Exploratory and Assimilating Reflection: Reflective Recall Cycle for Long-term Memory

Ganesh Senrayan [1] **Moyuru Yamada** [1] **Ishan Jindal** [1] **Kiran Purohit** [1]

Abstract

LLM-based autonomous agents require external memory to overcome their statelessness and limited context window for long-term interaction and dynamic knowledge reasoning. However, existing memory retrieval methods often lack adaptability and sample efficiency, and struggle to retrieve the right mixture of memories from heterogeneous stores. We propose *Exploratory-Assimilating Reflection (EAR)*, a framework for high initial retrieval performance and sample-efficient adaptation. EAR combines two mechanisms: Exploratory Reflection, which performs iterative search to bootstrap retrieval and collect useful experiences for each query, and Assimilating Reflection, which replays these experiences from an Experience Buffer to refine a global reranker more efficiently than methods relying only on immediate rewards. Experiments show that EAR improves retrieval by up to 17.9% over the baseline retriever on two long-term dialogue benchmarks. We also show that EAR is highly sample-efficient and robust to noisy feedback.

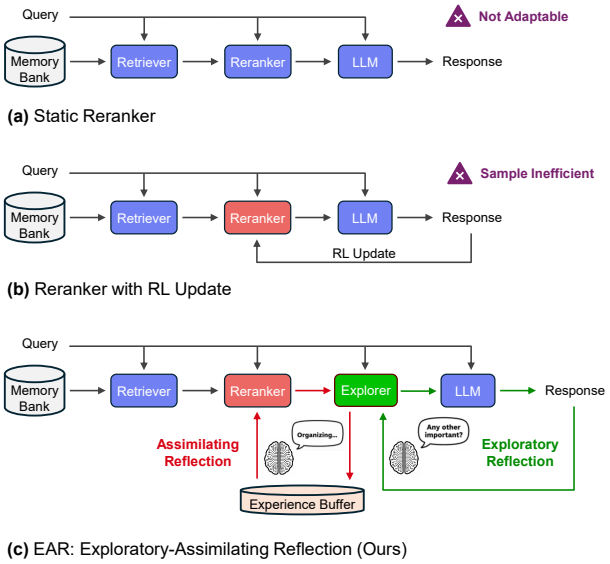

Figure 1. Comparison of memory retrieval paradigms. (a) Static reranker is unable to adapt to data distribution shifts. (b) Reranker with RL update struggles with sample inefficiency. (c) Our EAR (Exploratory-Assimilating Reflection) achieves both high initial recall and sample-efficient updates.

1. Introduction

Autonomous agents powered by Large Language Models (LLMs) hold immense potential for automating complex, real-world tasks (Mialon et al., 2023). However, the underlying architecture of LLMs is inherently stateless (Tan et al., 2025); their inability to consider context beyond a finite window makes it challenging to maintain long-term memory for personalized interaction or to reason over dynamically evolving knowledge bases. Thus, an external memory mechanism is crucial for the agents to overcome this fundamental limitation.

Surfacing the essential information for a query from a vast memory bank is a deceptively challenging task. The most standard approach employs a static retriever, pre-trained on large-scale data, to identify relevant memory chunks based on semantic similarity in an embedding space (Lewis et al., 2020). It relies heavily on the pre-trained retriever, which leads to significant performance degradation when applied to out-of-domain tasks (Thakur et al., 2021). A common approach to mitigate this limitation is to introduce a reranker that is tuned offline (Yu et al., 2024). However, this offline-tuning approach, has two major drawbacks. It demands expensive data collection and fails to adapt to shifts in data distribution over time (Gründel et al., 2024; Alkhalifa et al., 2024). (Fig. 1 (a)).

To adapt to domain shift, Tan et al. (2025) have introduced online adaptation method, which employs Reinforcement Learning (RL) to learn from LLM feedback during interactions and adapt to evolving domains and user patterns (Fig. 1 (b)). While this approach marks a significant step towards more adaptive and robust agents, the slow adaptation and instability inherent in its RL update makes it sample

[1]Fujitsu Research. Correspondence to: Ganesh Senrayan <ganesh.senrayan@fujitsu.com>.

inefficient. This requires the agent to undergo a substantial number of interactions to improve, leaving it vulnerable to poor performance during the crucial initial stages.

To address the aforementioned limitations, we propose **Exploratory-Assimilating Reflection (EAR)**, a novel framework for adaptive memory retrieval. For a given query, Exploratory Reflection performs an active, iterative search for the most relevant memory entries using an Explorer module. This search not only constructs a rich set of evidences for the given query but also produces valuable experiences for long-term adaptation. Assimilating Reflection is designed to achieve the long-term adaptation of the global adapter module. Unlike (Tan et al., 2025) that learns myopically from immediate rewards, Assimilating Reflection utilizes an Experience Buffer to store a history of exploratory experiences. By replaying and learning from this diverse set of past experiences, we efficiently and continuously update the reranker model, improving its overall performance over time.

Our framework is inspired by cognitive development. In humans, short-term memory (or working memory) maintains a limited amount of information for immediate use and long-term memory consolidates experiences over extended timescales (Baddeley, 2003; Squire, 2004). Cognitive theories suggest that rehearsal or replay enables transfer from short-term to long-term storage, and that long-term traces can be reactivated when needed (Danker & Anderson, 2010). Inspired by this interplay, EAR unifies query-specific evidence construction with continual model adaptation.

Extensive experiments on the LongMemEval and LoCoMo benchmarks demonstrate that our approach achieves significant improvements in recall, with absolute gains of up to 17.9% and 12.5% over the baseline retriever. Furthermore, EAR exhibits exceptional sample efficiency and remains highly robust to noisy feedback, validating the efficacy of its core components. To test EAR's capability to adapt to a heterogeneous memory environment, we setup a case study on the LoCoMo dataset where a mixture of dialogue (*episodic*) and observational (*semantic*) evidence is available. We demonstrate that EAR consistently selects a more effective, balance of memory types. This balancing improves downstream answering accuracy by to 13.6%, proving that EAR's adaptivity seamlessly extends to mixing distinct memory stores.

Our contributions are summarized as follows:

- We propose Exploratory-Assimilating Reflection (EAR), a novel framework for adaptive memory retrieval that combines iterative selection (Explorer) with long-term adaptation from past interactions (Experience Buffer).
- We highlight EAR's superior sample efficiency, demonstrating that it reaches static baseline performance in 77% fewer feedback steps than existing RL-based retrieval method.
- We show that EAR seamlessly balances heterogeneous memory sources. When presented with multiple memory stores, it adaptively retrieves an optimal, mixture of evidence.

2. Related Work

Our framework aims to equip LLM-based agents with a robust, continually adapting memory system. To contextualize our contributions, we study three interconnected areas of prior work: (1) LLM-based autonomous agents, (2) standard retrieval-augmented reasoning pipelines used to overcome their inherent statelessness, and (3) recent adaptive memory retrieval methods, where we identify the critical sample-efficiency gaps addressed by EAR.

LLM-based Autonomous Agents The advent of powerful Large Language Models (LLMs) such as GPT-4 (Achiam et al., 2023) and Llama 3 (Dubey et al., 2024) has led to the development of sophisticated autonomous agents. These agents aim to perform complex, multi-step tasks by leveraging the reasoning and planning capabilities of LLMs. Seminal works have explored various agent architectures, including those that use tools (Schick et al., 2023), generate task-specific plans (Yao et al., 2022), or engage in self-reflection to improve their performance (Shinn et al., 2023). However, a fundamental limitation of these agents is their reliance on the LLM's finite context window and static parametric knowledge. To this end, several architectures have been proposed to explicitly manage long-term memory (Zhong et al., 2024; Li et al., 2025), which are designed to maintain conversational consistency over extended interactions. Thus, the mechanism by which an agent retrieves information from its memory is a critical component that directly governs efficacy and intelligence.

Retrieval-Augmented Reasoning Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), has become a standard for supplementing LLMs with external knowledge. To improve the retrieval performance, subsequent works use a efficient retrievers (sparse retriever like BM25 (Robertson & Zaragoza, 2009), dense retrievers like Contriever (Izacard et al., 2022)) to fetch candidate documents followed by a powerful reranker (Yu et al., 2024).

While a finetuned reranker is effective for static knowledge bases, it faces significant challenges when applied to the long-term memory of an autonomous agent. Furthermore, a memory retrieval mechanism for autonomous agents must be robust and adaptable to the diverse retrieval demands of

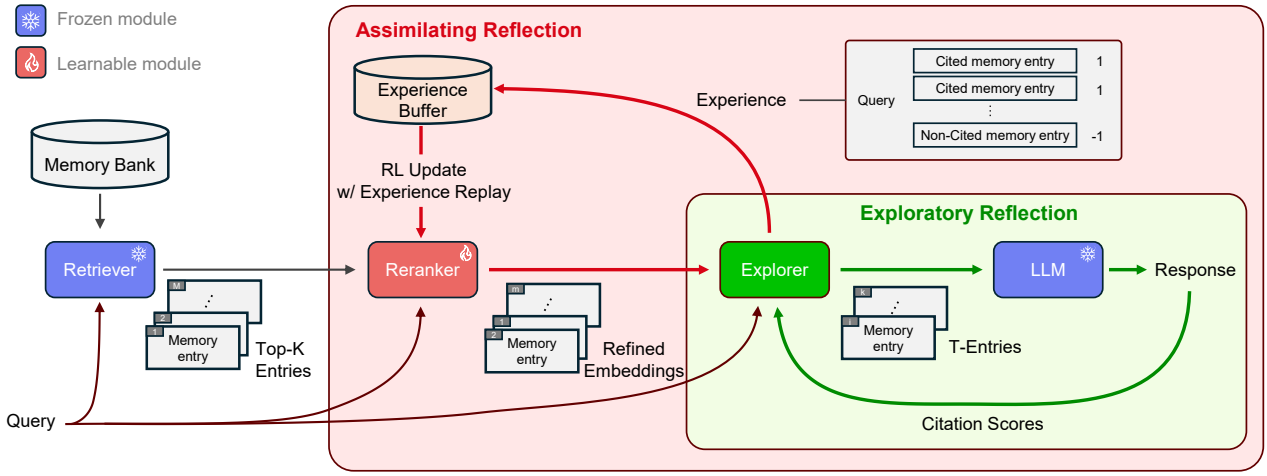

Figure 2. Overview of Exploratory-Assimilating Reflection (EAR) Framework. Exploratory Reflection performs an iterative deep search within a single query to build a rich set of useful context using Explorer, thereby bootstrapping the learning process of the adaptive reranker. Assimilating Reflection takes a broad view across a history of these explorations stored in Experience Buffer, efficiently improve the global Reranker model.

varying conversation domains and individual user interaction patterns. Methods trained on a fixed corpus struggle to handle this dynamic nature, necessitating adaptive methods.

