Beyond Markovian Forgetfulness: Episodic Memory for Reasoning-Intensive Retrieval

Anonymous ACL submission

Abstract

Reasoning-intensive information retrieval uses large language models to solve complex queries via multi-step reasoning. However, existing methods have critical limitations. Chain-of-Thought (CoT) approaches suffer from inefficiency, while state-based methods, despite better token efficiency, often fall into reasoning cycles that trap the query refinement process. To address these issues, we propose Episodic Memory for Retrieval (EMR), which enhances the state-based framework with an episodic memory. This module stores the full history of prior states for a query, allowing the model to avoid repetition of such cycles. Experiments on the BRIGHT benchmark show that EMR consistently outperforms both CoT and state-based baselines. Moreover, it is highly token-efficient, reducing token usage by 72% on average. Our work demonstrates that episodic memory is a robust and efficient solution for advanced reasoning-intensive retrieval. The code is available in the supplementary materials.

1 Introduction

Traditional Information Retrieval (IR) systems often struggle to answer complex queries that require multi-step reasoning (Yang et al., 2018; Feldman and El-Yaniv, 2019). Reasoning-intensive IR (Xiao et al., 2024; Das et al., 2025) leverages large language models (LLMs) to enhance retrieval performance through techniques such as iterative query refinement and document reranking.

Early approaches to reasoning-intensive IR, such as Rank1 (Weller et al., 2025) and Rank-R1 (Zhuang et al., 2025) leveraged the Chain-of-Thought (CoT) capabilities of LLMs. While effective in producing detailed reasoning traces, these approaches sacrifice token efficiency. As shown in Figure 1(a), the model generates a long and redundant reasoning chain within a single step. This process can lead to a *reasoning cycle* (e.g., `AI safety` → `AI safety`) that consumes substantial computation without yielding progress.

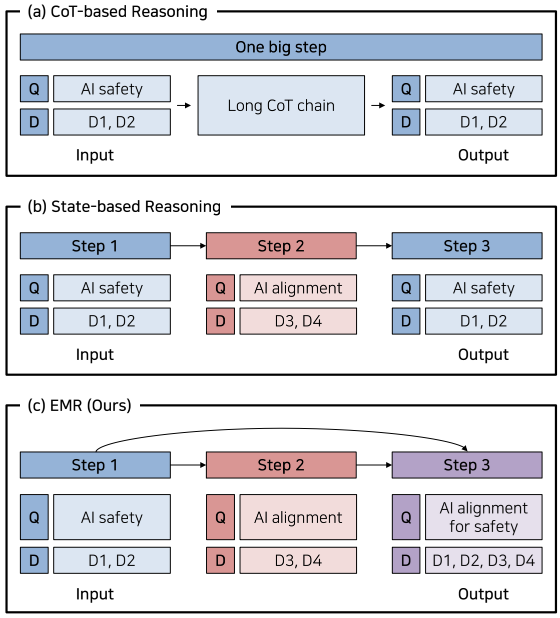

Figure 1: An illustration of different reasoning approaches for query refinement. (a) **CoT-based Reasoning** generates a single and long reasoning chain, which can be inefficient and may not result in a meaningful query update. (b) **State-based Reasoning** refines the query based solely on the immediate previous state, making it vulnerable to reasoning cycles. (c) **EMR** enhances state-based reasoning with an **episodic memory** that retains the history of all past states. This prevents cycles and enables progressive query refinement.

To improve efficiency, state-based methods such as SMR (Lee et al., 2025) decompose reasoning into iterative transitions. By design, each step is prompted to produce a new state, mitigating immediate self-loops and reducing token usage. However, this framework imposes a Markovian assumption, conditioning each new state only on the immediate predecessor. As a result, multi-step cycles can still emerge (e.g., `AI safety` → `AI alignment` → `AI safety`), as shown in Figure 1(b). In our experiments, we observed that such loops occur in up to 65% of queries, depending on the dataset.

We address these limitations with **E**pisodic **M**emory for **R**etrieval (EMR), which augments the state-based paradigm with an explicit memory module to eliminate reasoning cycles. Inspired by *episodic memory* in agentic approaches (DeChant, 2025; Nuxoll and Laird, 2012), which refers to the ability to recall past experiences, our method records the full history of states generated for a query. The model is then prompted to condition on this trajectory when producing the next state. This trajectory-aware mechanism prevents the revisiting of prior states and encourages the generation of progressively refined queries.

Figure 1(c) shows how a query progresses to refinement: For instance, by referencing its memory of having already generated `AI safety` and `AI alignment`, the model is guided to produce a more advanced query like `AI alignment for safety`, rather than repeating previous steps. This episodic memory is implemented as a text-based log within the model's system prompt, ensuring constant accessibility during inference. Furthermore, through prompt compression, EMR achieves superior token efficiency compared to the state-based method, despite the additional tokens to store past states.

To validate EMR, we conducted extensive experiments on BRIGHT, a widely-used benchmark for reasoning-intensive IR. The results demonstrate that EMR consistently outperforms existing CoT-based and state-based approaches across various settings, confirming its effectiveness. Furthermore, we show that our approach is token-efficient, reducing token consumption by 72% on average compared to the state-based approach. This result suggests that the gains in token efficiency from episodic memory are larger than the cost introduced by maintaining it.

2 Related Work

2.1 CoT-based Reasoning for IR

The advent of LLMs has enabled new approaches for addressing complex information retrieval tasks that demand multi-step reasoning. Early approaches in this domain, such as Rank1 (Weller et al., 2025) and Rank-R1 (Zhuang et al., 2025), leveraged CoT prompting to deconstruct intricate information needs.

