annotations (e.g., "D1:3" = Session 1, Turn 3).

```
speaker said, "text"
```

For example: "*Caroline said, 'I went to the LGBTQ support group.*'" This design has three key properties: (i) no overlap between chunks—each turn is atomic; (ii) no multi-turn aggregation—preserving LoCoMo's granularity and ensuring Recall@k metrics are directly comparable with gold evidence; (iii) timestamps are recorded in $\mathcal{M}_i$ and provided only during answer generation for temporal reasoning.

3.2 Offline Generative Indexing with Cue Questions

To bridge cue mismatch, ROXY generates *cue questions* for each memory chunk at indexing time.

Cue question generation. For each memory chunk b_i, we generate n_q questions that a user might ask such that x_i provides the answer:

$$\mathcal{Q}_i = \{q_{i1}, \ldots, q_{in_q}\} \leftarrow \mathrm{LLM}_{\mathrm{idx}}(x_i; n_q). \tag{2}$$

Here, n_q **is a hyperparameter** controlling the number of cue questions per memory. When $n_q = 0$, ROXY reduces to standard dense retrieval without generative indexing. All prompt templates are reported in Appendix A.5.

Indexing strategies. We incorporate cue questions into the index using one of two strategies.
Concat indexing. We concatenate x_i with its cue questions into a single augmented text and embed once, storing one vector per memory chunk:

$$\tilde{x}_i = x_i \oplus \mathrm{Join}(\mathcal{Q}_i), \qquad \mathbf{v}_i^{\mathrm{cat}} = \phi(\tilde{x}_i). \tag{3}$$

Multi-vector indexing. We embed the original memory and each cue question separately, creating

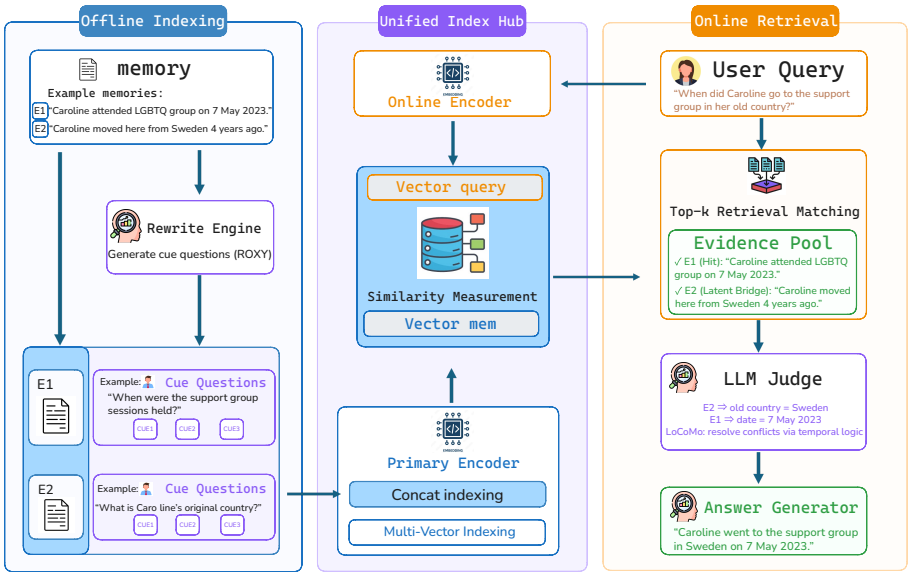

Figure 3: Overview of the ROXY pipeline. **(Left) Offline Generative Indexing**: Each memory chunk b_i is processed to generate n_q anticipatory cue questions $\mathcal{Q}_i$, enabling *query-to-query matching*. These cues are encoded via *Concat Indexing* (single fused vector) or *Multi-Vector Indexing* (separate entry points). **(Center) Unified Vector Index**: Encoded cue vectors are stored for similarity-based retrieval. **(Right) Online Retrieval & Generation**: Dense retrieval returns candidates $\mathcal{B}_q^{(0)}$; a conflict-aware LLM Judge produces $\mathcal{B}_q^{(1)}$; the Answer Generator outputs a memory-grounded response.

multiple entry points that map to the same memory id:

$$\mathbf{v}_{i0} = \phi(x_i), \qquad \mathbf{v}_{ij} = \phi(q_{ij})\ (j = 1, \ldots, n_q). \tag{4}$$

At retrieval time, all vectors $\{\mathbf{v}_{i0}, \mathbf{v}_{i1}, \ldots, \mathbf{v}_{in_q}\}$ vote for the same memory chunk b_i.

Notation for retrieval parameters. We use k_{ret} to denote the number of candidates retrieved in the dense stage (Top-k_{ret}), and k_{ctx} to denote the number of memories finally provided to the answer generator.

3.3 Online Dense Retrieval

At inference time, ROXY encodes the user query as $\mathbf{u} = \phi(q)$ and retrieves candidates by cosine similarity.

Concat retrieval. Each memory has one indexed vector $\mathbf{v}_i^{\text{cat}}$; we retrieve the top-$k_{\text{ret}}$ by similarity.