Adaptive Memory Retrieval More recent and sophisticated approaches aim to make the retrieval process itself more dynamic and intelligent at inference time. For instance, Self-RAG (Asai et al., 2024) fine-tunes an LLM to generate special "reflection tokens" that control the retrieval process. Similarly, approaches like Adaptive-RAG (Jeong et al., 2024) learn a policy to adapt the retrieval strategy based on the query's complexity. While these methods make the retrieval decision process highly adaptive, the control policies themselves are typically trained offline on a fixed dataset. They are not designed to continually update based on the feedback from ongoing agent interactions. The need for robust and adaptive retrieval is particularly important in agents designed for long-term interaction, such as Memory-Bank (Zhong et al., 2024) and LD-Agent (Li et al., 2025). To address the need for continual learning, the most recent method has proposed a method to update the reranker based on feedback signals from the LLM through REINFORCE algorithm (Tan et al., 2025). This method marks a significant step towards more adaptive agents. However, it suffers from critical sample inefficiency, requiring a substantial number of interactions to learn effectively.

3. Methodology

Our proposed framework shown in Figure 2 is composed of two distinct modules: **Exploratory Reflection** acts as a short-term memory, performing responsive, query-specific refinement. **Assimilating Reflection** serves as a long-term memory, integrating feedback across queries to adapt the global embedding space. Synergistically, they form a reflection cycle that balances immediate adaptivity and cumulative learning, rendering the process more sample-efficient and cognitively plausible.

3.1. Problem Setting

We consider the task of adaptive memory retrieval, which requires both extracting relevant memories for a given query and learning from the outcomes of that extraction. Formally, given a query q and a memory bank $\mathcal{M}$, the system must initially select a subset $\mathcal{S} \subseteq \mathcal{M}$ of size s to provide sufficient context for accurately answering q. Crucially, once $\mathcal{S}$ is selected, a reward is observed based on the relevance of these memories to q. The core problem is to leverage this available reward signal to update the retrieval mechanism, optimizing memory selection for subsequent queries. This setting differs from traditional static methods, framing retrieval as a dynamic reward-driven learning process.

3.2. EAR Framework

The Exploratory-Assimilating Reflection (EAR) framework consists of two synergistic components namely, **Exploratory Reflection** and **Assimilating Reflection**. The overall two-step process is summarized in Algorithm 1, with detailed component-level algorithms in **Appendix A**.

3.2.1. Exploratory Reflection

Exploratory Reflection serves as the short-term retrieval

Algorithm 1 Exploratory-Assimilating Reflection (EAR)

Input: Query q, memory bank $\mathcal{M}$, exploration steps T
Output: Final memory subset $\mathcal{S}_T$

$\mathcal{M}_k \leftarrow \text{Retriever}_\theta(q, \mathcal{M})$ *// Retrieve top-k*
$q', \mathcal{M}'_k \leftarrow \text{Adapter}_\phi(q, \mathcal{M}_k)$ *// Embed and adapt*
for $t = 1$ to T **do**
  $\mathcal{S}_t \leftarrow \text{BanditSelect}(q', \mathcal{M}'_k)$ *//Exploratory Reflection*
  $\mathbf{y}_t \leftarrow \text{LLMFeedback}(\mathcal{S}_t)$ *// $\mathbf{y}_t \in \{-1, +1\}^{|\mathcal{S}_t|}$*
  $\text{UpdateUCBStats}(\mathbf{y}_t)$
end for
$\bar{\mathbf{y}} \leftarrow \frac{1}{T}\sum_{t=1}^{T} \mathbf{y}_t$ *// Aggregate rewards*
$\mathcal{B} \leftarrow \mathcal{B} \cup \{(q, \mathcal{M}, \bar{\mathbf{y}})\}$ *// Experience Storage*
$b \sim \mathcal{B}$ *// Sample replay mini-batch*
// Assimilating Reflection
$\phi \leftarrow \phi - \eta \nabla_\phi \big(\mathcal{L}_{curr}(\phi) + \mathcal{L}_{replay}(\phi; b)\big)$
return $\mathcal{S}_T$

module of EAR. Given a query embedding q and a candidate memory pool $\mathcal{M}_k = \{m_1, \ldots, m_k\}$, its objective is to incrementally construct an effective subset $\mathcal{S}$ for downstream question-answering. Crucially, rather than relying on static selection, EAR must balance exploiting known semantic relevance with exploring uncertain memories to gather rich feedback for subsequent adaptation. To formalize this exploration-exploitation trade-off, we model iterative retrieval as a multi-armed bandit problem.

At each step, we pick the memory with the highest score calculated from Equation 1 without replacement. We use the following score that incorporates semantic similarity, subset diversity (via Maximal Marginal Relevance (Carbonell & Goldstein, 1998)), and an exploration bonus.

$$\begin{aligned}\text{score}(m_i|q, \mathcal{S}) &= w_s \,\text{Sim}(q, m_i) \\ &\quad - w_d \max_{m_j \in \mathcal{S}} \text{Sim}(m_i, m_j) + w_c \,\text{UCB}_i, \end{aligned} \tag{1}$$

where $\text{Sim}(\cdot, \cdot)$ denotes cosine similarity, and UCB_i is the upper-confidence estimate for memory m_i. Hyperparameters w_s, w_d, w_c balance the individual components of the score. After the subset is constructed, an LLM judges each selected memory as *useful* $(+1)$ or *not useful* (-1) for answering the query. Let $y_i \in \{-1, +1\}$ denote the LLM feedback for each selected memory $m_i \in \mathcal{S}$. The per-memory labels $\{y_i\}$ are used to update the bandit statistics (per-arm mean reward, number of arm pulls, etc.). This exploration loop is continued for a fixed number of T steps. **Computational Cost.** It is to be noted that enabling the explorer module results in a cost of $T - 1$ additional LLM calls per query.

3.2.2. Assimilating Reflection

While Exploratory Reflection improves retrieval for the current query, Assimilating Reflection converts this query-specific feedback into persistent improvements of the global adapter (reranker). Its goal is to reshape the embedding space so that memories that are found useful for answering a query, move closer to said query.

Following prior work (Tan et al., 2025), we apply lightweight residual linear adapters to both query q and memory embeddings $\mathcal{M}_k$ (top-k retrieved by the retriever):

$$q' = q + W_q q, \qquad m'_i = m_i + W_m m_i, \tag{2}$$

where, $W_q, W_m \in \mathbb{R}^{d \times d}$ are trainable parameters (where d is the embedding dimension). As discussed in Section 3.2.1, Exploratory Reflection selects a subset of memories $\mathcal{S} \subseteq \mathcal{M}_k$ (using the modified query and memory embeddings provided by the adapter) and obtains LLM feedback for each selected memory (denoted by y_i).

We then update the adapter by increasing the log-probability of positively rewarded memories and decreasing that of negatively rewarded memories:

$$\mathcal{L} = -\frac{1}{|\mathcal{S}|} \sum_{m_i \in \mathcal{S}} (y_i - b) \,\log \pi_\phi(i \mid q, \mathcal{M}_k) \tag{3}$$

where, b is a baseline used for variance reduction. To calculate log probabilities, we use:

$$z_i = \text{Sim}(q', m'_i); \pi_\phi(i \mid q, \mathcal{M}_k) = \text{Softmax}(z_i) \tag{4}$$

Intuitively, this encourages the adapter to assign higher probability to memories that are repeatedly validated by the LLM, thereby improving retrieval quality even before future exploration.

3.2.3. Experience Buffer

To improve sample efficiency, we maintain an *Experience Buffer* storing past query-level experiences. For each query, we aggregate the observed feedback into a reward vector $\bar{\mathbf{y}} \in \mathbb{R}^k$, where $\bar{y}_i$ denotes the average reward assigned to memory m_i across exploratory trials in which it was selected. The tuple $(q, \mathcal{M}_k, \bar{\mathbf{y}})$ is stored in the buffer and replayed during training. For a new query, past queries are sampled as replayed experience from the replay buffer. Sampling is based on cosine similarity to the current query (using pre-adapter embeddings). For a replayed experience, we recompute probabilities in Equation 4 under the current adapter and optimize the same reward-weighted log-probability objective using $\bar{\mathbf{y}}$ as supervision. The final training loss combines the current-query loss with the replay loss. This replay mechanism allows adapter to consolidate reward signals across queries, making learning substantially more sample-efficient than updating only from immediate feedback.