However, the lengthy reasoning chains they generated are often redundant, which could misguide the query refinement process. Subsequent CoT compression approaches, like O1-Pruner (Luo et al., 2025), aimed to mitigate this by shortening the CoT chain. Despite these efforts, the fundamental limitation of the CoT paradigm is its single-pass generation process. This approach is vulnerable to reasoning cycles, motivating a shift toward more granular, state-based methods.

2.2 State-based Reasoning for IR

To address the inefficiencies of CoT, methods like SMR (Lee et al., 2025) reformulated the task as a sequence of state transitions. This approach significantly improved token efficiency by breaking down the reasoning process into discrete steps.

However, its reliance on a Markovian assumption makes it susceptible to multi-step reasoning cycles. This motivates a mechanism that can retain and leverage the history of past states to ensure consistent forward progress.

2.3 Memory in Agentic Approaches

The need for a history-aware mechanism has been explored in the field of agentic AI. In this domain, episodic memory (DeChant, 2025; Nuxoll and Laird, 2012) refers to an agent's ability to record and recall a sequence of past experiences, allowing it to engage in more complex planning.

Inspired by this approach, our primary contribution is the integration of this episodic memory concept into the state-based IR framework. Unlike previous methods conditioned only on the prior state, our approach conditions the policy on the entire history of previously generated queries. This allows our model to systematically prevent reasoning cycles while preserving the high token efficiency of the state-based paradigm, leading to a more robust and progressive query refinement process.

3 Method

Our approach enhances the state-based reasoning paradigm as a sequential state-transition problem, defining its core components: states, actions, and the policy model (§3.1). Building upon this foundation, we then introduce our core contribution, EMR, and describe how its episodic memory is constructed and applied to guide the reasoning process (§3.2).

3.1 Problem Setup: State-based Reasoning

Following the recent work (Lee et al., 2025), we formulate the reasoning-intensive IR task as a sequential decision-making process. This process is characterized by three fundamental components: a state representing the current context, an action that modifies it, and a policy that selects the best action. This formulation can be viewed as a Markov Decision Process (MDP), where the policy guides the transitions between states to progressively refine the initial state.

State. We define the state s_t at step t as a tuple consisting of the current query and a set of retrieved documents. This represents a snapshot of the reasoning process at a given moment. Formally, a state is defined as:

$$s_t = (q_t, D_t) \tag{1}$$

where $q_t \in \mathcal{Q}$ is the query at step t, with $\mathcal{Q}$ being the list of all possible queries. $D_t \subset \mathcal{D}$ is the list of documents retrieved using the query q_t, where $\mathcal{D}$ represents the entire document corpus. The goal is to transition from an initial state s_0 to a final state s_n where D_n is optimized for the user's information need.

Action. An action a_t represents a strategic operation chosen by the policy to advance the search process. Our action space $\mathcal{A}$ consists of three distinct operations:

$$\mathcal{A} = \{\ \texttt{REFINE},\ \texttt{RERANK},\ \texttt{STOP}\ \} \tag{2}$$

The `REFINE` action is deployed to broaden the search space. It uses the LLM to generate a new query q_{t+1}, retrieves a new set of documents, and expands the current document list D_t by appending these new results. Conversely, the `RERANK` action is used to exploit the currently held information. It leverages the LLM to perform a listwise reranking of documents in D_t, prioritizing the most relevant items. For computational feasibility, this list is then truncated to the top-k entries, creating a more focused document set D_{t+1} while the query q_t remains stable. Finally, the process terminates with the `STOP` action when the policy determines that the gathered documents in D_t are sufficient. This terminates the loop and returns D_t as the output.

Policy. The Policy π serves as the decision-making engine of our reasoning agent, strategically guiding the search process toward a resolution. Embodied by an LLM, the policy assesses the current state $s_t = (q_t, D_t)$ at each step t to determine the most promising action $a_t \in \{\texttt{REFINE}, \texttt{RERANK}, \texttt{STOP}\}$ to execute. For efficiency, the LLM generates both the chosen action and its direct result within a single forward pass. For instance, if the policy chooses `REFINE`, it outputs the action itself along with the newly refined query. This one-call-per-step architecture minimizes latency and computational cost. To ensure the reasoning process remains bounded, the policy is also constrained by a predefined maximum number of steps, $T_{\max}$. If the current step count reaches this threshold, the policy issues a `STOP` action, preventing overly long or costly reasoning chains.

Reasoning Cycle. A significant drawback of the state-based formulation is its vulnerability to reasoning cycles, where the agent becomes trapped in a loop by revisiting previously explored states. This issue is a direct consequence of the policy's Markovian nature: its decisions are conditioned solely on the current state s_t, with no memory of the path taken to reach it. Formally, the reasoning cycle is defined as follows:

$$q_t = q_{t'} \ \ (t' < t) \tag{3}$$

where equality denotes exact lexical identity. Such loops prevent meaningful progress, leading to redundant computation and suboptimal retrieval performance. This limitation highlights the need for a mechanism that can break the Markovian chain by incorporating a memory of its history into the agent.

3.2 Proposed: EMR

This section discusses how EMR introduces an episodic memory, by first describing the conceptual role of episodic memory in state-based reasoning (§3.2.1) and then detailing its implementation for reading and writing states in textual form (§3.2.2). Finally, we introduce the memory compression mechanism designed to maintain high token efficiency, even with the overhead of storing memory states (§3.2.3).

3.2.1 Role of Episodic Memory

The role of the episodic memory in EMR is to break the Markovian dependency inherent in the standard state-based approach. As defined in §3.1, the vanilla policy π makes decisions based solely on the current state s_t, rendering it blind to the path taken to arrive there.