Multi-vector retrieval. Each memory has multiple vectors $\{\mathbf{v}_{i0}, \ldots, \mathbf{v}_{in_q}\}$ serving as alternative access points. The relevance score is the maximum similarity:

$$S_i(q) = \max_{j \in \{0,\ldots,n_q\}} \mathbf{u}^\top \mathbf{v}_{ij}. \tag{5}$$

We select the top-k_{ret} memories by $S_i(q)$ for reranking.

3.4 Conflict-Aware Reranking and Memory-Grounded Answering

Dense retrieval provides an efficient candidate set but may surface memories that are semantically similar yet logically incompatible (e.g., fragmented vs. complete statements, mutually contradictory evidence, or broken reasoning chains). Instead of injecting hand-crafted evolution rules into the memory store, ROXY resolves such conflicts at retrieval time by reranking candidates with an LLM-based judge.

Reranking score. Let $\mathcal{B}_q^{(0)}$ denote the top-k_{ret} candidates from dense retrieval. An LLM-based reranker assigns each candidate b_i an evidence utility score:

$$r_i = g_{\text{LLM}}\left(q,\ b_i \mid \mathcal{B}_q^{(0)}\right) \in [0, 1]. \tag{6}$$

The conditioning "$\mid \mathcal{B}_q^{(0)}$" allows the reranker to consider all candidates when scoring, enabling suppression of contradictory or irrelevant evidence.

Set-conditioned implementation. All k_{ret} candidates are presented to the LLM *simultaneously*, with instructions to reason about relevance, multi-hop connectivity, and logical consistency. The LLM assigns each candidate an integer score (0–10) normalized to $[0, 1]$, suppressing redundant or

contradictory evidence in a single pass. Although the output format provides a score per item, the scoring process is performed jointly over all candidates within the same context, making it a set-conditioned listwise approach. Prompt templates are in Appendix A.5.

Evidence selection and answer generation. We select the top-k_{ctx} memories by r_i:

$$\mathcal{B}_q^{(1)} = \text{TopK}\Big(\{(b_i, r_i)\}_{b_i \in \mathcal{B}_q^{(0)}},\ k_{\text{ctx}}\Big). \quad (7)$$

The answer generator then produces $\hat{a} \leftarrow \text{LLM}_{\text{gen}}(q, \mathcal{B}_q^{(1)})$.

4 Experiment

4.1 Benchmark and metrics

Benchmark We evaluate ROXY on the LoCoMo benchmark, which targets *very long-term* conversational memory over multi-session dialogues spanning extended periods (Maharana et al., 2024). LoCoMo provides a memory-grounded QA setting where the agent must answer questions by locating and using evidence from a long conversation history. The questions are grouped into five categories—*single-hop*, *multi-hop*, *temporal*, *open-domain*, and *adversarial*—covering direct recall, compositional reasoning across multiple turns, time-aware reasoning, context-dependent conversational understanding, and robustness to unanswerable or misleading queries (Maharana et al., 2024).

Metrics. Following the LoCoMo protocol and the evaluation setup adopted by MemInsight (Salama et al., 2025), we report two complementary metrics. (1) **Retrieval quality** is measured by **Recall@5 accuracy**. Recall@5 is computed as the proportion of gold evidence IDs (session-level following MemInsight, turn-level in ablations) that appear in the top-5 retrieved items. (2) **Answer quality** is measured by token-level **F1** between the generated answer and the reference answer. We report results by question category and overall, matching the table format in MemInsight for direct comparison.

4.2 Baselines and Evaluation Protocol

Experimental Setup and Alignment. To ensure a fair, controlled comparison with MemInsight (Salama et al., 2025), we keep the **embedding model** and **answer generation model** identical. Furthermore, we adopt MemInsight's *session-level* evidence matching criterion for Recall@5: a prediction is correct if the retrieved memories cover the required session IDs. This alignment allows us to isolate the impact of ROXY's Generative Indexing and conflict-aware reranking.

Baselines. We compare against representative LoCoMo-memory methods reported in prior work, including the LoCoMo baseline system (Maharana et al., 2024), long-context reading agents such as ReadAgent (Lee et al., 2024), and long-term memory frameworks such as MemoryBank (Zhong et al., 2024). For embedding-based retrieval, we include the DPR-style RAG baseline (Karpukhin et al., 2020) as a standard dense-retrieval reference point, and we use MemInsight's published LoCoMo results as the main comparison target (Salama et al., 2025).

System configuration. We follow a hybrid deployment consistent with the MemInsight-style evaluation pipeline: (1) embeddings are produced by Amazon Titan Embed Text v2 via AWS Bedrock (Amazon Web Services, 2025); (2) indexing-time question generation and online reranking are performed by a lightweight local LLM served with vLLM (Kwon et al., 2023), instantiated as Qwen3-4B-Instruct-2507 (Qwen Team, 2025); and (3) final answer generation is produced by Claude 3 Sonnet via AWS Bedrock (Anthropic, 2024). Exact prompts and implementation details are deferred to Appendix A.

4.3 Empirical Results

Retrieval analysis (Recall@5). Table 1 compares retrieval performance across embedding-based methods. Under the *concatenation-based* indexing strategy, ROXY with Qwen3-4B-Instruct achieves the best overall retrieval accuracy (**89.3** Recall@5), substantially improving over MemInsight's best reported result (60.5). The *multi-vector-based* variant also performs competitively at 87.8 overall. ROXY sets new state-of-the-art results in **single-hop** (94.7), **temporal** (94.1), and **adversarial** (89.9) categories. In particular, the large lift in **temporal** Recall@5 (94.1 for concat) suggests that GI substantially strengthens time-related retrievability (e.g., "when"-style questions), even without modifying the embedding or generator backbones.