4. Experiments

4.1. Implementation Details

We equip the EAR framework with the following three retrievers, namely, Contriever (Izacard et al., 2022), Stella

(Zhang et al., 2025) and GTE (Li et al., 2023). These retrievers have strong semantic representation capabilities and are widely used in personalized dialogue systems For generating answers, we use GPT 5 Mini (Singh et al., 2025). For the explorer, we use $T = 4$ exploration steps (this choice is explained in **Section 4.6**).

4.2. Datasets Used

We evaluate our framework on LONGMEMEVAL (Wu et al., 2025) and LOCOMO (Maharana et al., 2024). Both of these operate in a chat-based environment, well equipped for evaluating long-term memory capabilities of agents. In both the benchmarks, we use dialogues as memory candidates.
LongMemEval: We use the $\text{LONGMEMEVAL}_\text{S}$ version of the dataset that contains 500 questions, each tied to user-assistant chat histories. It spans more than 199k dialogue turns across 19k sessions.
LoCoMo: We use LOCOMO containing 10 conversations. The 10 conversations span 5882 turns of dialogue across 272 sessions. It encompasses 1540 queries divided into 4 distinct categories such as Single-hop , Multi-hop, Temporal and Open-Domain (we ignore Adversarial queries due to lack of availability of gold-memories).

4.3. Baselines

To evaluate the effectiveness of EAR, we compare it against the following baselines.
Only Retriever: We use retriever to select the top-s memories $\mathcal{S}$ based on semantic similarity alone.
Reranker + RL update: We use the Retrospective Reflection (Reranker + RL update) proposed by (Tan et al., 2025).

4.4. Training and Evaluation

We conduct our experiments in two phases. In the training phase, we use 80% queries from the dataset where the adapter is continually trained. The other 20% queries are used for evaluation. To evaluate the efficiency of our retrieval mechanism, we restrict the target memory subset size to $s = 5$. This strict capacity constraint establishes a rigorous baseline, demonstrating that EAR can successfully surface critical evidence even when operating under a heavily restricted context budget. We use *Recall@5* (all gold memories in top-5) and *Hitrate@5* (any gold memory in top-5; (IBM, 2025)) to evaluate retrieval quality. *F1* and *LLM-As-a-Judge* (denoted by *J*) measure question-answering accuracy. We employ turn-level granularity.

Citation Scores. To ensure reproducible evaluation devoid of LLM variance, we use a probabilistic reward simulator calibrated to real-world LLM precision/recall reported by (Tan et al., 2025) (Precision 88.0, Recall 86.0. See **Appendix A.3**) This also allows for systematic analysis of impact of feedback noise.

4.5. Results and Analysis

This section first demonstrates overall performance of our EAR framework over the baselines on LongMemEval benchmark. We then conduct various analysis including the sample efficiency and feedback-noise robustness.

4.5.1. PEFORMANCE COMPARISON

On both the benchmarks, we show that our method consistently outperforms the baselines (Table 1). On the LongMemEval dataset, our EAR framework achieves significant improvement in retrieval performance, showing an absolute **increase in Recall@5 of 20.6%** over the existing RL update algorithm and **17.9%** over the baseline retriever with Contriever. This is consistent with the LoCoMo dataset too (**29.1%** over baseline RL update and **12.5%** over baseline retriever). The gains in retrieval quality translate to downstream question-answering accuracy, improving **LLM-As-a-Judge score by 10.4%** (LongMemEval) and **9.0%** (LoCoMo) over baseline retriever (Contriever); and **15.8%** (LongMemEval) and **24.4%** (LoCoMo) over baseline RL update. The baseline RL update notably leads to a performance drop below the static retriever. We discuss this in next section.

With more training queries in LoCoMo, we see that EAR with larger retrievers like Stella and GTE outperform EAR with Contriever. But with fewer training queries in LongMemEval, EAR with Contriever outperforms larger retrievers. This suggests that under tight sample constraints EAR gives the largest improvement for smaller retrievers such as Contriever. This is likely due to number of samples required for training a larger adapter (size scales as $2 \times d_{embed}^2$ for more details, see **Appendix B.3**).

EAR is model-agnostic and plugs into standard retrieval pipelines. To test compatibility, we layer EAR on top of a fine-tuned text reranker and observe consistent gains over the reranker alone (details and full tables in **Appendix B.4**). This indicates EAR complements strong non-RL rerankers rather than merely replacing them.

4.5.2. SAMPLE EFFICIENCY

Figure 3 compares the sample efficiency of EAR against the conventional RL update (Tan et al., 2025) on LongMemEval. We disable the explorer at test time to isolate the learned adapter's performance. Comparing efficiency by the number of unique training queries observed, EAR achieves the strongest performance throughout the learning trajectory. While baseline RL update drops 10% below the static retriever initially, **EAR limits this to 3%** and **surpasses the static retriever after only 80 training queries**.

		LongMemEval				LoCoMo			
Method	**Retriever**	**Recall@5**	**Hitrate@5**	**F1**	**J**	**Recall@5**	**Hitrate@5**	**F1**	**J**
Only Retriever	Contriever	62.37	86.02	36.12	54.84	42.60	52.01	25.23	45.45
	Stella	66.67	90.32	38.23	54.84	54.22	63.31	29.42	57.02
	GTE	68.89	93.55	40.98	60.22	57.59	71.42	33.38	62.72
Reranker + RL Update	Contriever	59.71	82.08	35.06	49.46	25.97	34.31	17.11	30.08
	Stella	54.80	80.65	33.74	50.90	51.95	66.53	31.43	58.12
	GTE	33.11	68.10	25.44	37.99	39.31	52.27	27.87	49.89
EAR (Ours)	Contriever	**80.28**	**94.98**	**41.34**	**65.23**	55.06	70.83	29.85	54.44
	Stella	73.47	93.55	40.49	62.37	**63.73**	**79.61**	34.10	63.09
	GTE	62.90	91.76	38.41	58.42	59.74	77.81	**35.02**	**64.94**

Table 1. Retrieval and Question-Answering performance on LongMemEval and LoCoMo. **Recall@5** and **Hitrate@5** denote retrieval quality metrics. **F1** denotes F1-Score between LLM-generated answer and gold answer. **J** denotes the question-answering accuracy using LLM-as-a-Judge. The results are averaged over 3 runs.

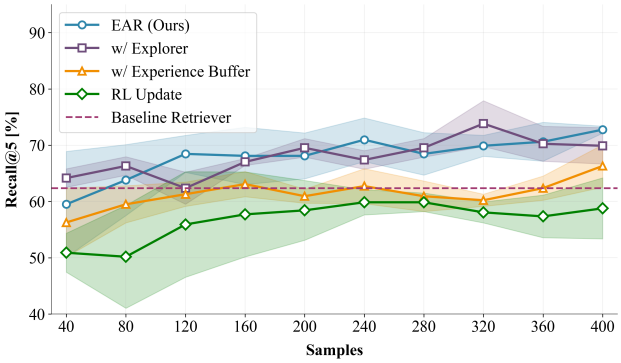

Figure 3. Sample efficiency comparison on LongMemEval (Explorer is disabled at test time). Each curve represents Mean ± Std for 3 runs.

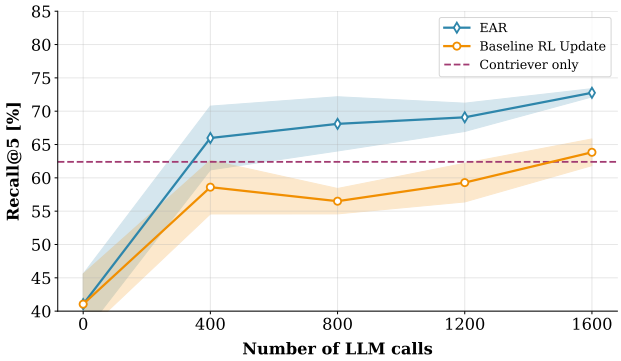

Figure 4. Comparison with Baseline RL update under same computational budget (Explorer is disabled at test time). Each curve represents Mean ± Std for 3 runs.

In our low-resource continual adaptation setting (400 unique queries), the conventional RL method underperforms the static baseline due to sample inefficiency. To ensure a fair comparison of computational overhead (Figure 4), we also evaluate performance normalized by the total number of LLM calls. When the oracle budget is matched ($400 \times T$ Exploration Steps $= 1600$ LLM calls; for baseline this was done by running the 400 query-train set for 4 epochs), the baseline RL update overtakes the static retriever (after 1400 LLM calls vs EAR's 320). But even when controlling for this feedback budget, EAR maintains a stable lead.

Finally, Figure 3 shows this improvement arises from EAR's complementary components. The Explorer mitigates mainly the severe cold-start problem, stabilizing learning under limited samples. The Experience Buffer then accelerates the overall learning trajectory, enabling the model to quickly outperform the baseline. Together, they resolve the key weaknesses of poor initial performance and inefficient learning of the prior methods.