To address this, we introduce an episodic memory, denoted as M_t, which is the complete history

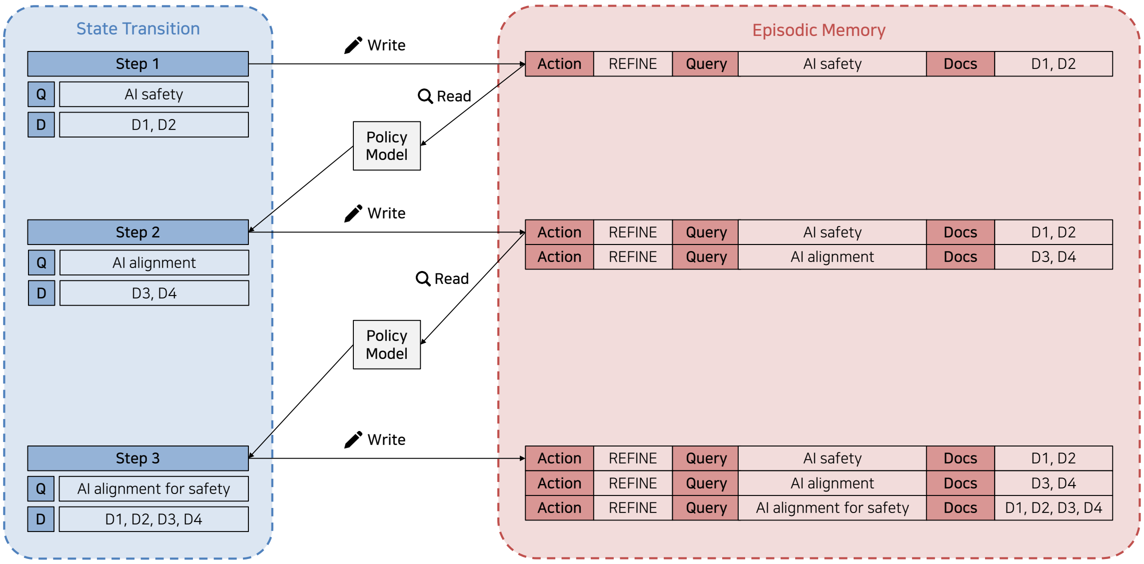

Figure 2: An illustration of the EMR architecture and the role of episodic memory. The figure shows the interplay between the iterative **State Transition** process (left) and the **Episodic Memory** (right). At each step, the policy model generates the next state based on the memory's history, and then updates the memory with that newly generated state.

of all states visited up to step t. Formally, the memory is a sequence of past states:

$$M_t = (s_0, s_1, \ldots, s_{t-1}) \tag{4}$$

where $s_i = (q_i, D_i)$ is the state at step i.

By incorporating this memory, we redefine the policy function. The original policy $\pi(s_t)$ is transformed into a memory-augmented policy, π', which conditions its decisions on both the current state and the entire history stored in the memory. The action selection process is thus updated as follows:

$$a_t = \pi'(s_t, M_t) \tag{5}$$

This seemingly simple modification has two profound implications for the reasoning process, as illustrated in Figure 2.

First, the memory-augmented policy can explicitly prevent cycles. By having access to the set of all past queries $\{q_0, q_1, \ldots, q_{t-1}\}$ within M_t, the policy π' can be instructed to avoid generating any query that has been previously explored. This directly resolves the cyclic behavior defined in Equation 3.

Second, beyond cycle prevention, the memory provides a rich historical context to generate more targeted queries, rather than just reacting to the immediate present. The following sections will detail the practical implementation of how this conceptual memory is read and written as a textual log within the LLM's prompt.

3.2.2 Memory Read and Write Operations

To utilize the conceptual memory M_t described in the previous section, we designed a token-efficient textual format that can be seamlessly integrated into the policy model's prompt. This process involves two key operations: writing the historical states into a textual log and reading this log as context for the next decision. The entire memory structure is prepended to the main task prompt at each step t, ensuring the model has constant access to its history.

The memory is formatted as plain text and organized into two distinct sections as shown in Table 1: a history of actions and a repository of document contents.

```
## History of Recent Actions
[1] Action: {a_1} Query: {q_1} Ranks: {IDs of D_1}
...
[t-1] Action: {a_{t-1}} Query: {q_{t-1}} Ranks: {IDs
      of D_{t-1}}

## Memory of Documents
[ID of d_1] {d_1}
[ID of d_2] {d_2}
...
[ID of d_n] {d_n}
```

Table 1: Text format of the history of previous states in the system prompt of EMR.

`The History of Recent Actions` section

serves as a log of the reasoning trajectory. A crucial design choice for token efficiency is that each entry stores only the identifiers (IDs) of the documents retrieved at that step, rather than their full contents. This significantly reduces the token count that grows with each step, as document contents can be hundreds of tokens long while their IDs are just single tokens or short strings.

The `Memory of Documents` section acts as a deduplicated repository. It maps document IDs to their full textual content. This structure decouples the log from the voluminous document content, preventing redundant storage of the same document if it is retrieved multiple times across different steps.

We deliberately chose a simple plain text format over structured formats like JSON or XML. Our preliminary experiments indicated that while structured formats offered no significant performance improvement, they consistently incurred a higher token overhead due to syntactic characters (e.g., brackets, quotes, and tags). The plain text approach provides the best balance of performance and token economy for this task.

Read Operation. The read operation is performed implicitly at the beginning of each step t. The entire two-part memory block, representing M_t, is formatted as a single string and placed within the LLM's prompt, typically following the system message. This provides the full historical context required for the memory-augmented policy $\pi'(s_t, M_t)$ to make its next decision.