Relevance reranker vs. conflict-aware judge. Interestingly, replacing the Qwen3-4B-Instruct

Model	Single	Multi	Temp.	Open	Adv.	Overall	Avg Rank↓
Embedding-Based Retrieval							
RAG Baseline (DPR)	15.7	31.4	15.4	15.4	34.9	26.5	7.8
MemInsight (Llama $v3_{Pri.}$)	31.3	63.6	23.8	53.4	28.7	44.9	6.2
MemInsight (Mistral $v1_{Pri.}$)	31.4	63.9	26.9	58.1	36.7	48.9	4.5
MemInsight (C3-$Sonnet_{Basic}$)	33.2	67.1	29.5	56.2	35.7	48.8	4.5
MemInsight (C3-$Sonnet_{Pri.}$)	39.7	**75.1**	32.6	**70.9**	49.7	60.5	2.5
Cue-Augmented Retrieval (Ours)							
Concatenation-based							
ROXY$_{concat}$ (Qwen3-4B-Instruct)	**94.7**	74.7	**94.1**	64.6	**89.9**	**89.3**	**1.5**
ROXY$_{concat}$ (Qwen3-4B-Reranker)	93.0	66.2	89.4	62.5	89.7	86.5	3.2
Multi-vector-based							
ROXY$_{multivec}$ (Qwen3-4B-Instruct)	94.0	72.4	93.4	67.3	85.9	87.8	2.0

Table 1: **Retrieval accuracy** (session-level Recall@5) on LoCoMo for embedding-based retrieval. Columns: Single-hop, Multi-hop, Temporal, Open-domain, Adversarial question categories. **Rank**↓: average rank across categories (lower is better). Non-ROXY rows are from MemInsight (Salama et al., 2025). ROXY results use k_{ret}=40 and n_q=3 generated cue questions.

Model	Single	Multi	Temp.	Open	Adv.	Overall	Avg Rank↓
Baseline (Claude-3-Sonnet)	15.0	10.0	3.3	26.0	45.3	26.1	7.0
LoCoMo (Mistral v1)	10.2	12.8	16.1	19.5	17.0	13.9	6.0
ReadAgent (GPT-4o)	9.1	12.6	5.3	9.6	9.8	8.5	9.4
MemoryBank (GPT-4o)	5.0	9.6	5.5	6.6	7.3	6.2	10.4
Attribute-based Retrieval							
MemInsight (Claude-3-Sonnet)	18.0	10.3	7.5	**27.0**	58.3	29.1	4.2
Embedding-Based Retrieval							
RAG Baseline (DPR)	11.9	9.0	6.3	12.0	**89.9**	28.7	6.8
MemInsight (Llama $v3_{Pri.}$)	14.3	13.4	6.0	15.8	82.7	29.7	5.4
MemInsight (Mistral $v1_{Pri.}$)	16.1	14.1	6.1	16.7	81.2	30.0	4.2
MemInsight (C3-$Sonnet_{Basic}$)	14.7	13.8	5.8	15.6	82.1	29.6	5.6
MemInsight (C3-$Sonnet_{Pri.}$)	15.8	15.8	6.7	19.7	75.3	30.1	3.8
Cue-Augmented Retrieval (Ours)							
ROXY$_{concat}$ (Claude-3-Sonnet)	**37.9**	**27.6**	**17.0**	**14.4**	79.2	**41.2**	**2.8**

Table 2: **Answer generation accuracy** (F1 %) on LoCoMo with k_{ctx}=5 retrieved memories. Columns: Single-hop, Multi-hop, Temporal, Open-domain, Adversarial question categories. **Rank**↓: average rank across categories (lower is better). Non-ROXY rows are from MemInsight (Salama et al., 2025).

judge with a dedicated Qwen3-4B-Reranker model—which is explicitly trained for relevance scoring—leads to *lower* retrieval performance (86.5 vs. 89.3 overall Recall@5). The performance drop is most pronounced in **multi-hop** (66.2 vs. 74.7). This result aligns with our design intuition: while a pure relevance reranker excels at query–document matching, it does not inherently handle *conflict resolution*, *temporal updates*, or *evidence set composition*—capabilities that are critical for LoCoMo's multi-session dialogue setting where memories may contradict or supersede each other. This finding further motivates ROXY's use of a **conflict-aware judge** rather than a standard relevance-based reranker for final evidence selection.

Performance analysis (F1). For answer generation evaluation (Table 2), we adopt **ROXY$_{concat}$** with Qwen3-4B-Instruct as the retrieval configuration, since it achieves the highest overall Recall@5 (89.3) and thus provides the most comprehensive evidence for the downstream generator. ROXY achieves the best overall answer accuracy (**41.2** F1), substantially surpassing the strongest MemInsight setting (30.1). The gains are most pronounced in **single-hop** (37.9) and **multi-hop** (27.6), which are precisely the categories where user queries frequently deviate from the original utterance surface form. This pattern is consistent with ROXY's core mechanism: indexing-time cue questions increase the probability that a future query matches at least one stored cue, making relevant evidence more accessible for generation. (See Appendix A.7 for diverse qualitative examples.)