4.5.3. Impact of Feedback Quality

As discussed in Section 4.4, we simulate the LLM's citation score using a probabilistic model. Real-world LLMs are prone to distinct biases and hallucinations which manifest as degraded precision and recall in citation scoring. To investigate the effect of feedback quality, we conduct experiments with varying precision and recall. Figure 5 indicates how lower feedback quality from the LLM affects the adapter's update process. By averaging the feedback from multiple trials generated by the Explorer, EAR stays resilient to high feedback noise, maintaining a stable learning curve. Even in highly noisy conditions (Precision = 70, Recall = 70) EAR outperforms baseline retriever. To test validity of simulations, we test with real LLM citation scores and found the performance was close to simulations. (See **Appendix B.2**)

4.5.4. Ablation Study

We conduct an ablation study to isolate the contribution of each component in EAR. Without Explorer at test time, adding either exploration-based training or the Experience Buffer independently already improves retrieval, showing that both components help train a stronger adapter. Combining them yields the best performance, indicating that they contribute complementary training signals. This also shows that EAR allows the costly exploration module to be discarded at test time, achieving high retrieval accuracy purely through the updated lightweight adapter. When comparing

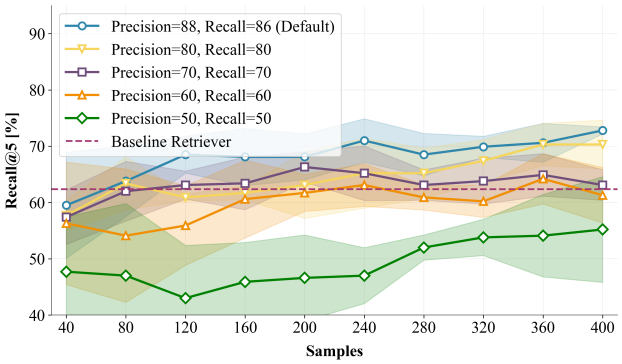

Figure 5. EAR's robustness against feedback noise (Explorer is disabled at test time). Curves represent Mean ± Std for 3 runs.

Variant	LongMemEval Recall@5	LoCoMo Recall@5
Without Explorer at test time		
Adapter (Tan et al., 2025)	54.12	25.97
+ Only Explorer during training	69.91	39.99
+ Only Experience Buffer during training	66.32	29.54
EAR w/o Explorer at test time	72.80	43.56
With Explorer at test time		
Only Explorer at test time	68.81	41.20
EAR (Ours)	**80.28**	**55.06**

Table 2. **Ablation study.** The top block reports retrieval performance without using Explorer at test time, isolating the effect of training-time components. The bottom block reports when Explorer is enabled at test time.

against test-time latency introduced by each method, we see that disabling explorer at test time gives the best balance of latency and performance (see **Appendix B.1**).

With Explorer at test time, using Explorer alone without the learned adapter performs worse than using both together. This confirms that EAR performs best when the adapter and Explorer operate synergistically.

4.6. Impact of Hyperparameters

We test our EAR framework on the LongMemEval dataset with different hyper-parameters. The results in Table 3 suggest a trade-off regarding the number of exploration steps. An insufficient number of steps risks overlooking valuable feedback, whereas an excessive number may increase the vulnerability to noise within the feedback signals. Similarly, our experiments on the batch size used in Experience Replay indicate that an optimal balance must be struck between prioritizing immediate feedback from the current query and assimilating broader lessons from past experiences stored in the buffer.

4.7. Impact of Sampling Strategy

We test three sampling strategies for Experience Replay. *Similarity* strategy retrieves experiences with top-n most similar queries. *Reward* strategy uses bottom-n experiences by reward. Finally, *Random* strategy uniformly samples n experiences from the buffer. The results in Table 4 show retrieval quality with different sampling strategies indicate that leveraging past experiences from similar queries is the most effective strategy. We hypothesize that this is because these relevant experiences help to guide the reranker's update direction more effectively.

# of Exploration	Replay Batch Size	Recall@5
2	4	68.5
4	4	**72.8**
8	4	69.5
4	2	71.3
4	8	71.0

Table 3. Impact of the number of exploration and the size of experience buffer. Recall@5 shows recall rate of updated reranker in percentage (%) (i.e., Explorer is disabled at test time).

Sampling Strategy	Recall@5
Similarity	**72.8**
Reward	69.9
Random	70.3

Table 4. Impact of sampling strategies of Experience Replay. Recall@5 shows recall rate of updated reranker in percentage (%) (i.e., Explorer is disabled at test time).

5. Memory Selection from Multiple Stores

Modern agentic memory systems often have *multiple stores of memory* associated with them, namely, *episodic, semantic* etc. (Wang & Chen, 2025) Every query requires a different proportion of these memory stores to be adequately answered. In this section we demonstrate EAR's capability to extract the relevant query-dependent mixture of memories.

5.1. Experimental Setup

We use the LoCoMo dataset, treating *dialogues as episodic memory* and session-wise *observations as semantic memory*. To isolate the effect of memory mixing *after* retrieval, we enforce a balanced candidate set at the retriever stage: instead of retrieving the top-k memories jointly from the union of both stores, the retriever independently fetches the top-$\frac{k}{2}$ dialogues and the top-$\frac{k}{2}$ observations. The resulting combined set of size k is then passed to the *EAR* framework, from which the final top-s memories are selected ($k = 20$, $s = 5$). This design prevents the initial candidate pool from being dominated by either observations or dialogues. For the *Retriever Only* baseline, the final top-s memories are selected directly from this same combined candidate set. We use Contriever as the base retriever.

Method	Type	F1	J
Retriever Only	Only Dialogues	25.23	45.45
	Only Observations	23.77	34.74
	Dialogue + Observations	28.88	47.73
EAR	Only Dialogues	31.27	54.22
	Only Observations	25.83	39.94
	Dialogue + Observations	**34.40**	**59.09**

Table 5. Results across different input types in LoCoMo dataset. **J denotes accuracy with LLM-as-a-Judge**

Method	Type	Recall@5 Dialogue	Recall@5 Observations	J
Retriever Only	Single Hop	32.95	63.58	61.85
	Multi Hop	1.89	16.98	20.75
	Temporal	26.56	60.94	37.50
	Open Domain	5.56	22.22	27.78
EAR	Single Hop	54.33	62.43	73.99
	Multi Hop	7.55	16.67	30.19
	Temporal	46.88	53.13	51.56
	Open Domain	16.67	22.22	27.78

Table 6. Results across query types in LoCoMo dataset. Both dialogues and observations are retrieved. **J denotes the accuracy with LLM-as-a-Judge**

5.2. Results and Analysis

This section demonstrates how EAR performs against baseline retriever when presented with multiple memory types (episodic and semantic). From Table 5, we observe that providing both dialogues and observations in context benefits both methods (EAR and Retriever only). However, EAR makes substantially better use of this mixed memory pool. Figure 6 shows that *Retriever only* selects mostly observations in top-s, likely because observations contain denser semantic information and achieve higher similarity scores with the query. **In contrast, EAR uses feedback to adapt this balance, benefitting from both types of memory.** From Table 6, compared to the retriever baseline, EAR consistently improves **Recall@5** for dialogues across all query types and maintains **Recall@5** for observations. This suggests that EAR selects more of the required dialogues in its top-5, which in turn improves downstream accuracy. For example, the largest gain appears for *Temporal* queries, where **J** increases from **37.50 to 51.56**. This indicates having an observation-dominant top-5 is insufficient for answering *Temporal* queries. EAR instead retrieves more of the relevant dialogues into the top-5, which substantially improves **J**. For *Open Domain* queries, retrieval improves, but answering accuracy remains unchanged, suggesting that performance there depends more on the LLM's parametric knowledge than on memory selection.

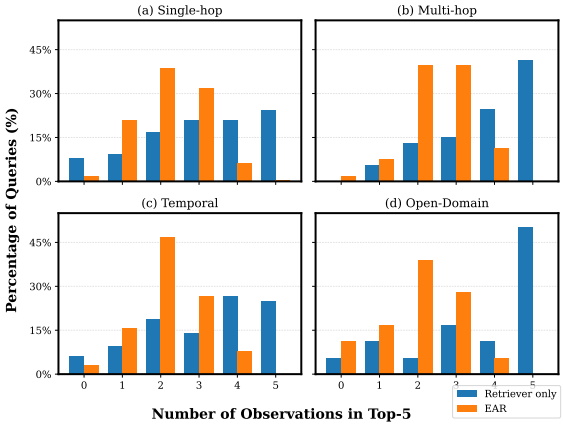

Figure 6. Number of observations chosen in Top-5 across different question types. We see that retriever almost always leans towards picking more observations in its top-5 whereas EAR's top-5 is more balanced.

Geometric Interpretation. We use a controlled diagnostic setup to study the embedding space. We curate a balanced episodic/semantic memory pool and store-specific queries, and measure how the adapter changes the query-store proximity in embedding space. After applying the trained adapter, episodic memory dependent queries move closer to episodic memories and semantic memory dependent queries move closer to semantic memories (**Appendix B.5, Tables 10–11**).

6. Conclusion and Limitations

In conclusion, we introduced the Exploratory-Assimilating Reflection (EAR) an adaptive memory retrieval framework designed to improve both retrieval quality and sample-efficient adaptation. Experiments show that our framework significantly outperforms existing adaptive methods, mitigating the severe sample inefficiency of prior approaches. We also show that EAR can retrieve an effective mixture of heterogenous memory types without any architectural changes. Our analysis demonstrates EAR's robustness against feedback noise, supporting its use in noisy settings.

While this work demonstrates a promising direction, we acknowledge several limitations that open avenues for future research. First, there is further scope for improving the strategies for utilizing experiences from the buffer; developing more sophisticated sampling or replay mechanisms is a key area for future work. Second our findings suggest that performance gains diminish as the size of the base retriever is increased. This warrants a better and more efficient RL algorithm to update the adapter effectively. In future works, we aim to investigate efficient learning techniques such as Low-Rank Adaptation (LoRA) for training adapters for larger and stronger retrievers.

References

Achiam, O. J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., ing Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., laine Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., abella Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, L., Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J. R., Knight, M., Kokotajlo, D., Kondraciuk, L., Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C., Lim, R., Lin, M., Lin, S., teusz Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., drey Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D. P., Mu, T., Murati, M., Murk, O., M'ely, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Long, O., O'Keefe, C., Pachocki, J. W., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Pokorny, M., Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J. W., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M. D., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N. A., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C. L., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., ing Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report. 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Alkhalifa, R., Borkakoty, H., Deveaud, R., El-Ebshihy, A., Espinosa-Anke, L., Fink, T., Galuščáková, P., Gonzalez-Saez, G., Goeuriot, L., Iommi, D., Liakata, M., Madabushi, H. T., Medina-Alias, P., Mulhem, P., Piroi, F., Popel, M., and Zubiaga, A. Longeval: Longitudinal evaluation of model performance at clef 2024. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction (CLEF 2024)*, Lecture Notes in Computer Science. Springer, 2024.