Write Operation. The write operation occurs after the policy executes an action a_t and a new state $s_t = (q_t, D_t)$ is generated. The process involves two updates:

- A new formatted string is appended to the History of Recent Actions.
- Each document in the retrieved set D_t is checked. If a document's ID is not already present in the Memory of Documents, its ID and full content are appended to that section.

This read-write cycle continues until the STOP action is issued, ensuring that the memory is always a complete and up-to-date record of the entire reasoning process.

3.2.3 Memory Compression

Despite deduplication, the cumulative length of unique document contents still poses a significant challenge to token efficiency. As a result, as the reasoning process explores more documents, the memory's token footprint can grow excessively, leading to increased computational costs and potential context window limitations. To mitigate this issue, we introduce a memory compression mechanism that reduces each document to only its most query-relevant sentences.

This compression is performed via a three-step extractive summarization process whenever a new set of documents D_t is retrieved by a query q_t.

Sentence Segmentation. All documents in D_t are split into sentences using spaCy[1], creating a large pool of candidate sentences.

Relevance Scoring. Each sentence is scored for relevance to q_t using a pre-trained MiniLM cross-encoder[2].

Global Filtering and Composition. Unlike per-document selection, we rank all sentences from the entire pool and keep only the top-k overall. This global selection ensures that only the most relevant sentences enter the Memory of Documents, implicitly pruning irrelevant documents entirely. The resulting compressed summaries reduce token overhead while focusing the policy model on the most salient information for ongoing reasoning.

4 Results and Analysis

4.1 Experimental Setup

Datasets. We evaluate our method using the BRIGHT benchmark (Su et al., 2024), which is designed for reasoning-intensive IR tasks. The benchmark is composed of 12 diverse datasets spanning domains such as mathematics, code, and scientific questions. To demonstrate the generalizability of our approach, we report performance across all of them.

Evaluation Metrics. Consistent with the baseline methods, we employ nDCG@10 as our primary evaluation metric, a standard metric in IR. We also report other metrics such as Recall and MAP in the Appendix to further validate our robustness.

Baselines. We compare EMR against the following baselines. See Appendix A.1 for more details.

[1] https://spacy.io/

[2] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2

- **CoT-based Reasoning**: We include Rank1 (Weller et al., 2025) and Rank-R1 (Zhuang et al., 2025), which are strong CoT-based models, and O1-Pruner (Luo et al., 2025), a method for compressing reasoning trajectories.

- **State-based Reasoning**: We use SMR (Lee et al., 2025), which models the retrieval process as transitions between states.

- **Retriever**: We include BM25 (Robertson et al., 2009), a traditional sparse retriever, and ReasonIR (Shao et al., 2025), a dense retriever trained for reasoning tasks.

Implementation Details. All baselines follow published implementations provided in official repositories and papers. Hyperparameters not mentioned here follow those in the original publications. To ensure a comprehensive comparison across LLMs with differing performance, we conduct our experiment on both Qwen2.5-32B and Qwen3-32B. See Appendix A.2 for more details.

4.2 Analysis

We structure our analysis around three research questions to validate the effectiveness of EMR.

4.2.1 RQ1. Does Reducing Cycles with Episodic Memory Boost Performance?

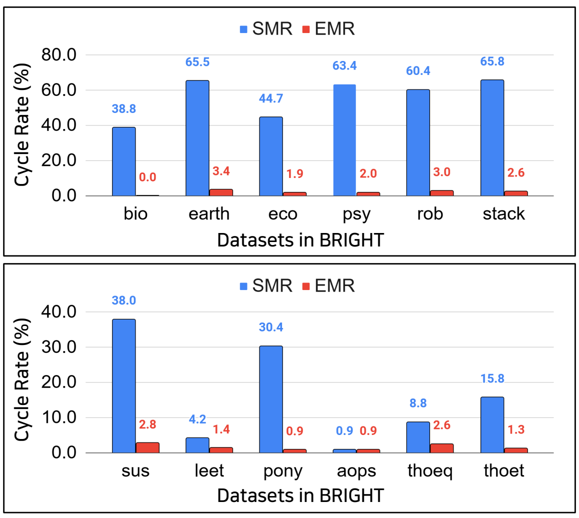

Figure 3: Comparison of the cycle ratio between EMR and SMR on the BRIGHT benchmark. The ratio indicates the percentage of queries that encounter one or more reasoning cycles during the refinement process.

Reducing Reasoning Cycles. To answer RQ1, we first measure the occurrence of reasoning cycles as defined in Eq. 3: any instance where a refined query revisits a previously generated one.

As illustrated in Figure 3, EMR (red) consistently suppresses reasoning cycles compared to SMR (blue) across all BRIGHT datasets. Aggregated over all datasets, the average cycle rate is reduced by 94%, from 38.75% for SMR to just 2.25% for EMR.

A closer analysis reveals that the baseline's vulnerability to reasoning cycles is heterogeneous, varying with the type of the queries in each domain. On datasets like *Leetcode* and *AoPS*, SMR already exhibits few cycles. This is likely because their queries often contain code snippets or mathematical formulas, whose high lexical specificity makes exact repetition less probable. In contrast, on datasets with natural language queries such as *EarthScience*, *Psychology*, and *StackOverflow*, SMR is highly vulnerable to cyclic behavior, with cycle rates exceeding 60%. Across all datasets, EMR reduces the cycle rates to under 4%. This demonstrates that episodic memory provides a robust solution to the reasoning cycle problem inherent in state-based models.