4.4 Ablation Study

Since final answer generation directly depends on retrieving the exact evidence turns, we report **turn-**

Method	Single	Multi	Temp.	Open	Adv.	Overall
w/o GI & Rerank	33.5	70.3	27.5	64.8	47.7	55.6
w/o GI	49.0	82.2	41.2	80.9	63.7	70.9
w/o Rerank	39.2	76.0	34.0	70.5	47.3	60.0
ROXY (Full)	**51.9**	**82.5**	**44.4**	**81.9**	**63.6**	**71.9**

Table 3: Ablation study on the contribution of each component. GI = Generative Indexing (cue question generation). Rerank = LLM-based conflict-aware reranking. All experiments use concat indexing with k_{ret}=40. Turn-level Recall@5 (%) reported.

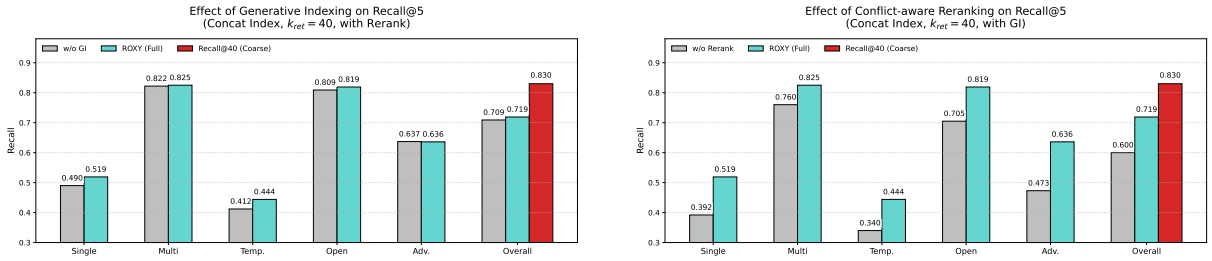

Figure 4: **Left:** Effect of Generative Indexing (GI) on retrieval performance. GI improves Recall@5 by transforming memory representations from statement semantics to query semantics. **Right:** Effect of Conflict-Aware Reranking on retrieval performance. Reranking substantially improves Recall@5 by filtering noisy candidates and resolving contradictory evidence. The coarse candidate set (Recall@40) serves as an upper bound for the final evidence selection (Recall@5).

level Recall@5 in this section—a prediction is correct only if the retrieved turns exactly match the annotated gold evidence. This turn-level metric better reflects each component's contribution to downstream answer quality, as partial or approximate matches do not guarantee sufficient context for accurate generation. We ablate Generative Indexing (GI) and Conflict-Aware Reranking separately (Table 3).

Effect of Reranking. Reranking is the dominant contributor. Removing it ("w/o Rerank") drops overall Recall@5 from 71.9 to 60.0 (**–11.9 points**). The degradation is most severe for **adversarial** (–16.3) and **single-hop** (–12.7) categories, where suppressing misleading or outdated evidence is critical. This confirms that dense retrieval alone retrieves a noisy candidate set; the LLM-based judge is essential for selecting coherent, non-contradictory evidence.

Effect of Generative Indexing. When applied alone (without reranking), GI improves overall Recall@5 from 55.6 to 60.0 (**+4.4 points**), with the largest gains in **temporal** (+6.5) and **single-hop/open-domain** (+5.7). This improvement stems from the core design of GI: by generating cue questions at indexing time, we transform memory representations from *statement semantics* to *query semantics*, enabling **query-to-query matching** rather than query-to-memory matching. Since user queries are naturally phrased as questions, matching against pre-generated questions of similar intent yields stronger semantic overlap than matching against raw conversational utterances.

Component synergy. When combined with reranking, GI still contributes an additional **+1.0 point** (from 70.9 to 71.9), demonstrating that the two mechanisms are *complementary* rather than redundant. GI expands retrieval coverage by injecting future-oriented cues at indexing time, while reranking refines the candidate set by resolving conflicts at retrieval time—together delivering the best overall performance.

Dense retrieval vs. reranked evidence. With dense retrieval depth k_{ret}=40, ROXY reaches Recall@40 of 83.0 overall (Multi-hop 72.9, Temporal 90.4, Single-hop 61.6, Open-domain 91.0, Adversarial 73.3), and reranking selects the final top-5 evidence achieving Recall@5 of 71.9. This gap indicates that most gold evidence is present in the coarse candidate pool. When Recall@40 is high but Recall@5 remains low, the failure is not due to retrieval coverage, but due to the inability to distinguish logically invalid yet semantically similar memories, necessitating conflict-aware evidence selection under a tight context budget.