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Baddeley, A. Working memory: looking back and looking forward. *Nature reviews neuroscience*, 4(10):829–839, 2003.

Carbonell, J. and Goldstein, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336, 1998.

Danker, J. F. and Anderson, J. R. The ghosts of brain states past: remembering reactivates the brain regions engaged during encoding. *Psychological bulletin*, 136(1):87, 2010.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A. S., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Rozière, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., tian Cantón Ferrer, C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E. A., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron,

H., Zarov, I., Ibarra, I. A., Kloumann, I. M., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J.-Q., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., neth Heafield, K.-., Stone, K. R., El-Arini, K., Iyer, K., Malik, K., ley Chiu, K., Bhalla, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., hesh Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M. H. M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., lay Bashlykov, N., Bogoychev, N., Chatterji, N. S., Duchenne, O., cCelebi, O., Alrassy, P., Zhang, P., Li, P., Vasić, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Girdhar, R., Patel, R., Sauvestre, R., nie Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., hana Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S. C., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., ginie Do, V., Vogeti, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., ney Meers, W., Martinet, X., Wang, X., Tan, X. E., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A. K., Grattafiori, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Vaughan, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Franco, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, P.-Y. B., Loyd, B., de Paola, B., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Civin, D., Beaty, D., Kreymer, D., Li, S.-W., Wyatt, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Ozgenel, F., Caggioni, F., Guzm'an, F., Kanayet, F. J., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Thattai, G., Herman, G., Sizov, G. G., Zhang, G., Lakshminarayanan, G., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Molybog, I., Tufanov, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., KamHou, U., Saxena, K., Prasad, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Huang, K., Chawla, K., Lakhotia, K., Huang, K., Chen, L., Garg, L., Lavender, A., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Tsimpoukelli, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., ish Bansal, M., Santhanam, N., Parks, N., White, N., ata Bawa, N., Singhal, N., Egebo, N., Usunier, N., Laptev, N. P., Dong, N., Zhang, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., dro Rittner, P., Bontrager, P., Roux, P., Dollár, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Li, R., Hogan, R., Battey, R., Wang, R., Maheswari, R., Howes, R., Rinott, R., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Shankar, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S. K., Cho, S.-B., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Kohler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V. A., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wang, X., Wu, X., Wang, X., Xia, X., Wu, X., Gao, X., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Wang, Y., Hao, Y., Qian, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., and Zhao, Z. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. URL https://api.semanticscholar.org/CorpusID:271571434.

Gründel, M., Weber, M., Franke, J., and Merker, J. H. Team Galápagos Tortoise at LongEval 2024: Neural Re-Ranking and Rank Fusion for Temporal Stability. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings.

CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3740/paper-219.pdf.

IBM. *Hit rate evaluation metric*. International Business Machines Corporation, 2025. URL https://www.ibm.com/docs/en/watsonx/saas?topic=metrics-hit-rate. Accessed: 2026-03-17.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Jeong, S., Baek, J., Cho, S., Hwang, S. J., and Park, J. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.389/.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Li, H., Yang, C., Zhang, A., Deng, Y., Wang, X., and Chua, T.-S. Hello again! LLM-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5259–5276. Association for Computational Linguistics, April 2025. URL https://aclanthology.org/2025.naacl-long.272/.

Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., and Zhang, M. Towards general text embeddings with multi-stage contrastive learning, 2023. URL https://arxiv.org/abs/2308.03281.

Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., and Fang, Y. 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, 2024.

Mialon, G., Dessi, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Roziere, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., Grave, E., LeCun, Y., and Scialom, T. Augmented language models: a survey. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Robertson, S. and Zaragoza, H. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3 (4):333–389, April 2009. ISSN 1554-0669. URL https://doi.org/10.1561/1500000019.

Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: language agents with verbal reinforcement learning. NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., Nathan, A., Luo, A., Helyar, A., Madry, A., Efremov, A., Spyra, A., Baker-Whitcomb, A., Beutel, A., Karpenko, A., Makelov, A., Neitz, A., Wei, A., Barr, A., Kirchmeyer, A., Ivanov, A., Christakis, A., Gillespie, A., Tam, A., Bennett, A., Wan, A., Huang, A., Sandjideh, A. M., Yang, A., Kumar, A., Saraiva, A., Vallone, A., Gheorghe, A., Garcia, A. G., Braunstein, A., Liu, A., Schmidt, A., Mereskin, A., Mishchenko, A., Applebaum, A., Rogerson, A., Rajan, A., Wei, A., Kotha, A., Srivastava, A., Agrawal, A., Vijayvergiya, A., Tyra, A., Nair, A., Nayak, A., Eggers, B., Ji, B., Hoover, B., Chen, B., Chen, B., Barak, B., Minaiev, B., Hao, B., Baker, B., Lightcap, B., McKinzie, B., Wang, B., Quinn, B., Fioca, B., Hsu, B., Yang, B., Yu, B., Zhang, B., Brenner, B., Zetino, C. R., Raymond, C., Lugaresi, C., Paz, C., Hudson, C., Whitney, C., Li, C., Chen, C., Cole, C., Voss, C., Ding, C., Shen, C., Huang, C., Colby, C., Hallacy, C., Koch, C., Lu, C., Kaplan, C., Kim, C., Minott-Henriques, C., Frey, C., Yu, C., Czarnecki, C., Reid, C., Wei, C., Decareaux, C., Scheau, C., Zhang, C., Forbes, C., Tang, D., Goldberg, D., Roberts, D., Palmie, D., Kappler, D., Levine, D., Wright, D., Leo, D., Lin, D., Robinson, D., Grabb, D., Chen, D., Lim, D., Salama, D., Bhattacharjee, D., Tsipras, D., Li, D., Yu, D., Strouse, D., Williams, D., Hunn, D., Bayes, E., Arbus, E., Akyurek, E., Le, E. Y., Widmann, E., Yani, E., Proehl, E., Sert, E., Cheung, E., Schwartz, E., Han, E., Jiang, E., Mitchell, E., Sigler, E., Wallace, E., Ritter, E., Kavanaugh, E., Mays, E., Nikishin, E., Li, F., Such, F. P., de Avila Belbute Peres, F., Raso, F., Bekerman, F., Tsimpourlas, F., Chantzis, F., Song, F., Zhang, F., Raila, G., McGrath, G., Briggs, G., Yang, G., Parascandolo, G., Chabot, G., Kim, G., Zhao, G., Valiant,

G., Leclerc, G., Salman, H., Wang, H., Sheng, H., Jiang, H., Wang, H., Jin, H., Sikchi, H., Schmidt, H., Aspegren, H., Chen, H., Qiu, H., Lightman, H., Covert, I., Kivlichan, I., Silber, I., Sohl, I., Hammoud, I., Clavera, I., Lan, I., Akkaya, I., Kostrikov, I., Kofman, I., Etinger, I., Singal, I., Hehir, J., Huh, J., Pan, J., Wilczynski, J., Pachocki, J., Lee, J., Quinn, J., Kiros, J., Kalra, J., Samaroo, J., Wang, J., Wolfe, J., Chen, J., Wang, J., Harb, J., Han, J., Wang, J., Zhao, J., Chen, J., Yang, J., Tworek, J., Chand, J., Landon, J., Liang, J., Lin, J., Liu, J., Wang, J., Tang, J., Yin, J., Jang, J., Morris, J., Flynn, J., Ferstad, J., Heidecke, J., Fishbein, J., Hallman, J., Grant, J., Chien, J., Gordon, J., Park, J., Liss, J., Kraaijeveld, J., Guay, J., Mo, J., Lawson, J., McGrath, J., Vendrow, J., Jiao, J., Lee, J., Steele, J., Wang, J., Mao, J., Chen, K., Hayashi, K., Xiao, K., Salahi, K., Wu, K., Sekhri, K., Sharma, K., Singhal, K., Li, K., Nguyen, K., Gu-Lemberg, K., King, K., Liu, K., Stone, K., Yu, K., Ying, K., Georgiev, K., Lim, K., Tirumala, K., Miller, K., Ahmad, L., Lv, L., Clare, L., Fauconnet, L., Itow, L., Yang, L., Romaniuk, L., Anise, L., Byron, L., Pathak, L., Maksin, L., Lo, L., Ho, L., Jing, L., Wu, L., Xiong, L., Mamitsuka, L., Yang, L., McCallum, L., Held, L., Bourgeois, L., Engstrom, L., Kuhn, L., Feuvrier, L., Zhang, L., Switzer, L., Kondraciuk, L., Kaiser, L., Joglekar, M., Singh, M., Shah, M., Stratta, M., Williams, M., Chen, M., Sun, M., Cayton, M., Li, M., Zhang, M., Aljubeh, M., Nichols, M., Haines, M., Schwarzer, M., Gupta, M., Shah, M., Huang, M., Dong, M., Wang, M., Glaese, M., Carroll, M., Lampe, M., Malek, M., Sharman, M., Zhang, M., Wang, M., Pokrass, M., Florian, M., Pavlov, M., Wang, M., Chen, M., Wang, M., Feng, M., Bavarian, M., Lin, M., Abdool, M., Rohaninejad, M., Soto, N., Staudacher, N., LaFontaine, N., Marwell, N., Liu, N., Preston, N., Turley, N., Ansman, N., Blades, N., Pancha, N., Mikhaylin, N., Felix, N., Handa, N., Rai, N., Keskar, N., Brown, N., Nachum, O., Boiko, O., Murk, O., Watkins, O., Gleeson, O., Mishkin, P., Lesiewicz, P., Baltescu, P., Belov, P., Zhokhov, P., Pronin, P., Guo, P., Thacker, P., Liu, Q., Yuan, Q., Liu, Q., Dias, R., Puckett, R., Arora, R., Mullapudi, R. T., Gaon, R., Miyara, R., Song, R., Aggarwal, R., Marsan, R., Yemiru, R., Xiong, R., Kshirsagar, R., Nuttall, R., Tsiupa, R., Eldan, R., Wang, R., James, R., Ziv, R., Shu, R., Nigmatullin, R., Jain, S., Talaie, S., Altman, S., Arnesen, S., Toizer, S., Toyer, S., Miserendino, S., Agarwal, S., Yoo, S., Heon, S., Ethersmith, S., Grove, S., Taylor, S., Bubeck, S., Banesiu, S., Amdo, S., Zhao, S., Wu, S., Santurkar, S., Zhao, S., Chaudhuri, S. R., Krishnaswamy, S., Shuaiqi, Xia, Cheng, S., Anadkat, S., Fishman, S. P., Tobin, S., Fu, S., Jain, S., Mei, S., Egoian, S., Kim, S., Golden, S., Mah, S., Lin, S., Imm, S., Sharpe, S., Yadlowsky, S., Choudhry, S., Eum, S., Sanjeev, S., Khan, T., Stramer, T., Wang, T., Xin, T., Gogineni, T., Christianson, T., Sanders, T., Patwardhan, T., Degry, T., Shadwell, T., Fu, T., Gao, T., Garipov, T., Sriskandarajah, T., Sherbakov, T., Kaftan, T., Hiratsuka, T., Wang, T., Song, T., Zhao, T., Peterson, T., Kharitonov, V., Chernova, V., Kosaraju, V., Kuo, V., Pong, V., Verma, V., Petrov, V., Jiang, W., Zhang, W., Zhou, W., Xie, W., Zhan, W., McCabe, W., DePue, W., Ellsworth, W., Bain, W., Thompson, W., Chen, X., Qi, X., Xiang, X., Shi, X., Dubois, Y., Yu, Y., Khakbaz, Y., Wu, Y., Qian, Y., Lee, Y. T., Chen, Y., Zhang, Y., Xiong, Y., Tian, Y., Cha, Y., Bai, Y., Yang, Y., Yuan, Y., Li, Y., Zhang, Y., Yang, Y., Jin, Y., Jiang, Y., Wang, Y., Wang, Y., Liu, Y., Stubenvoll, Z., Dou, Z., Wu, Z., and Wang, Z. Openai gpt-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