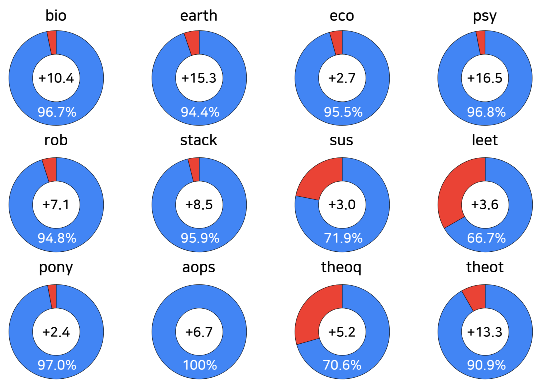

Figure 4: Analysis of queries that encounter cycles with SMR baseline. For each BRIGHT dataset, the donut chart displays the percentage of these queries for which EMR successfully resolves the cycle (blue). The central number indicates the average nDCG@10 point gain for these resolved queries.

Performance Gain in Cycle Queries. To further quantify the impact of cycle mitigation, we conducted a fine-grained analysis on queries where SMR encounters a reasoning cycle. We measure both the rate at which EMR resolves the cycle and the subsequent impact on retrieval performance.

	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Leet	Pony	AoPS	TheoQ	TheoT	**Avg**
Retriever													
BM25	18.9	27.2	14.9	12.5	13.6	18.4	15.0	24.4	7.9	6.2	10.4	4.9	14.5
CoT-based Reasoning													
BM25 + Rank1	22.1	31.7	14.6	15.7	15.8	17.7	20.3	22.9	**9.7**	5.9	11.8	9.3	16.5
BM25 + Rank-R1	23.6	33.5	16.8	15.4	18.8	18.4	20.3	24.9	9.1	6.9	12.1	8.7	17.4
BM25 + O1-Pruner	23.6	31.9	17.7	18.0	17.2	20.0	19.8	24.2	8.4	6.6	10.7	7.0	17.1
State-based Reasoning													
BM25 + SMR (Qwen2.5)	28.7	34.9	20.4	20.7	20.9	20.8	19.2	22.1	6.3	6.3	18.0	20.3	19.9
BM25 + SMR (Qwen3)	44.0	42.1	22.9	23.3	18.3	21.1	29.5	14.5	7.8	3.1	6.9	6.4	20.0
EMR: State-based Reasoing with Episodic Memory													
BM25 + EMR (Qwen2.5)	41.2	42.1	**24.7**	33.1	19.2	23.8	26.1	**26.4**	8.3	6.8	**26.5**	**29.2**	26.4
BM25 + EMR (Qwen3)	**53.1**	**54.6**	23.2	**37.4**	**23.4**	**26.9**	**31.8**	15.8	9.0	**7.2**	20.9	17.3	**26.7**

Table 2: Retrieval performance (nDCG@10) on the **BRIGHT** benchmark using **BM25** as the underlying retriever. All methods differ only in their reasoning strategy. Best scores per dataset are bolded.

	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Leet	Pony	AoPS	TheoQ	TheoT	**Avg**
Retriever													
ReasonIR	43.6	42.9	32.7	38.8	20.9	25.8	27.5	31.5	19.6	7.4	33.1	35.7	30.0
CoT-based Reasoning													
ReasonIR + Rank1	49.7	35.8	22.0	37.5	22.5	21.7	35.0	18.8	**32.5**	10.8	22.9	43.7	29.4
ReasonIR + Rank-R1	50.3	36.1	24.4	38.9	23.7	22.0	34.2	22.6	31.3	10.7	24.5	44.0	30.2
ReasonIR + O1-Pruner	48.6	37.4	24.4	38.8	24.0	21.6	35.3	22.5	31.1	11.2	24.0	**44.1**	30.3
State-based Reasoning													
ReasonIR + SMR (Qwen2.5)	52.2	51.1	27.5	45.1	22.8	29.4	30.9	27.9	18.1	7.4	36.4	32.6	31.8
ReasonIR + SMR (Qwen3)	53.7	52.3	29.5	46.9	24.5	30.6	32.5	29.0	19.8	9.2	38.0	34.2	33.4
EMR: State-based Reasoing with Episodic Memory													
ReasonIR + EMR (Qwen2.5)	55.9	55.4	33.3	49.4	26.7	33.6	35.1	31.2	22.6	11.1	39.0	35.6	35.7
ReasonIR + EMR (Qwen3)	**56.9**	**56.0**	**33.5**	**50.0**	**28.2**	**34.1**	**35.7**	**32.4**	23.1	**12.4**	**40.5**	37.2	**36.7**

Table 3: Retrieval performance (nDCG@10) on **GPT4-Reason queries** of the **BRIGHT** benchmark. Best scores per dataset are bolded.

Figure 4 shows that EMR successfully resolves the cycle in over 90% of these failure cases on average, as indicated by the blue portion of each chart. Moreover, this resolution translates into substantial gain. The number at the center of each chart represents the nDCG@10 gain for these queries, amounting to 8%p increase in average. Notably, the largest performance gains are observed in datasets with natural language queries, such as *EarthScience* (+15.3) and *Psychology* (+16.5). This observation aligns with our earlier findings, confirming that EMR effectively prevents cycles even in scenarios where the baseline is most prone to failure.

4.2.2 RQ2. Is EMR More Effective and Efficient?

To answer RQ2, we designed a comprehensive evaluation to measure two key performance axes: retrieval effectiveness and token efficiency.

Retrieval Effectiveness. As shown in Table 2, which evaluates performance in nDCG@10 with the BM25 retriever, EMR consistently outperforms all CoT-based and state-based baselines. On average, EMR achieves a 6.7%p gain in nDCG@10 compared to SMR. See Appendix A.3 for a comprehensive evaluation across other metrics such as Recall and MAP.