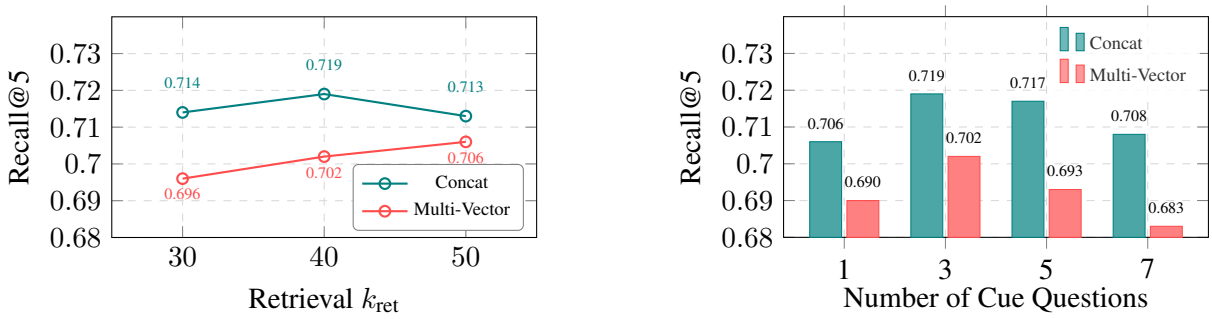

Figure 5: **Left:** Effect of retrieval k_{ret} on turn-level Recall@5. **Right:** Effect of the number of cue questions (n_q) on turn-level Recall@5. All experiments use reranking. Concat indexing outperforms Multi-Vector across settings.

4.5 Hyperparameter Analysis

We analyze two key hyperparameters: the retrieval depth k_{ret} and the number of generated cue questions n_q (Figure 5).

Effect of retrieval k_{ret}. We vary the coarse retrieval depth $k_{\text{ret}} \in \{30, 40, 50\}$ while keeping n_q=3 fixed. Concat indexing peaks at k_{ret}=40 (71.9% Recall@5), while Multi-Vector continues to improve with larger k_{ret} (70.6% at k_{ret}=50), suggesting that its sparser matching pattern benefits from a larger candidate pool. Although Multi-Vector's upward trend implies further gains at $k_{\text{ret}} > 50$, we limit our experiments to $k_{\text{ret}} \leq 50$ due to the memory constraints of the LLM-based reranker: the KV cache grows linearly with the number of candidates, and larger k_{ret} significantly increases GPU memory consumption. In principle, one could rerank the entire memory bank, but in practice this introduces a trade-off between retrieval quality and computational cost.

Effect of cue question count (n_q). We vary the number of generated cue questions $n_q \in \{1, 3, 5, 7\}$ with k_{ret}=40. For concat indexing, performance peaks at n_q=3 (71.9%) and remains stable at n_q=5, then declines at n_q=7 (71.3%). Multi-Vector shows a similar pattern but with sharper degradation at higher n_q. Excessive questions introduce noise that dilutes the core memory semantics, particularly for multi-vector indexing where each question creates an independent retrieval anchor. Based on these results, we adopt k_{ret}=40 and n_q=3 as the default configuration.

5 Conclusion

We presented **ROXY**, a long-horizon conversational memory framework based on **Generative Indexing** and **Conflict-Aware Reranking**. Motivated by the Encoding Specificity Principle (Tulving and Thomson, 1973) (and transfer-appropriate processing (Morris et al., 1977)), ROXY performs *Generative Indexing*: it injects future-oriented retrieval cues at indexing time by generating question-style access points for each memory chunk, instantiated with both cue-fusion (Concat) and cue-separation (Multi-Vector) indexing. At retrieval time, ROXY further uses a *set-conditioned* LLM judge to select a coherent evidence subset, explicitly addressing the practical reality that long-term conversational memories contain updates and contradictions. Evaluated on LoCoMo (Maharana et al., 2024) under the same embedding and generation configuration as MemInsight (Salama et al., 2025), ROXY achieves new best results on both retrieval (Recall@5) and end-to-end answer accuracy (F1), demonstrating that strong gains can be obtained without injecting hand-crafted memory evolution rules into storage. More broadly, ROXY suggests a shift in how we build long-term memory for LLM agents: instead of hard-coding brittle evolution heuristics, we *return conflict resolution to the LLM*—treating contradictions as an evidence selection problem and letting the model's own reasoning capacity arbitrate among competing candidate memories.

Limitations

ROXY's reliance on an LLM-based reranker introduces additional latency, which future work could mitigate through efficiency optimizations of distillation. System inherits the limitations of underlying LLMs; hallucinations in generated cues may negatively impact retrieval precision. Finally, our evaluation is limited to the LoCoMo benchmark; validating generality across broader dialogue distributions and implementing robust privacy controls remain important directions for future deployment.

Ethics Statement

This work involves the development of long-term conversational memory systems, which inherently raises privacy and security concerns. Storing and retrieving user dialogues over extended periods necessitates strict data governance, including encryption, access controls, and mechanisms for users to inspect or delete stored memories ("right to be forgotten"). While our experiments use the public LoCoMo benchmark, real-world deployment requires careful consideration of PII (Personally Identifiable Information) protection. Furthermore, the generative components (cue generation and reranking) rely on LLMs that may exhibit biases; future work must evaluate whether such biases are amplified by the memory selection process. We do not foresee direct misuse potential beyond standard LLM risks, but enhanced long-term persuasion capabilities warrant monitoring to prevent manipulative behaviors.

References

Amazon Web Services. 2025. Amazon titan text embeddings v2. https://docs.aws.amazon.com/bedrock/latest/userguide/titan-embedding-models.html. Accessed 2026-01-02.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. *arXiv preprint arXiv:2010.04898*.