Squire, L. R. Memory systems of the brain: a brief history and current perspective. *Neurobiology of learning and memory*, 82(3):171–177, 2004.

Tan, Z., Yan, J., Hsu, I.-H., Han, R., Wang, Z., Le, L., Song, Y., Chen, Y., Palangi, H., Lee, G., Iyer, A. R., Chen, T., Liu, H., Lee, C.-Y., and Pfister, T. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439. Association for Computational Linguistics, 2025.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models, 2021. URL https://arxiv.org/abs/2104.08663.

Wang, Y. and Chen, X. Mirix: Multi-agent memory system for llm-based agents, 2025. URL https://arxiv.org/abs/2507.07957.

Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., and Yu, D. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629, 2022. URL https://api.semanticscholar.org/CorpusID:252762395.

Yu, Y., Ping, W., Liu, Z., Wang, B., You, J., Zhang, C., Shoeybi, M., and Catanzaro, B. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Zhang, D., Li, J., Zeng, Z., and Wang, F. Jasper and stella: distillation of sota embedding models, 2025. URL https://arxiv.org/abs/2412.19048.

Zhong, W., Guo, L., Gao, Q., Ye, H., and Wang, Y. Memorybank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, Mar. 2024.

A. Implementation Details

For ease of reproduction, we provide the exact algorithms for the individual modules along with the hyperparameters used.

A.1. Exploratory Reflection

Algorithm 2 Exploratory Reflection Algorithm

Input: Query q, memory candidates $\mathcal{M} = \{m_1, \ldots, m_K\}$
Output: Selected memory set $\mathcal{S}$

1: Initialize $\mathcal{S} \leftarrow \emptyset$
2: Initialize UCB scores $c_i \leftarrow 0$ for all $i \in \{1, \ldots, K\}$
3: Initialize empirical mean rewards $\hat{r}_i \leftarrow 0$, pull counts $n_i \leftarrow 0$, and total pulls $t \leftarrow 0$
4: Initialize candidate pool $\mathcal{C} \leftarrow \mathcal{M}$
5: **for** $j = 1$ to s **do**
6: **for** each $m_i \in \mathcal{C}$ **do**
7: $s_i \leftarrow \mathrm{Sim}(q, m_i)$ *// query–memory similarity*
8: $d_i \leftarrow -\max_{m_j \in \mathcal{S}} \mathrm{Sim}(m_i, m_j)$ *// diversity penalty*
9: $c_i \leftarrow \hat{r}_i + \alpha \cdot \sqrt{\frac{\ln t}{n_i}}$ *// UCB exploration bonus*
10: **end for**
11: $i^* \leftarrow \arg\max_{m_i \in \mathcal{C}} \{w_s s_i + w_d d_i + w_c c_i\}$
12: $\mathcal{S} \leftarrow \mathcal{S} \,\|\, m_{i^*}$ *// append chosen memory*
13: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{m_{i^*}\}$ *// remove from candidate pool*
14: **end for**
15: Obtain LLM citations $y_i \in \{-1, 1\}$ for each $m_i \in \mathcal{S}$
16: **for** each $m_i \in \mathcal{S}$ **do**
17: $n_i \leftarrow n_i + 1;\ t \leftarrow t + 1$
18: $\hat{r}_i \leftarrow \hat{r}_i + \frac{y_i - \hat{r}_i}{n_i}$
19: **end for**

For Algorithm 2, we use the following hyperparameters. The hyperparameters for the Explorer were chosen empirically, to achieve right balance between similarity, diversity and exploration. The weights are designed in such a way to give equal importance to memory slate diversity and exploration within the retrieval space and slightly higher importance to similarity to candidate memories. The UCB coeffecient was set to 0.5 to achieve a balance between exploration and exploitation.

- Required Subset Size s: 5
- Coefficients w_s, w_d, w_c: 0.4, 0.3, 0.3
- UCB coefficient α: 0.5

Step-by-step walkthrough

- The explorer recieves the query and memory embeddings as input. In the EAR setup, the query and memory embeddings are modified beforehand by the adapter.
- (Line 2-13) Initially the required subset $\mathcal{S}$ empty. Then we calculate similarity s_i, diversity d_i and UCB c_i for each memory m_i. The memory maximizing the weighted average of the three scores is chosen. This chosen memory is removed from the candidate pool. This step continues till the desired subset size $|\mathcal{S}| = s$ is achieved.
- (Line 15) Once the final subset is formed, an LLM-evaluator provides binary citations (+1, -1) for each memory indicating its usefulness for answering the query.
- (Line 16 - 18) After this we update the UCB statistics (for each memory m_i - number of times it was chosen n_i, mean reward r_i and total number of arm pulls done across all memories, t).
- The above steps constitute one iteration of exploration.

Algorithm 3 Assimilating Reflection with Experience Replay

Input: Query $\mathbf{q}$, memory candidates $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_k\}$
Output: Updated (W_q, W_m)

1: Initialize replay buffer $\mathcal{B} \leftarrow \emptyset$
2: Adapt embeddings: $\mathbf{q}' \leftarrow \mathbf{q} + W_q\mathbf{q}$; $\mathcal{M}' \leftarrow \{\mathbf{m}'_i = \mathbf{m}_i + W_m\mathbf{m}_i\}_{i=1}^{k}$
3: Loss $\mathcal{L}_{\text{query}} \leftarrow 0$
4: **for** $\ell = 1$ to T **do**
5: $\mathcal{S}_\ell \sim \textsc{ExploratoryReflection}(\mathbf{q}', \mathcal{M}', T)$
6: $y_\ell \leftarrow \textsc{Citations}(\mathcal{S}_\ell)$ *// LLM citation score*
7: Update UCB statistics using $\mathcal{S}_\ell$ and y_ℓ
8: $\mathcal{L}_{\text{query}} \leftarrow \mathcal{L}_{\text{query}} - \frac{1}{|\mathcal{S}_\ell|}\sum_{m_i \in \mathcal{S}_\ell}(y_\ell^i - b)\log \pi_\phi(i \mid q', \mathcal{M}')$,
9: **end for**
10: $\mathcal{L}_{\text{query}} \leftarrow \mathcal{L}_{\text{query}}/L$
11: Store per-memory citations over T explorations in $\bar{y}$
12: **Store experience**: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(\mathbf{q}, \mathcal{M}, \bar{y}\})\}$
13: **Replay and update**
14: Sample mini-batch $\{(\mathbf{q}^{(j)}, \mathcal{M}^{(j)}, \bar{\mathbf{y}}^{(j)})\}_{j=1}^{B} \sim \mathcal{B}$
15: Experience replay loss $\mathcal{L}_{\text{batch}} \leftarrow 0$
16: **for** $j = 1$ to $|\mathcal{B}|$ **do**
17: Adapt embeddings $\mathbf{q}^{(j)}$ and $\mathcal{M}^{(j)}$
18: Sample $\tilde{\mathcal{S}}^{(j)}$ using similarity-scores between $\mathbf{q}'^{(j)}$ and $\mathcal{M}'^{(j)}$ with Gumbel trick
19: Recompute reward $\tilde{r}^{(j)}$ for $\tilde{\mathcal{S}}^{(j)}$ using $\bar{y}^{(j)}$
20: $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} - \frac{1}{|\tilde{\mathcal{S}}^{(j)}|}\sum_{m_i \in \tilde{\mathcal{S}}^{(j)}}(\tilde{r}^{(j)} - b)\log \pi_\phi(i \mid q'^{(j)}, \mathcal{M}'^{(j)})$,
21: **end for**
22: $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}}/|\mathcal{B}|$
23: $\phi \leftarrow \phi - \eta\,\nabla_\phi\big(\mathcal{L}_{\text{query}} + \mathcal{L}_{\text{batch}}\big)$

A.2. Assimilating reflection

We implement our version of reranker with RL updates inspired from (Tan et al., 2025), since their code is not publicly available. We use a similar set of parameters for the reranker as mentioned in (Tan et al., 2025). We also incorporate Exploratory Reflection and Experience Buffer to enhance the RL update.