This performance gain is not just the result of better initial query refinement of LLMs. To verify this, we conducted an experiment using strong initial queries refined by GPT-4, provided by the BRIGHT benchmark. As detailed in Table 3, EMR remains the top-performing method, still outperforming SMR by a 3.3%p margin in nDCG@10.

Furthermore, the superiority of EMR is not simply from using a more powerful LLM. Across both

	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Leet	Pony	AoPS	TheoQ	TheoT	**Avg**
Retriever													
ReasonIR	26.3	31.5	23.3	30.3	17.8	24.0	20.6	35.0	10.3	14.3	31.6	27.2	24.4
CoT-based Reasoning													
ReasonIR + Rank1	32.0	30.0	22.3	31.1	16.7	25.8	22.4	31.4	15.5	12.1	27.8	26.3	24.5
ReasonIR + Rank-R1	33.0	34.1	24.2	33.5	20.2	25.4	22.8	33.5	14.1	10.7	30.3	27.8	25.8
ReasonIR + O1-Pruner	31.6	34.9	24.5	33.2	21.0	24.9	24.7	33.3	11.8	12.9	29.9	27.1	25.8
State-based Reasoning													
ReasonIR + SMR (Qwen2.5)	34.7	35.1	26.2	32.8	20.9	25.2	24.2	30.8	10.4	13.5	30.1	28.6	26.0
ReasonIR + SMR (Qwen3)	41.7	37.8	26.4	31.2	20.3	26.0	32.5	28.8	10.5	14.8	33.0	28.5	27.6
EMR: State-based Reasoing with Episodic Memory													
ReasonIR + EMR (Qwen2.5)	41.7	41.5	31.4	36.6	22.8	25.8	34.9	32.0	11.2	15.1	38.4	34.3	30.5
ReasonIR + EMR (Qwen3)	**52.4**	**46.8**	**32.2**	**39.9**	**24.7**	**30.2**	**35.3**	**36.1**	**12.2**	**15.3**	**40.7**	**34.6**	**33.4**

Table 4: Retrieval performance (nDCG@10) on the **BRIGHT** benchmark using **ReasonIR** as the underlying retriever. All methods differ only in their reasoning strategy. Best scores per dataset are bolded.

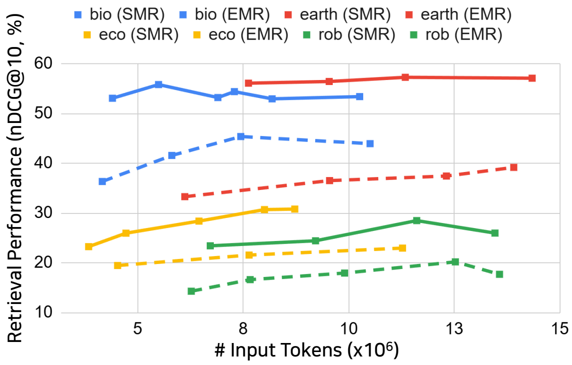

Figure 5: Retrieval performance versus cumulative input tokens for EMR (solid lines) and SMR (dashed lines) on selected BRIGHT datasets. Each color represents a different dataset.

Table 2 and Table 3, while the Qwen3-based models generally outperform Qwen2.5-based models, EMR maintains the highest average score regardless of the underlying LLM. This indicates that episodic memory provides a robust advantage that is not dependent on initial query quality or biased towards a specific LLM.

Token Efficiency. Beyond performance, EMR demonstrates superior token efficiency. Figure 5 shows retrieval performance against the number of tokens consumed for the representative four datasets of the BRIGHT benchmark. In this figure, each color represents a distinct dataset, while solid lines denote EMR and dashed lines indicate SMR. See Appendix A.4 for more details.

The figure shows that EMR dominates SMR across all token budget ranges. This confirms that the token overhead introduced by maintaining the episodic memory is more than offset by the efficiency gains from avoiding redundant reasoning steps and our memory compression strategy.

4.2.3 RQ3. Is EMR Complementary to Stronger Retrievers?

To answer RQ3, we investigate whether the benefits of EMR are complementary to the gains from using more advanced retrieval components.

We test if the performance gains from EMR persist when using a more powerful retriever. We replaced the sparse retriever BM25 with ReasonIR, a strong dense retriever trained for reasoning tasks. Table 4 shows that the advantage of EMR remains with a significant margin. Specifically, EMR with Qwen3 achieves an average nDCG@10 of 33.4, surpassing the strongest baseline (SMR with Qwen3) by 5.8%p. This demonstrates that the improvements from EMR's trajectory-aware reasoning and ReasonIR's stronger retrieval are orthogonal and additive.

5 Conclusion

In this work, we address reasoning cycles in state-based approaches for reasoning-intensive IR by introducing EMR, a framework that integrates an episodic memory module to guide the reasoning process. By conditioning the policy on its entire history of visited states, EMR reduces the cycle rate by about 94%. Experiments on the BRIGHT benchmark demonstrate that this approach consistently outperforms existing CoT-based and state-based baselines in both retrieval effectiveness and token efficiency.

6 Limitations

While EMR demonstrates robust performance, it has a modular design that allows for several promising extensions. For instance, the current action space is intentionally concise (`Refine`, `Rerank`, `Stop`). This set could easily be expanded to include more specialized tools like external API calls and user feedback, transforming EMR into a more comprehensive information searcher. Such interactions would enhance our episodic memory with a more comprehensive history.

References

Debrup Das, Sam O' Nuallain, and Razieh Rahimi. 2025. Rader: Reasoning-aware dense retrieval models. *arXiv preprint arXiv:2505.18405*.

Chad DeChant. 2025. Episodic memory in ai agents poses risks that should be studied and mitigated. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 321–332. IEEE.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. *arXiv preprint arXiv:1906.06606*.