Anthropic. 2024. Claude 3 model family. https://www.anthropic.com/news/claude-3-family. Accessed 2026-01-02.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. *arXiv preprint arXiv:2107.05720*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. *arXiv preprint arXiv:2309.06180*.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv preprint arXiv:2005.11401*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Ali Modarressi. 2025. Nolima: Long-context evaluation beyond literal matching. *arXiv preprint arXiv:2502.05167*.

C. Donald Morris, John D. Bransford, and Jeffrey J. Franks. 1977. Levels of processing versus transfer-appropriate processing. *Journal of Verbal Learning and Verbal Behavior*, 16:519–533.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019a. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019b. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. MemInsight: Autonomous memory augmentation for LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33124–33140, Suzhou, China. Association for Computational Linguistics.

Endel Tulving and Donald M. Thomson. 1973. Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, 80(5):352–373.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. *arXiv preprint arXiv:2005.11723*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. *arXiv preprint arXiv:2105.04166*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Additional Experimental Details

This appendix provides comprehensive implementation details for the **ROXY** memory retrieval and QA pipeline, including baseline configurations, metric definitions, hyperparameter settings, prompt templates, and reproducibility instructions.

A.1 Baseline Details

We compare our approach against the following baselines. All baselines use the same embedding model (Amazon Titan Embed Text v2 (Amazon Web Services, 2025)) and evaluation metrics for fair comparison with Salama et al. (2025).

A.1.1 Model Configuration

A.2 Additional Metrics

A.2.1 Primary Metrics

F1 Score (LoCoMo Official) Following the LoCoMo benchmark (Maharana et al., 2024):

- **Definition:** Token-level F1 with Porter stemming and text normalization.
- **Normalization Steps:**
  1. Remove commas
  2. Remove articles (a, an, the, and)
  3. Fix whitespace
  4. Remove punctuation
  5. Convert to lowercase
- **Implementation:**

```
def f1_score_locomo(prediction, ground_truth):
    prediction_tokens = [ps.stem(w) for w in
     normalize_answer(prediction).split()]
    ground_truth_tokens = [ps.stem(w) for w in
     normalize_answer(ground_truth).split()]
    common = Counter(prediction_tokens) &
     Counter(ground_truth_tokens)
    num_same = sum(common.values())
    precision = num_same / len(prediction_tokens
    )
    recall = num_same / len(ground_truth_tokens)
    f1 = (2 * precision * recall) / (precision +
     recall)
    return f1
```

Category-Specific F1 Logic

- **Category 5 (Adversarial):** If prediction contains "no information available" or "not mentioned", returns 1.0.
- **Category 1 (Multi-hop):** Splits by comma, computes partial F1 across all sub-answers.
- **Categories 2, 3, 4:** Standard token F1.

A.2.2 Secondary Metrics

Recall@K (Proportional)

- **Definition:** Proportion of ground truth evidence IDs found in top-K retrieved results.
- **Formula:** $\text{Recall@K} = \frac{|\text{Retrieved} \cap \text{Evidence}|}{|\text{Evidence}|}$
- **Implementation:**

```
def compute_recall_k(retrieved_ids, relevant_ids,
     k):
    evidence_set = set(relevant_ids)
    retrieved_at_k = set(retrieved_ids[:k])
    hits = len(evidence_set.intersection(
     retrieved_at_k))
    return hits / len(evidence_set)
```

A.2.3 Normalization

- All text is normalized using the LoCoMo official `normalize_answer()` function before metric computation.
- Porter Stemmer is applied to all tokens.

A.3 Extended Results

A.3.1 Question Category Distribution (LoCoMo-10)

A.4 Hyperparameter Settings

A.4.1 Core Pipeline Parameters

```
Model: amazon.titan-embed-text-v2:0
Dimensions: 1024
Normalize: True
Max Input Length: 8000 characters
```

```
Model: Qwen/Qwen3-4B-Instruct-2507
Batch Size: 50 documents per LLM call
Temperature: 0.1 (low for deterministic scoring)
Max Tokens: 1024
Score Range: 0.0 - 1.0
```

```
Model: anthropic.claude-3-sonnet-20240229-v1:0
Temperature: 0.5
Max Tokens: 256
Context Size: Top 5 reranked memories
```

A.4.2 Experimental Grid (Batch Experiments)

A.5 Prompt Templates and Structured Outputs

Baseline	Description	Retrieval Budget	Key Implementation Notes
Raw Embedding (0q)	Direct embedding-based retrieval without query rewriting	Top-K = k_{ret} (default 40; swept in {30,40,50})	Uses original dialogue text only for indexing
Concat Mode	Appends generated questions to original text before embedding	Top-K = k_{ret} (default 40; swept in {30,40,50})	Format: "{original_text} [Potential questions: Q1 | Q2 | ...]"
Multivec Mode	Maintains separate embedding vectors for original text and each generated question	Top-K = k_{ret} (default 40; swept in {30,40,50})	Index size expands with question count
No Rerank	Retrieval-only pipeline, skipping the LLM-based reranking stage	Top-K = k_{ret} (default 40; swept in {30,40,50})	Computes Recall directly on retrieval results
No Rewrite	Skips generative indexing, uses raw embeddings only	Top-K = k_{ret} (default 40; swept in {30,40,50})	Ablation mode: `no_rewrite`
Full Pipeline	Complete pipeline with Rewrite + Concat + Rerank	Top-K = k_{ret} (default 40; swept in {30,40,50}), Final Top-5	Uses Qwen3-4B (Qwen Team, 2025) for reranking

Table 4: Baseline comparisons and implementation details.