- Batch size: 4
- Learning Rate: 0.001
- Baseline b: 0.5
- Gumbel Temperature: 0.5
- Experience replay batch size: 4

Step-by-step walkthrough

- We take query q and top-k memory candidates $\mathcal{M}$ provided by retriever as input.
- (Line 1) Initally the replay buffer is empty.
- (Line 2) Using the residual linear adapter, the query and memory embeddings are modified.
- (Line 3 - 9) Using the modified embeddings, we perform T exploration steps. At each step, the Exploratory Reflection module is invoked to provide a subset of s memories. The LLM evaluator provides citation scores which in turn is used to update UCB statistics. We then calculate a loss using the log probabilities mentioned in Section 3.2.2.
- (Line 11-12) After T explorations, the average citation recieved by each candidate memory is stored in $\bar{y}$. The query, candidate memories and average citations per memory are stored in the Experience Buffer.
- (Line 14) To replay experiences from the Experience Buffer, we sample a mini batch $\mathcal{B}$. We sample them based on cosine similarity to the current query q. Note that experience buffer contains the original query and memory embeddings before applying the adapter. Impact of different sampling strategies is discussed in Section 4.7.
- (Line 17 - 18) For each sampled past experience, we adapt the past query and memory embeddings using the current adapter. To enable stochastic sampling for each past experience we employ the Gumbel-trick. Instead of top-s directly through the updated query and memory embedding, we add Gumbel noise to the similarity scores.

$$s'_i = q'.m'_i + g_i; g_i = -log(-log(u_i)) \tag{5}$$

  Where $u_i \sim Uniform(0, 1)$. Gumbel trick perturbs the similarity scores using uniform noise. The perturbed scores are normalized using a softmax function. After sampling, we get a slate $\tilde{\mathcal{S}}$.
- (Line 19) Since we have average citations for each memory in the experience buffer, we convert it to binary rewards as

$$\tilde{r}_i = \begin{cases} +1, & \text{if } \bar{y}_i > 0 \\ -1, & \text{if } \bar{y}_i \leq 0 \end{cases} \tag{6}$$

  Using the calculated reward, loss is calculated.
- (Line 23) Using a combination of the current query loss and experience replay loss, optimization is done.

A.3. Simulating Citation Scores

To eliminate the confounding effects of the LLM on our experiment, we use a probabilistic reward model that labels the selected memories as useful/not useful to answering the query. This replaces the LLM based citation scores in our experiment. Here citations refer to which of the selected top-5 memories where found useful. So the prediction space is limited to $\mathcal{U}$.

$$\mathcal{U} = \{0, 1, 2, 3, 4\}.$$

Let $\mathcal{G}$ denote the set of unique ground-truth citations, and let $n = |\mathcal{G}|$ be the number of distinct gold citations.

True positive generation. To match a target recall $r \in [0, 1]$, each ground-truth citation $g \in \mathcal{G}$ is independently included in the prediction set with probability r. Therefore, the expected number of true positives is

$$\mathbb{E}[\text{TP}] = nr.$$

This directly aligns the expected recall with the target value, since

$$\mathbb{E}[\text{Recall}] \approx \frac{\mathbb{E}[\text{TP}]}{n} = r.$$

False positive generation. False positives are drawn from the remaining citation indices,

$$\mathcal{F} = \mathcal{U} \setminus \mathcal{G},$$

with size

$$u = |\mathcal{F}| = 5 - n.$$

Each candidate false positive $f \in \mathcal{F}$ is independently included with probability p_{fp}. This probability is chosen so that the expected precision matches a target value $p \in [0, 1]$. Using

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

and substituting the expected counts $\mathbb{E}[\text{TP}] = nr$ and $\mathbb{E}[\text{FP}] = up_{\text{fp}}$, we solve

$$p \approx \frac{nr}{nr + up_{\text{fp}}}.$$

Method	Latency (s)	Recall@5
Retriever Only	1.80	62.37
Reranker + RL Update	2.30	54.12
EAR w/o Explorer	2.30	72.80
EAR w/ Explorer	16.88	80.28

Table 7. Latency introduced by each method per query during test time. Performance is shown on LongMemEval dataset.

Citation Scores	Precision	Recall	Recall@5
Simulated	88	86	72.8
GPT 5 Mini	78	86	70.1

Table 8. Performance with GPT 5 Mini as the evaluator. Recall@5 shows recall rate of updated reranker (%) in LongMemEval dataset (i.e., Explorer is disabled at test time).

Rearranging gives

$$p_{\text{fp}} = \frac{nr(1-p)}{up}.$$

Thus, the expected number of false positives becomes

$$\mathbb{E}[\text{FP}] = up_{\text{fp}},$$

which yields the desired trade-off between recall and precision.

B. Additional Results

B.1. Performance–Cost Tradeoff

Though EAR achieves significant improvement in performance over the baseline retriever, it introduces additional computational cost in the form of extra latency and LLM call. In this section, we analyze the tradeoff between performance and latency across all compared methods.

Table 7 reports the average test-time latency per query for each method, along with the corresponding Recall@5 on LongMemEval. In these measurements, we assume that the query has already been embedded and that memory embeddings are precomputed and stored. We use a file based storage system for the embeddings. We further exclude the latency of the downstream answer generation step, so the reported numbers isolate the cost of memory retrieval.

The *Retriever Only* baseline incurs the lowest latency, as it simply returns the top-k nearest memories from the embedding store. Adding the lightweight linear adapter increases latency only marginally, from 1.80s to 2.30s per query. This adapter architecture is identical for *Reranker + RL Update* and *EAR*; therefore, when the explorer is disabled at test time, both methods are functionally equivalent in terms of inference cost. However, despite having the same test-time architecture, *EAR w/o Explorer* achieves substantially higher Recall@5 than *Reranker + RL Update*, indicating that EAR's training procedure learns a much more effective adapter.

Enabling the explorer at test time yields the best retrieval performance, improving Recall@5 from 72.80 to 80.28, an absolute gain of 7.48 points over *EAR w/o Explorer*. This gain, however, comes at the cost of additional latency: test-time inference rises from 2.30s to 16.88s per query. The increase is primarily due to the iterative exploration process, which requires an additional $T-1$ LLM calls for each query.

Overall, these results highlight two practical operating points for EAR. *EAR w/o Explorer* offers a strong performance–efficiency balance, delivering a large improvement over the baseline with only a minor latency increase. In contrast, *EAR w/ Explorer* provides the highest retrieval accuracy, but at a substantially higher inference cost. This makes the explorer particularly suitable for settings where retrieval quality is critical and additional latency can be tolerated.

B.2. Using Real World LLMs for Citations

To validate whether our simulated citation scores agree to LLM based citations, we conduct the experimental setup with GPT 5 Mini as the evaluator instead of the probabilistic model. From Table 8 We find the results with the actual LLM are close to the results with simulated probabilistic model. Results for GPT 5 Mini are reported after one run.

B.3. Impact of Base Retriever Size

We evaluate EAR with three base retrievers: Contriever, Stella-1.5B, and GTE-7B. Across both LongMemEval and LoCoMo, we observe that the absolute improvement obtained by EAR decreases as the retriever size increases. A likely reason is that larger retrievers produce higher-dimensional embeddings (Contriever: 768, Stella: 1024, GTE: 3584), which in turn increase the size of the trainable adapter. In our implementation, the number of adapter parameters scales as $2d_{\text{embed}}^2$, where d_{embed} is the embedding dimension (this comes from the two projection matrices W_q and W_m mentioned in section 3.2.2). This suggests that larger retrievers may require more training examples for EAR to realize substantial gains.

Our results support this interpretation. On LongMemEval, which contains 400 training queries, EAR with Stella improves over the baseline retriever but remains below EAR with Contriever. In contrast, on LoCoMo, which provides 1232 training queries, EAR with Stella substantially outperforms EAR with Contriever. A similar trend is observed for GTE: on LongMemEval, EAR does not surpass the baseline retriever, whereas on LoCoMo it achieves clear gains over the baseline.

Figure 7 further supports this trend on LongMemEval. Al-

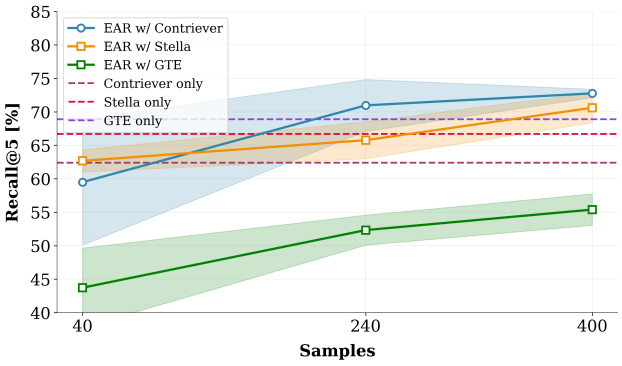

Figure 7. Comparison of recall rates across different retrievers. Recall rate in percentage on LongMemEval.

though EAR with GTE does not yet outperform the baseline in this low-data regime, its recall improves steadily during training, indicating a positive learning curve similar to the other retrievers. This suggests that with sufficient training data, EAR can also improve over stronger and larger base retrievers.