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*.

Dohyeon Lee, Yeonseok Jeong, and Seung-won Hwang. 2025. From token to action: State machine reasoning to mitigate overthinking in information retrieval. *arXiv preprint arXiv:2505.23059*.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

Andrew M Nuxoll and John E Laird. 2012. Enhancing intelligent agents with episodic memory. *Cognitive Systems Research*, 17:34–48.

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

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, et al. 2025. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint arXiv:2504.20595*.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. Rank1: Test-time compute for reranking in information retrieval. *arXiv preprint arXiv:2502.18418*.

Chenghao Xiao, G Thomas Hudson, and Noura Al Moubayed. 2024. Rar-b: Reasoning as retrieval benchmark. *arXiv preprint arXiv:2404.06347*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2025. Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning. *arXiv preprint arXiv:2503.06034*.

A Appendix

A.1 Baseline Methods

To provide a comprehensive evaluation of EMR, we compare it against various baseline methods. All models described in this section are released under the MIT License.

CoT-based Reasoning. We include CoT-based baselines as they are standard approach in the literature and represent a widely adopted methodology, ensuring our evaluation is grounded in established practices.

- **Rank1** (Weller et al., 2025) and **Rank-R1** (Zhuang et al., 2025) are strong CoT-based reasoning models. Rank1 is fine-tuned on reasoning traces, while Rank-R1 further enhances this capability through reinforcement learning. We select these models over earlier methods like ReAct (Yao et al., 2023) because they are specifically adapted for IR and represent the leading CoT-based methods in this domain.

- **O1-Pruner** (Luo et al., 2025) focuses on the efficiency of CoT. It employs reinforcement learning to compress the lengthy reasoning paths generated by CoT models, aiming to reduce computational overhead without sacrificing the final answer quality.

State-based Reasoning. We include state-based baselines to compare EMR, aiming to demonstrate not only superior overall performance but also how our proposed episodic memory component enhances the efficiency of state transitions.

- **SMR** (Lee et al., 2025) frames the multi-step retrieval process as a Markov Decision Process. At each step, a policy model observes the current state (e.g., the query and retrieved documents) and selects a discrete action to transition to the next state, finally building a set of relevant documents for the answer.

Retriever. We include both a traditional sparse retriever and a strong dense retriever to investigate the synergy between the retrieval backbone and EMR. This comparison is crucial to demonstrate that the benefits of EMR are orthogonal to the retriever's capabilities. Our method provides consistent improvements even when paired with a more powerful retriever.

- **BM25** (Robertson et al., 2009) serves as our traditional sparse retrieval baseline. It is a keyword-based algorithm that has been a longstanding and robust baseline in IR for decades, relying on term frequency and inverse document frequency.

- **ReasonIR** (Shao et al., 2025) represents a strong baseline in dense retrieval for reasoning-intensive IR. It is highly effective for tasks that require understanding context and intent beyond simple keyword matching. Its strong performance over other notable dense retrievers like Contriever (Izacard et al., 2021) and RankLLaMA (Ma et al., 2024) makes it a challenging and relevant baseline.

A.2 Implementation Details

Models and Adaptations. The baseline models used in our experiments are derived from various LLMs. Rank1-32B utilizes Qwen2.5-32B[1] as its backbone, while Rank-R1-14B is built upon Qwen2.5-14B[2]. Due to the absence of a 32B version for Rank-R1, our experiments employ the 14B model. For O1-Pruner, which is not inherently a retrieval model, we implemented a custom prompt to enable query rewriting and document reranking functionalities; its 32B version originates from QwQ-32B-preview[3]. To ensure a fair and comprehensive evaluation, we also benchmark against the base LLMs, Qwen2.5-32B[1] and Qwen3-32B[4].

Experimental Settings. All experiments were performed on a single NVIDIA A6000 GPU. The hyperparameters for EMR were fixed across all datasets and models unless otherwise stated: we set the batch size to 8, LLM temperature to 0.1, top-k for retrieval to 10, and the maximum number of reasoning steps to 16. For calculating computational cost, we define token usage as the sum of all input tokens across the reasoning process.

To ensure a fair comparison and isolate the improvements of our method, we use the same system prompt as SMR for our policy model, as shown in Table 5. This approach mitigates potential performance differences due to prompt variations.

A.3 Evaluation on Other Metrics

To further demonstrate the robustness of our proposed method, this section supplements the primary nDCG@10 evaluation with results on additional metrics. We evaluated performance on the BRIGHT benchmark using BM25 as the retriever, measuring results with two additional metrics: MAP@10 (shown in Table 6) and Recall@10 (shown in Table 7). Consistent with our main findings, EMR maintains a clear performance advantage over both CoT and compressed CoT baselines across these metrics.

A.4 Token Efficiency

We assess the token efficiency of EMR by comparing its token consumption to that of SMR under the same experimental setup (top-k=10). Table 8 shows that EMR is more efficient, requiring about 72% fewer tokens on average. This demonstrates that the episodic memory in EMR not only enhances retrieval performance but also achieves superior token efficiency.

[1] Qwen/Qwen2.5-32B-Instruct
[2] Qwen/Qwen2.5-14B-Instruct
[3] Qwen/QwQ-32B-Preview
[4] Qwen/Qwen3-32B-FP8

````
You are a highly intelligent artificial agent responsible for managing a search system. Your role is to
     either refine the given query or re-rank retrieved search results, thereby enhancing both recall and
     precision of the search. You can output exactly one of the following operations, after which another
     agent will execute it and return the results to you.

## Input Format
The input provided to you will have the following structure:

```
{
"query": "<current version of a query>",
"ranks": [
    ("<docid>", "document contents"),
    ("<docid>", "document contents"),
    ...
]
}
```

### Decision policy (check in order):

1. Query Refinement
   Choose "action = refine" if any of the following are met:
   - The query is ambiguous or generic
   - The retrieved search results are unsatisfactory
   - The query is short
   - Key domain terms are missing in the query

2. Re-ranking
   Choose "action = refine" only if the query already looks good and at least one retrieved document seems
        on-topic.

3. Stop
   Choose "action = stop" only when you are *certain* that no further improvement is possible.


## Possible Outputs (select exactly one)

### Query Refinement
You may refine the query by rewriting it into a clear, specific, and formal version that is better suited
     for retrieving relevant information from a list of passages. Only return the document IDs (`docid`) in
     the `reranked` list. Do not include document contents. Output format:

```
{
"action": "refine",
"query": "<refined version of a query>",
"reason": "<reason for this action>"
}
```

### Re-ranking
You may reorder the retrieved documents (do not remove non-relevant ones). The results should be sorted in
     descending order of relevance. Output format:

```
{
"action": "rerank",
"ranks": ["<docid>", "<docid>", ...],
"reason": "<reason for this action>"
}
```

### Stop
You may stop this iteration when the results are satisfactory. Output format:

```
{
"action": "stop",
"reason": "<reason for this action>"
}
```
````

Table 5: System prompt used in EMR.

	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Leet	Pony	AoPS	TheoQ	TheoT	**Avg**
Retriever													
BM25	13.1	18.1	8.9	7.0	8.5	13.4	10.0	19.7	1.6	6.2	10.4	4.9	10.2
CoT-based Reasoning													
BM25 + Rank1	16.1	23.0	7.9	11.3	10.5	12.3	16.2	17.6	**2.2**	3.1	10.4	8.6	11.6
BM25 + Rank-R1	17.7	25.0	10.3	10.7	14.1	13.1	16.1	20.2	2.1	4.9	10.8	7.7	12.7
BM25 + O1-Pruner	17.2	22.6	10.7	12.9	12.7	14.9	15.5	19.5	1.8	3.7	8.9	5.6	12.2
State-based Reasoning													
BM25 + SMR (Qwen2.5)	22.0	25.7	13.6	15.4	15.0	15.5	16.6	21.0	1.6	3.1	14.1	15.2	14.9
BM25 + SMR (Qwen3)	23.1	24.4	14.0	15.7	15.3	16.2	17.0	21.8	1.9	3.4	16.2	15.4	15.4
EMR: State-based Reasoing with Episodic Memory													
BM25 + EMR (Qwen2.5)	33.9	42.1	**16.1**	26.3	15.2	17.3	19.7	**22.0**	1.9	4.0	**24.2**	**25.8**	**20.7**
BM25 + EMR (Qwen3)	**44.3**	**44.5**	15.4	**28.2**	**17.3**	**20.5**	**24.6**	10.4	**2.2**	**4.2**	19.1	15.2	20.5

Table 6: Retrieval performance (MAP@10) on the **BRIGHT** benchmark using **BM25** as the underlying retriever. All methods differ only in their reasoning strategy. Best scores per dataset are bolded.

	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Leet	Pony	AoPS	TheoQ	TheoT	**Avg**
Retriever													
BM25	21.7	31.9	16.8	15.6	19.6	21.3	21.3	29.5	4.1	6.0	8.6	9.2	17.1
CoT-based Reasoning													
BM25 + Rank1	21.7	31.9	16.8	15.6	19.6	21.3	21.3	29.5	4.1	6.0	8.6	9.2	17.1
BM25 + Rank-R1	21.7	31.9	16.8	15.6	19.6	21.3	21.3	29.5	4.1	6.0	8.6	9.2	17.1
BM25 + O1-Pruner	23.8	34.4	20.2	19.8	19.6	21.3	21.5	29.5	2.3	6.0	12.9	10.3	18.5
State-based Reasoning													
BM25 + SMR (Qwen2.5)	25.3	33.0	19.5	21.7	21.4	24.6	21.6	26.6	3.6	6.1	20.9	24.8	20.8
BM25 + SMR (Qwen3)	25.9	34.1	22.9	22.4	20.8	25.0	22.1	27.2	3.7	6.6	21.3	25.1	21.1
EMR: State-based Reasoing with Episodic Memory													
BM25 + EMR (Qwen2.5)	38.3	47.0	**23.3**	34.5	17.4	29.1	26.8	**30.7**	**4.3**	6.4	**26.0**	**30.7**	26.2
BM25 + EMR (Qwen3)	**52.8**	**53.7**	23.2	**38.3**	**24.3**	**32.5**	**31.2**	25.6	**4.3**	**6.9**	21.0	17.9	**27.6**

Table 7: Retrieval performance (Recall@10) on the **BRIGHT** benchmark using **BM25** as the underlying retriever. All methods differ only in their reasoning strategy. Best scores per dataset are bolded.

Dataset	SMR	EMR	Reduction
bio	10.5	4.4	-58.1%
earth	40.7	5.0	-87.7%
econ	14.3	3.8	-73.2%
psy	22.7	3.4	-85.1%
rob	18.5	6.7	-63.8%
stack	30.2	5.1	-83.0%
sus	13.8	5.1	-62.8%
leet	6.5	5.0	-22.6%
pony	5.0	2.0	-60.1%
aops	3.3	2.9	-11.0%
theoq	6.2	4.3	-31.0%
theot	3.9	1.3	-66.9%
Average	14.6	4.1	-72.1%

Table 8: Comparison of token usage (in millions) between EMR and SMR across various datasets in the BRIGHT benchmark. The values represent the total number of input tokens required per task. The final column shows the average of all datasets.