Stage	Model	Backend	Notes
Embed	`titan-embed-v2`	Bedrock	1024-dim norm.
Rewrite	`Qwen3-4B-Instr`	vLLM	Generative IDX
Retr.	FAISS	Local	Dense retrieval
Rerank	`Qwen3-4B-Instr`	vLLM	Listwise (set-conditioned) scoring with JSON output
Gen.	`Claude-3-Sonnet`	Bedrock	QA Gen.

Table 5: Model configuration for each stage of the pipeline.

Category ID	Category Name	Count
4	Single-hop	841
1	Multi-hop	282
2	Temporal	321
3	Open-domain	96
5	Adversarial	446
Total	-	**1986**

Table 6: Distribution of question categories in LoCoMo-10 (Maharana et al., 2024).

Parameter	Values
Num Questions	0, 1, 3, 5, 7
Index Method	concat, multivec
Retrieval Top-K	30, 40, 50

Table 7: Hyperparameters for ablation study.

A.5.1 Cue Question Generation (Rewrite) Prompt

```
Generate exactly {num_questions} question(s)
    that users might ask,
where this memory would be the answer.

Memory: {memory_text}

Rules:
- Each question should capture a DIFFERENT
    aspect
- Include both direct facts and logical
    inferences
- Output ONLY the questions, one per line

Questions:
```

Example Input:

```
Memory: [dia_1_2] Alice: I just got back from my
     trip to Japan last week.
```

Example Output:

```
1. When did Alice return from Japan?
2. Where did Alice travel to recently?
3. How long ago was Alice's Japan trip?
```

A.5.2 Retrieval Reranking Prompt

```
Score relevance (0.0-1.0) for each candidate.

Question: {query}

Candidates:
[0] {document_0}
[1] {document_1}
...
[kret-1] {document_last}

Output JSON: {"scores": [{"id": 0, "score": 0.8},
     ...]}
```

Note: Documents are truncated to 1000 characters. Batch size is 50 documents per call. We emphasize that this score is primarily used for conflict-aware judging and selection, rather than merely for sorting candidates.

A.5.3 Answer Generation Prompts

```
Based on the above context, write an answer in
    the form of a short phrase for the following
     question. Answer with exact words from the
    context whenever possible.
Context:
[Memory 1]: {memory_1}
[Memory 2]: {memory_2}
...

Question: {query}
Short answer:
```

```
Based on the above context, write an answer in
     the form of a short phrase for the following
      question. Answer with exact words from the
     context whenever possible.
Context:
[Memory 1]: DATE: {session_date} - {memory_1}
...

Question: {query} Use DATE of CONVERSATION to
     answer with an approximate date.
Short answer:
```

```
Based on the above context, answer the following
      question.
Context:
[Memory 1]: {memory_1}
...

Question: {query} Select the correct answer:
(a) Not mentioned in the conversation (b) {
     ground_truth_answer}.
Short answer:
```

A.5.4 JSON Schema for Reranking Output

```
{
  "$schema": "http://json-schema.org/draft-07/
     schema#",
  "type": "object",
  "properties": {
    "scores": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
            "id": {
            "type": "integer",
            "description": "Index of the
     candidate document (0-indexed)"
            },
            "score": {
            "type": "number",
            "minimum": 0.0,
            "maximum": 1.0,
            "description": "Relevance score"
            }
        },
        "required": ["id", "score"]
      }
    }
  },
  "required": ["scores"]
}
```

A.6 Reproducibility Checklist

A.6.1 Environment Setup

```
# Clone repository
git clone <repository_url>
cd ROXY

# Create environment
conda create -n roxy python=3.10
conda activate roxy

# Install dependencies
pip install -r requirements.txt
```

requirements.txt:

```
boto3>=1.26.0
numpy>=1.21.0
tqdm>=4.62.0
loguru>=0.6.0
python-dotenv>=1.0.0
anthropic>=0.18.0
nltk>=3.8.0
regex>=2023.0.0
faiss-cpu>=1.7.0
pyyaml>=6.0
```

A.6.2 AWS Credentials Configuration

PowerShell (Windows)

```
$env:AWS_ACCESS_KEY_ID="your_access_key"
$env:AWS_SECRET_ACCESS_KEY="your_secret_key"
$env:AWS_DEFAULT_REGION="us-east-1"
```

Bash (Linux/macOS)

```
export AWS_ACCESS_KEY_ID="your_access_key"
export AWS_SECRET_ACCESS_KEY="your_secret_key"
export AWS_DEFAULT_REGION="us-east-1"
```

A.6.3 vLLM Server Setup

```
# Start vLLM via Docker Compose
docker-compose up -d vllm-server

# Verify server is running
curl http://localhost:8000/v1/models
```

A.6.4 Dataset Preprocessing

- Dataset: LoCoMo-10 (`locomo10.json`)
- Location: `../locomo-main/data/locomo10.json`
- No additional preprocessing required.

A.6.5 Running the Full Pipeline

```
# Step 1: Embedding (one-time)
python scripts/run_pipeline.py --phase embed

# Step 2: Rewrite (Generative Indexing)
python scripts/run_pipeline.py --phase rewrite
     --num_questions 3

# Step 3: Retrieval
python scripts/run_pipeline.py --phase retrieve
     \
    --index_method concat --num_questions 3 --
     top_k 40

# Step 4: Reranking
```

Type	Query & Generated Cues	Retrieved Context (Top-3 w/ Rerank)	Answer
Single-Hop	**Q:** Does Calvin want to expand his brand? *Generated Cues:* 1. What is Calvin's current plan for expanding his brand? 2. What specific steps is Calvin taking to grow his global fanbase?	**[D12:11]** Calvin: My plan for now is to expand my brand worldwide and grow my fanbase. I want my music to reach more people... **[D6:11]** Calvin: ...Super excited for my upcoming performance in Tokyo... expand my following!	**Pred:** Yes, "expand my brand worldwide"... **Gold:** Yes
Multi-Hop	**Q:** When did Dave host a card-playing night with his friends? *Generated Cues:* 1. What happened during Dave's card-night with his friends last Friday?	**[D15:1]** Dave: Hey Calvin! ... Last Friday I had a card-night with my friends... **[D10:3]** Dave: ...Just been hanging out with friends at parks lately...	**Pred:** 22 August, 2023 **Gold:** on the Friday before 22 August, 2023
Temporal	**Q:** Does Calvin love music tours? *Generated Cues:* 1. What are Calvin's plans after the tour? 2. What does Calvin expect to achieve...	**[D3:9]** Calvin: ...I'm dreaming of touring the world... **[D30:20]** Calvin: ...accepted an invitation to perform... **[D7:1]** Calvin: ...Touring with Frank Ocean last week was wild.	**Pred:** Yes, "an absolute high!" **Gold:** yes
Open-Dom.	**Q:** According to John, who is his favorite character from Lord of the Rings? *Generated Cues:* 1. What is John's favorite character... 2. How has Aragorn's character development...	**[D27:24]** John: ...My favorite character is Aragorn, he grows so much... **[D27:30]** John: Aragorn's brave, selfless, down-to-earth attitude is what inspired me...	**Pred:** Aragorn **Gold:** Aragorn
Adversarial	**Q:** According to John, who is his least favorite character from Lord of the Rings? *Generated Cues:* 1. What is John's favorite character...	**[D27:30]** ...Aragorn's brave... **[D27:24]** ...My favorite character is Aragorn... **[D1:27]** John: ...Unfortunately, I can't share my love for him with you...	**Pred:** (a) Not mentioned **Gold:** (Empty/Not mentioned)

Table 8: Qualitative case studies from LoCoMo illustrating ROXY's performance across different question types. The **Generated Cues** column shows exemplary questions generated by the Rewrite module that bridge the gap between the user query and the memory content. The **Retrieved Context** shows the most relevant memories surfaced by the system.

```
python scripts/run_pipeline.py --phase rerank \
    --index_method concat --num_questions 3 --
    top_k 40

# Step 5: Generation
python scripts/run_pipeline.py --phase generate
    \
    --index_method concat --num_questions 3 --
    top_k 40

# Step 6: Evaluation
python scripts/run_pipeline.py --phase evaluate
    \
    --index_method concat --num_questions 3 --
    top_k 40
```

A.6.6 Batch Experiments

```
# Run all ablation experiments (Num Questions x
    Index Method x Top-K)
python scripts/run_batch_experiments.py
```

This will iterate over:

- **Num Questions:** 0, 1, 3, 5, 7
- **Index Method:** concat, multivec
- **Top-K:** 30, 40, 50

Results are saved to `ablation_results_bedrock.csv`.

A.6.7 Evaluation Scripts

```
# Calculate Generation F1 (per-category
    breakdown)
python scripts/calc_generation_f1.py

# Calculate Recall by Category
python scripts/eval_recall_by_category.py

# Calculate Session-level Recall (aligned with
    MemInsight)
python scripts/calc_session_recall.py
```

A.6.8 Cache and Output Structure

```
output/
 cache/
    embeddings.json          # All memory
     embeddings
    rewrite_3q.json          # Rewritten
     questions
```

```
    retrieval_top40_3q_concat.json
    rerank_top40_3q_concat.json
    generation_top40_3q_concat.json
eval_results_3q_concat_top40.json
ablation_results_bedrock.csv
```

A.6.9 Hardware Requirements

- **GPU:** NVIDIA GPU with $\geq$16GB VRAM (for vLLM with Qwen3-4B (Qwen Team, 2025))
- **RAM:** $\geq$32GB recommended for large batch experiments
- **Storage:** $\sim$5GB for cache files and embeddings

A.7 Qualitative Analysis

We provide qualitative examples from the LoCoMo dataset to illustrate how ROXY's Generative Indexing and Conflict-Aware Reranking operate across different question categories. Table 8 shows detailed traces for Single-hop, Multi-hop, Temporal, Open-domain, and Adversarial queries. In particular, note how Generative Indexing creates specific questions (e.g., "When did Alice return from Japan?") that bridge the lexical gap between the user query and the declarative memory text. For the **Adversarial** case (Category 5), the retrieval correctly surfaces related but non-answerable context (e.g., favorite character is Aragorn), and the Conflict-Aware Judge correctly determines that the specific query (least favorite character) is not mentioned.