B.4. Non-RL based rerankers

In addition to our RL-trained adapter, we evaluate whether EAR can be composed with standard, non-RL rerankers that are trained for generic text reranking. Concretely, we use MONOT5-BASE-MSMARCO, a T5-base cross-encoder reranker fine-tuned on the MS MARCO passage ranking task. Given a query, MONOT5 produces relevance scores for query–memory pairs and reranks the candidate set. EAR is then applied *on top of* this reranked pool: the explorer operates over the same top-k candidates using Contriever embeddings, and we additionally make the MONOT5 scores available as side-information during exploration (e.g., as an auxiliary signal when proposing memory subsets).

Table 9 shows that combining MONOT5 with EAR improves **Recall@5** over using MONOT5 alone on both LongMemEval and LoCoMo. This result highlights a key design property of EAR: it is not tied to a particular retrieval stack. Instead, EAR functions as a *retrieval-time, feedback-driven layer* that can be placed downstream of diverse candidate generators and rerankers, leveraging their strengths while correcting failures through iterative selection.

Why feedback-driven exploration matters. Static rerankers such as MONOT5 are trained to approximate broad, general-purpose relevance notions from large public corpora. In contrast, memory retrieval for agents is often *domain- and objective-dependent* (e.g., code-assistants requiring execution-relevant context, tool-use traces, or repository-specific conventions). In such niche settings, purely static reranking may misalign with the true task objective. EAR's explorer, however, optimizes memory selection using *task feedback* (an external critic signal; an LLM in our implementation), allowing the retrieval process to adapt its behavior to the environment and the downstream success criterion without requiring the reranker to have been pretrained on that niche distribution. We therefore view EAR as complementary to pretrained rerankers: cross-encoders can improve candidate ordering, while EAR provides an adaptive mechanism to refine the final retrieved set under the task's feedback signal.

Dataset	Method	Paramter Size	Recall@5
LongMemEval	EAR	1.18M	80.28
	MonoT5	220M	82.20
	MonoT5 + EAR	221.18M	**85.50**
LoCoMo	EAR	1.18M	55.06
	MonoT5	220M	61.46
	MonoT5 + EAR	221.18M	**62.08**

Table 9. Comparison of recall rates when using RL based rerankers with EAR

B.5. Qualitative Analysis for Selection from Different Memory Stores

LoCoMo does not provide explicit labels indicating whether a query should be resolved primarily from episodic (event-like) or semantic (fact-like) memories. To probe whether the trained adapter induces *store-consistent* structure in the embedding space, we conduct a controlled diagnostic study with a clearly separated memory pool and a small set of hand-labeled queries.

S.No	Query	Memory
1	Who took chapter 6 notes on 2025-03-25?	Episodic
2	When did Kai and Rina have their prep meeting on 25th April, 2025?	Episodic
3	When did Ethan's coffee machine last malfunction?	Episodic
4	Where is the meeting agenda doc shared by Kai?	Episodic
5	At what time is Leo's party happening on 8th March 2025?	Episodic
6	What is Caroline's preference when texting?	Semantic
7	What is Leo's preference for meetings?	Semantic
8	Why does Leo use music for gatherings?	Semantic
9	What are Kai and Rina's strengths?	Semantic
10	What is Caroline's view towards LGBTQ?	Semantic

Table 10. Examples of queries and the memory type it predominantly requires.

Curated setup. We construct a pool of 50 memories with unambiguous store membership: 25 episodic and 25 se-

mantic. We also curate 10 diagnostic queries, where 5 predominantly require episodic memories and 5 predominantly require semantic memories (Table 10). These queries are not used for training; they are only used for qualitative analysis.

Proximity metric. Let q be the normalized query embedding and M_e, M_s be the sets of normalized episodic and semantic memory embeddings. We use cosine distance defined as $d(q, m) = 1 - \mathrm{Sim}(q, m)$, where $\mathrm{Sim}(\cdot, \cdot)$ denotes cosine similarity.

$$d_e(q) = \frac{1}{|M_e|} \sum_{m \in M_e} \big(1 - \mathrm{Sim}(q, m)\big), \tag{7}$$

$$d_s(q) = \frac{1}{|M_s|} \sum_{m \in M_s} \big(1 - \mathrm{Sim}(q, m)\big), \tag{8}$$

$$\Delta(q) = d_e(q) - d_s(q). \tag{9}$$

A positive Δ indicates the query is closer to semantic memories than episodic memories (i.e., $d_s(q) < d_e(q)$), while a negative Δ indicates higher affinity to episodic memories (i.e., $d_e(q) < d_s(q)$). We compute Δ_{before} using base embeddings and Δ_{after} after applying the trained adapter.

Query	Memory Required	Δ_{before}	Δ_{after}
1	Episodic	-0.001	-0.069
2	Episodic	-0.023	-0.079
3	Episodic	-0.033	-0.101
4	Episodic	-0.023	-0.050
5	Episodic	-0.008	-0.073
6	Semantic	-0.001	0.149
7	Semantic	0.033	0.054
8	Semantic	0.002	0.064
9	Semantic	-0.006	0.022
10	Semantic	-0.004	0.124

Table 11. Comparison of the query's proximity to either of the memory types before and after training the adapter.

Observation. From Table 11, we observe that prior to training the adapter, Δ_{before} values are close to zero, indicating weak separation between episodic and semantic affinity in the base embedding space. After applying the trained adapter, the magnitude of Δ increases and becomes store-consistent: episodic queries exhibit more negative Δ_{after}, whereas semantic queries exhibit more positive Δ_{after}. This suggests that the adapter reshapes the embedding space to increase the relative proximity between queries and the memory store they predominantly require.

C. Prompts Used

Prompt For Answer Generation

You are a reasoning assistant tasked with answering questions based on retrieved conversational memory.

You are given:
• A question
• The top-5 retrieved texts from a long conversation
• Each text may be:
• A dialogue entry between two participants, OR
• An observation derived from the dialogue, OR
• A dialogue entry from user to a chat assistant

Your task is to:
1. Carefully read all retrieved texts.
2. Identify the information relevant to the question.
3. Combine evidence across multiple texts if needed.
4. Resolve references (e.g., pronouns, implicit mentions).
5. If conflicting information appears, prefer the most recent or most specific evidence.
6. Return a precise concise answer. If the question cannot be answered using the retrieved texts, respond with: "Insufficient information."

Question: {question}
Retrieved Texts: {List of Retrieved Texts}

Output Format:
Answer: <final answer>

Prompt For LLM-As-a-Judge

You are an expert language model evaluator. I will provide you with a question, a ground-truth answer, and a model-generated response. Your task is to determine whether the response correctly answers the question by following these evaluation rules:
• Answer Yes if the response contains or directly matches the correct answer.
• Answer No if the response provides only a partial answer or omits essential information.
• Answer No if the response does not sufficiently address the question.
Examples:
Example 1: Correct Response
• Question: What is the capital of France?
• Ground-truth Answer: Paris
• Response: The capital of France is Paris.
Evaluation:
• Output: Yes

Example 2: Incorrect Response
• Question: What is the capital of France?
• Ground-truth Answer: Paris
• Response: France is a country in Europe.
Evaluation:
• Output: No

Additional Instructions:
• Apply the evaluation criteria consistently.
• Base your decision strictly on the information in the response.
• Avoid subjective interpretations and adhere to the provided examples.
Input:
• **Question:** {question}
• **Ground-truth Answer:** {gold_answer}
• **Response:** {answer}
Output:

Prompt For LLM Based Citations

Task Description: Given a user query and a list of memories consisting of personal summaries with their corresponding original turns, generate a natural and fluent response while adhering to the following guidelines:
• Cite useful memories using [i], where i corresponds to the index of the cited memory.
• Do not cite memories that are not useful. If no useful memory exist, output [NO_CITE].
• Each memory is independent and may repeat or contradict others. The response must be directly supported by cited memories.
• If the response relies on multiple memories, list all corresponding indices, e.g., [i, j, k].
• The citation is evaluated based on whether the response references the original turns, not the summaries.

Examples:
Case 1: Useful Memories Found
INPUT:
• User Query: What kind of work do I do and do I have any pets?
• Memories:
- Memory [0]: Speaker 1: Please make sure not to order anything with peanuts, I'm severely allergic.
- Memory [1]: Speaker 1: I need to head home a bit early today to take Max to the vet.
- Memory [2]: Speaker 1: I just landed a huge client for a new logo and branding project!

Output:
Answer: You work as a freelance graphic designer and you have a golden retriever named Max.
Citations: [1, 2]

Case 2: No Useful Memories
INPUT:
• User Query: What is my favorite movie?
• Memories:
- Memory [0]: Speaker 1: Please make sure not to order anything with peanuts, I'm severely allergic.
- Memory [1]: Speaker 1: I need to head home a bit early today to take Max to the vet.
- Memory [2]: Speaker 1: I just landed a huge client for a new logo and branding project!

Output:
Answer: I don't have enough information to answer that.
Citations: [NO_CITE]

Additional Instructions: • Ensure the response is fluent and directly answers the user's query.
• Always cite the useful memory indices explicitly.
• The citation is evaluated based on whether the response references the original turns, not the summaries.
• Follow the format of the examples provided above.

Input:
• User Query: {query}
• Memories: {memories}
Output: