From Token to Action: State Machine Reasoning to Mitigate Overthinking in Information Retrieval

Dohyeon Lee[1], **Yeonseok Jeong**[2], **Seung-won Hwang**[1 2*]
Computer Science and Engineering, Seoul National University[1],
Interdisciplinary Program in Artificial Intelligence, Seoul National University[2]
{waylight3, jys3136, seungwonh}@snu.ac.kr

Abstract

Chain-of-Thought (CoT) prompting enables complex reasoning in large language models (LLMs), including applications in information retrieval (IR). However, it often leads to overthinking, where models produce excessively long and semantically redundant traces with little or no benefit. We identify two key challenges in IR: redundant trajectories that revisit similar states and misguided reasoning that diverges from user intent. To address these, we propose **S**tate **M**achine **R**easoning (SMR), a transition-based reasoning framework composed of discrete actions (REFINE, RERANK, STOP) that support early stopping and fine-grained control. Experiments on the BEIR and BRIGHT benchmarks show that SMR improves retrieval performance (nDCG@10) by 3.4% while reducing token usage by 74.4%. It generalizes across LLMs and retrievers without requiring task-specific tuning, offering a practical alternative to conventional CoT reasoning.[1]

1 Introduction

Chain-of-Thought (CoT) prompting (Wei et al., 2022) has emerged as a powerful paradigm for enhancing large language model (LLM) reasoning across complex tasks (Guo et al., 2025; Team, 2025) such as math and planning. This paper focuses on extending CoT-based reasoning to information retrieval (IR) (Li et al., 2025b; Shao et al., 2025), to better align retrieval with user intent for improved query rewriting (Ma et al., 2023) and document reranking (Sun et al., 2023).

Despite its strengths, CoT prompting often suffers from *overthinking* (Chen et al., 2024; Liu et al., 2024; Fan et al., 2025)—the generation of semantically redundant or incorrect reasoning steps that offer no benefit and may even be detrimental.

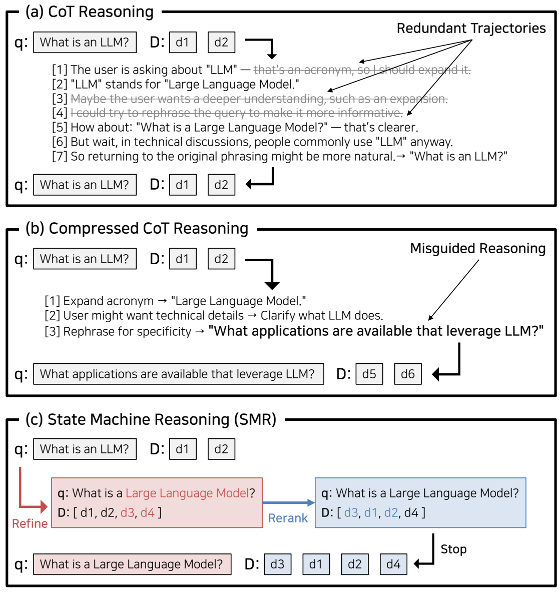

Figure 1: Conceptual comparison between standard CoT reasoning, compressed CoT reasoning, and our proposed SMR framework. (a) Standard CoT performs token-level generation in a single forward pass, often leading to redundant intermediate states. (b) Compressed CoT shortens trajectories via reinforcement learning, at the risk of misaligned outputs. (c) SMR decomposes reasoning into structured state transitions (q, D) guided by IR-specific actions.

We identify two key IR-specific challenges arising from this inefficiency that remain unaddressed by current methods.

Redundant Reasoning Standard CoT prompting generates reasoning at the token level, often revisiting semantically equivalent steps and continuing unnecessarily even after reaching the answer. As illustrated in Figure 1(a), CoT-based query rewriting frequently produces redundant paraphrases that fail to introduce new evidence for retrieval, e.g., text shown in grey in the figure, which only inflate with no gain in inference. A useful evidence that is missing would be a passage mentioning LLM

* Corresponding Authors
[1]The code and details are available at https://github.com/ldilab/SMR.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7048–7064
November 4-9, 2025 ©2025 Association for Computational Linguistics

in a full name, to retrieve a higher recall results including d3 and d4.

Misguided Reasoning In an attempt to compress CoT trajectories using reinforcement learning (RL) to shorten reasoning paths, as in O1-Pruner (Luo et al., 2025), reasoning may misalign to user intent in open-ended domain (Li et al., 2025a). As illustrated in the bold text of Figure 1(b), such compression can result in syntactically concise yet semantically misaligned queries, e.g., retrieving LLM "applications," drifting from an original intention of its "definition". This misalignment causes the retrieval of irrelevant documents such as d5 and d6. Furthermore, compression requires task-specific training and reward engineering, and thus limits generalization.

To address these challenges, we propose **S**tate **M**achine **R**easoning (SMR), a transition-based reasoning framework based on structured state transitions that avoids token-level generation. Instead of relying on autoregressive decoding or compressing reasoning trajectories, SMR formulates reasoning as transitions between intermediate representation states. Inspired by decision-theoretic frameworks (Liu et al., 2023), SMR represents each step as a transition between states (q, D), where q denotes the current query, and D a ranked list of retrieved documents. Figure 1(c) illustrates how SMR reaches the goal. Our policy model first selects the REFINE action to expand the acronym "LLM," followed by the RERANK action to reorder the newly found document d3, which is the most relevant. This policy model terminates when next step produces no gains, by selecting STOP action. This mechanism ensures that each state transition yields a incremental gain and preserves the evolving context through structured state representations. This formulation brings two key advantages.

Token Efficiency Our method avoids redundant reasoning and achieves greater token efficiency by operating over explicitly defined states. Each step updates a structured state, allowing the system to detect when it revisits an equivalent state. This contrasts with token-level generation, which lacks a mechanism to recognize semantic repetition.

Action Effectiveness By grounding each reasoning step in IR-relevant operations, SMR enables retrieval systems to make improvements through two actions: REFINE for query rewriting and RERANK for document reranking. These actions enable the system to reissue queries when current results are insufficient, or reorder documents when initial rankings are suboptimal. This design supports incremental retrieval improvements and offers precise control over which component of the pipeline to adjust at each step. In contrast, token-level generation lacks explicit validation of improvement at each step, can resulting in outputs that diverge from user intent.

We validate SMR on two benchmarks: BEIR (Thakur et al., 2021), a widely-used benchmark for standard IR evaluation, and BRIGHT (Su et al., 2024), which emphasizes reasoning-intensive retrieval. SMR achieves up to **3.4%** improvement in nDCG@10 while reducing token usage by **74.7%** on the BRIGHT benchmark, consistently outperforming prior baselines across both retrieval quality and efficiency metrics. SMR demonstrates robustness across sparse/dense retrievers and LLMs with and without reasoning. These results validate the practicality and generalizability of our approach, suggesting that SMR can be seamlessly integrated into a wide range of retrieval systems without requiring model-specific tuning.

2 Related Work

2.1 LLM-based Reasoning for IR

LLMs have recently been used to interleave reasoning with retrieval actions, enhancing retrieval quality via query understanding and iterative refinement. While early neural retrievers relied on static query-document matching (Karpukhin et al., 2020; Nogueira and Cho, 2019), later approaches incorporated reasoning modules into the retrieval loop. These methods typically operate at the token level, relying on long reasoning traces that are not always efficient or robust. ReAct (Yao et al., 2023) pioneered interleaved reasoning and acting for multi-hop QA, inspiring follow-up work in CoT-based query rewriting (Li et al., 2025b) and document reranking (Weller et al., 2025; Zhuang et al., 2025). While ReAct interleaves reasoning and tool use, it lacks task-specific structure and often relies on verbose traces, limiting its efficiency in retrieval-focused tasks.

2.2 Overthinking in CoT Reasoning

Despite their effectiveness, CoT-based models often suffer from *overthinking*, unnecessarily long reasoning sequences that offer no additional semantic gain (Chen et al., 2024; Liu et al., 2024; Fan et al., 2025). In the context of IR, this manifests as two distinct problems: (1) *Redundant trajectories*, where semantically similar steps are revisited repeatedly, and (2) *Misguided reasoning*, where shortened traces drift away from user intent or task objectives. These issues increase inference cost and degrade alignment with the user's retrieval goal.

Recent work (Sui et al., 2025; Qu et al., 2025; Wang et al., 2025) has investigated approaches to mitigate overthinking by compressing CoT traces or learning to generate shorter ones. O1-Pruner (Luo et al., 2025) applies RL to shorten reasoning while preserving output correctness. Other methods use value estimation (Shridhar et al., 2023) or supervised distillation to train models to prefer concise traces.

2.3 Our Distinctions

Compared to LLM-based reasoning like ReAct (Section 2.1), our controlled action space enables a principled transition-based control against overthinking, without incurring task-specific training or reward engineering of existing work (Section 2.2). We are the first to generalize state machine, beyond math and code (Liu et al., 2023) where verification is straightforward, to retrieval. We show SMR is tuning-free, generalizable, and performs consistently across retrievers and LLMs.

3 Proposed Method

Our framework can be viewed as a Markov Decision Process (MDP) with a discrete action space and structured state representation. While we do not explicitly learn a value function, our design is inspired by decision-theoretic frameworks (Liu et al., 2023), where each reasoning step corresponds to a transition between abstract states. Based on this formulation, we recast reasoning as transitions over IR-specific states and actions, rather than as token-level generation.

3.1 Structured State Representation

To address the first challenge (redundant trajectories), we propose representing the reasoning state as a structured tuple (q_t, D_t) at each step t, where q_t denotes the current query and D_t is a ranked list of top-k retrieved documents:

$$s_t = (q_t, D_t), \quad D_t = \{d_1, d_2, \dots, d_k\}. \tag{1}$$

This representation serves as the foundation of our transition-based reasoning by explicitly encoding the query and its retrieved documents. Specifically, the query q preserves the user's intent, while the document list D encapsulates the retrieval context that guides subsequent reasoning steps.

The initial state $s_0 = (q_0, D_0)$ is constructed using the user-issued query q_0 and the corresponding retrieved documents D_0 obtained from a static retriever. Subsequent reasoning steps update either the query or the document list, producing a trajectory of structured states. This formulation guides reasoning via IR-specific actions while maintaining alignment with retrieval quality and intent.

Stop Decision To avoid redundant reasoning and mitigate overthinking, SMR employs a stopping mechanism that detects when the system has reached an equivalent state. Instead of explicitly comparing the current state $s_t = (q_t, D_t)$ with all previously visited states $\{s_0, s_1, \dots, s_{t-1}\}$, we assess whether the current state provides any meaningful gain. Specifically, if the retrieved documents are identical and the query remains unchanged from the previous state s_{t-1}, then s_t is treated as equivalent to s_{t-1}. This ensures that state transitions reflect incremental improvements rather than redundant cycles. Crucially, this mechanism allows the model to determine whether it has gathered sufficient information to stop reasoning, a capability that conventional CoT-based methods lack. This decision is primarily made by the policy model, which selects the STOP action when it predicts that further reasoning would yield no meaningful change.

3.2 IR-Specific Reasoning Actions

To address the second challenge (misguided reasoning), we replace token-level decoding with a set of controllable reasoning actions. This action space allows the system to updates the current reasoning state based on its retrieval context, rather than relying on implicit generation dynamics that may drift from user intent.

Our action space $\mathcal{A}$ consists of three discrete actions:

$$\mathcal{A} = \{\textsc{Refine}, \textsc{Rerank}, \textsc{Stop}\}. \tag{2}$$

Each action modifies the reasoning state to improve retrieval quality or terminate the reasoning process.

Refine The REFINE action updates the query to better reflect the user's information need, guided by the current retrieval context:

$$q_{t+1} = \mathrm{LLM}_{\textsc{Refine}}(q_t, D_t). \tag{3}$$

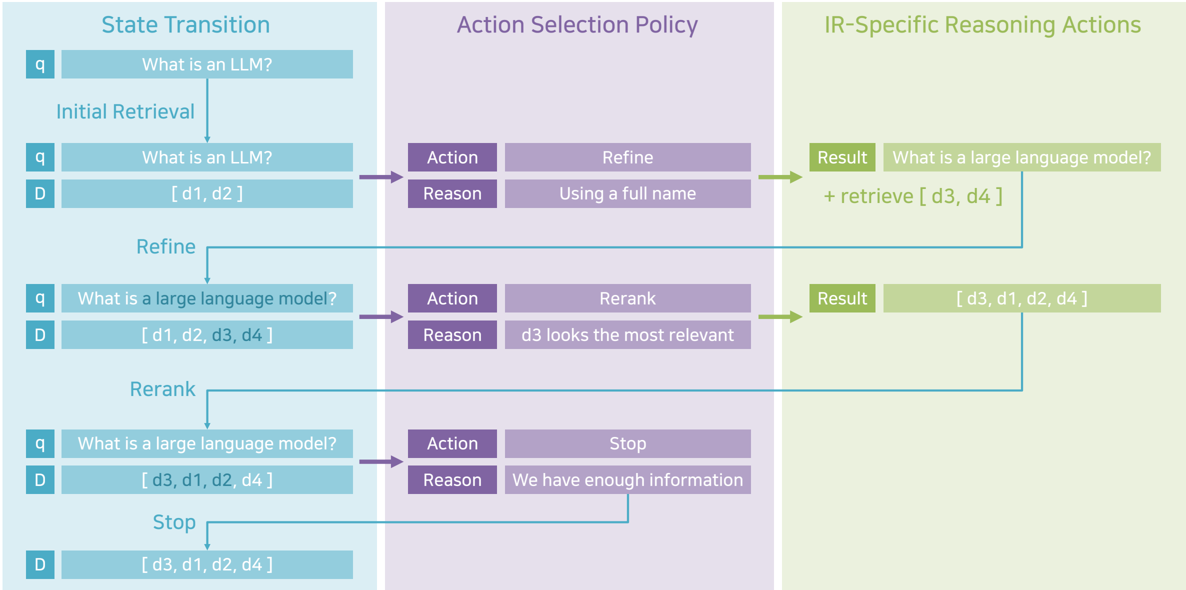

Figure 2: Illustration of our proposed reasoning framework (SMR). Beginning from an initial query and its retrieved documents, SMR transitions through structured states via three actions: REFINE (query rewriting), RERANK (document reordering), and STOP (termination). At each step, the LLM selects an action with justification, and the document list is updated accordingly.

This enables the model to perform query rewriting or expansion, conditioned on evidence from the retrieved documents.

Following each REFINE action, we invoke the retriever with the updated query q_{t+1} to obtain new candidate documents. If any of the retrieved documents are not already present in D_t, they are appended to the end of the current list. This augmentation strategy ensures meaningful evolution of the retrieval state without discarding the existing context. For further details on the REFINE action, refer to Appendix A.1.

Rerank The RERANK action adjusts the ordering of the document list without modifying the query:

$$D_{t+1} = \text{LLM}_{\text{RERANK}}(q_t, D_t). \qquad (4)$$

It refines the document ranking when the initial retrieval is imperfect, allowing better relevance estimation while keeping the query fixed.

To address potential hallucinations during reranking, we impose structural constraints on the output. Specifically, if the reranked list contains documents that were not present in the original D_t, we discard those entries and exclude them from the updated state. This prevents spurious hallucinated documents from contaminating the reasoning trajectory. Conversely, if the reranked list omits any documents from the original set, we re-append the missing items to the end of the list in their original order. This ensures that no context is inadvertently lost during reranking, preserving the integrity of the retrieval state across transitions.

Stop The STOP action terminates the reasoning process, returning the current state $s_t = (q_t, D_t)$ as the final output. This allows the system to avoid unnecessary steps once sufficient retrieval quality is achieved, promoting token efficiency and preventing semantic drift.

In addition to semantic equivalence, we also impose a hard cap on the total number of reasoning steps to control inference cost. Specifically, we set a maximum number of transitions, typically `max_steps = 16`, beyond which the system automatically issues a STOP action. This provides an upper bound on computation and ensures robustness in deployment scenarios with limited resources. When compute budget permits, increasing `max_steps` can yield improved retrieval performance by allowing deeper reasoning.

By decoupling reasoning into these interpretable transitions, our framework enables targeted improvements at each stage, supports early stopping, and remains robust across diverse retrieval scenarios without task-specific training.

3.3 Action Selection Policy

At each reasoning step, the system selects one of the three available actions based on the current state (q_t, D_t). Instead of relying on a fixed heuristic or trained controller, we adopt a prompt-based strategy where an LLM itself determines the next action. In this setup, the LLM serves as a judge that evaluates the current reasoning context and chooses the most appropriate next step.

At each step, the system selects the next action based on the current state using an instruction-following LLM guided by a structured prompt. The prompt describes the agent's role as a decision-maker responsible for improving the quality of retrieved results, and presents the current query and its associated documents in a structured format. The LLM is prompted to choose exactly one of the three actions. The full prompt format is provided in Appendix A.2.

Rather than relying on learned scoring or implicit decoding dynamics, the action decision is made through rule-grounded prompting that embeds heuristics into the LLM's instruction space. For example, the LLM is encouraged to choose REFINE when the query appears vague or the results are unsatisfactory, and to prefer STOP only when confident that no further improvement is possible. This design allows the decision process to remain transparent, easily modifiable, and robust to misalignment.

While the rationale is not used by the system for downstream decisions, we retain it for transparency and interpretability. This enables human inspection of the model's behavior and helps ensure that the policy aligns with the intended design. Remaining details of SMR are provided in Appendix A.3.

4 Results and Analysis

4.1 Experimental Setup

Datasets We evaluate our method on two retrieval benchmarks with differing reasoning complexity. **BRIGHT** (Su et al., 2024) is a recent benchmark designed to evaluate reasoning-intensive retrieval scenarios. It consists of 12 subsets spanning diverse domains such as StackExchange forums, coding problems, and STEM question-answering tasks, where queries often require domain-specific reasoning to identify relevant passages. This serves as our primary benchmark, enabling us to assess how well different methods handle complex reasoning demands. For completeness, we also evaluate on **BEIR** (Thakur et al., 2021), a widely used benchmark comprising diverse domains (e.g., factoid QA, biomedical search, and entity-centric retrieval). BEIR primarily measures general retrieval performance and helps validate the effectiveness of our method in standard IR settings.

Evaluation Metrics We report **nDCG@10** as the primary evaluation metric, following standard IR practice. Additional ranking metrics such as MAP and Recall are reported in the Appendix A.5 for completeness.

Baselines We compare SMR against the following baselines:

- **Retrievers**: BM25 (Robertson et al., 2009) is a traditional sparse retriever widely used in IR. ReasonIR (Shao et al., 2025) is a state-of-the-art dense retriever specifically trained for general reasoning tasks, outperforming other strong retrievers such as Contriever (Izacard et al., 2021) and RankLLaMA (Ma et al., 2024).

- **Standard CoT**: Rank1 (Weller et al., 2025) and Rank-R1 (Zhuang et al., 2025) are CoT-based reasoning models that represent the state-of-the-art among fine-tuned and RL-trained approaches. We select these models over ReAct (Yao et al., 2023) as they are specifically adapted for IR and represent the leading CoT-based reasoning in this domain.

- **Compressed CoT**: O1-Pruner (Luo et al., 2025) is an RL-based method for compressing reasoning trajectories while preserving answer quality.

To ensure a fair comparison across LLMs with differing capabilities, SMR is implemented using two backbone models: non-reasoning (Qwen2.5-32B,) and reasoning (QwQ-32B) LLMs. This allows us to isolate the effectiveness of the reasoning framework from the inherent capabilities of the underlying language model.

Implementation Details All baselines follow public implementations provided in official repositories, with necessary adaptations for our experimental protocol. Hyperparameters not mentioned here follow those in the original publications. See Appendix A.4 for complete implementation details.

4.2 Analysis

To further validate the effectiveness of SMR, we assess whether our goals have been achieved through the following three research questions.

4.2.1 RQ1: Does SMR Improve Retrieval Effectiveness?

Overall Effectiveness To assess the retrieval effectiveness of SMR, we report the nDCG@10 scores of the final ranked documents on the BRIGHT benchmark which is a recent suite of tasks explicitly designed to evaluate retrieval performance under reasoning-intensive scenarios. Evaluations on additional metrics such as MAP and Recall are provided in Appendix A.5.

Table 1 and Table 2 compare our method against a variety of baselines, including standard CoT-based reranking approaches (Rank1, Rank-R1) and a compressed variant (O1-Pruner), under both sparse (BM25) and dense (ReasonIR) retrieval settings. Across all 12 domains, SMR consistently outperforms both standard and compressed CoT baselines. Notably, both variants of SMR achieve the highest average nDCG@10 gain of **+5.4%** on sparse retriever and **+2.1%** on dense retriever, demonstrating that our structured reasoning framework remains highly effective in complex retrieval scenarios, regardless of the underlying LLM.

Policy Effectiveness Table 3 analyzes how SMR adapts its actions based on query type. Based on the original category of BRIGHT (Su et al., 2024), we categorize queries into three groups: **Theorem-based** (ThoeT, ThoeQ, AoPS), **Coding** (Pony, Leet), and **Stack Exchange** (others).

The model's strategy changes with query difficulty. For challenging queries (Theorem-based and Coding), it favors the REFINE action, an "explore-first" approach to discover relevant documents through multiple trajectories. Conversely, for simpler queries (Stack Exchange), it predominantly uses the RERANK action, an "exploit" strategy to finalize the ranking of already-found documents in fewer trajectories. Full results for each dataset are given in Appendix A.6.

This pattern correlates strongly with underlying retrieval difficulty. Faced with limited initial evidence, SMR adaptively runs multiple rounds of query refinement, attempting to surface better documents before stopping. This behavior demonstrates that our action policy is not based on simple rule-based heuristics, but rather leverages contextual cues to make informed decisions.

Moreover, we conduct an ablation study comparing SMR with several fixed or naive strategies in Appendix A.7, demonstrating the effectiveness of our prompt-based policy design. Specifically, we compare our full policy with three simplified baselines: one that uses only the REFINE action, one that uses only RERANK, and one that selects actions uniformly at random. SMR consistently achieves the best performance among these, indicating that our prompt-based policy design effectively balances refinement and reranking to support more accurate and efficient reasoning.

Intent Drift To verify that our iterative query refinement does not suffer from intent drift, we automatically evaluated the semantic alignment between the original and rewritten queries on the BRIGHT benchmark. At each REFINE step (up to 16 per query), we measured how well the rewritten query preserved the original intent. We use GPT-4o-mini as the reference evaluator to compute alignment scores. On average, across all datasets and refinement steps, intent alignment consistently remained above **0.9**, indicating that the refined queries retained strong fidelity to the user's original intent throughout the reasoning process. More details are provided in Appendix A.8.

4.2.2 RQ2: Does SMR Enhance Token Efficiency?

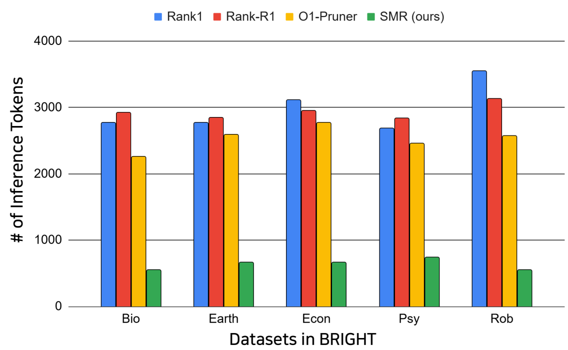

Figure 3: Inference token usage across five representative datasets in the BRIGHT benchmark. SMR (green bars) achieves significantly lower token consumption than Rank1, Rank-R1, and O1-Pruner, while improving retrieval performance. Full results including all datasets are presented in Appendix A.9.

Token Efficiency To evaluate the token efficiency of our method, we compare the total number of inference tokens used by each system on the

	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 Reasoning													
Rank1 (32B)	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
Rank-R1 (14B)	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
Compressed CoT Reasoning													
O1-Pruner (32B)	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
Ours													
SMR (Qwen2.5-32B)	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**
SMR (QwQ-32B)	**28.8**	34.4	19.8	**21.0**	19.8	**21.1**	**21.4**	**25.5**	7.5	5.7	16.9	17.2	**19.9**

Table 1: Retrieval performance (nDCG@10) on BRIGHT benchmark using **sparse retriever (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	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 Reasoning													
Rank1 (32B)	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
Rank-R1 (14B)	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
Compressed CoT Reasoning													
O1-Pruner (32B)	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
Ours													
SMR (Qwen2.5-32B)	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
SMR (QwQ-32B)	**35.2**	**35.5**	**27.0**	**33.7**	19.4	**25.9**	23.4	31.6	11.3	**14.1**	**30.8**	**29.8**	**26.5**

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

Query Type	**Refine Ratio (%)**	**Rerank Ratio (%)**	**Average Steps**
Theorem-based	63.4	36.6	13.1
Coding	55.3	44.7	11.2
Stack Exchange	22.8	77.2	9.6

Table 3: Distribution of reasoning actions and average steps by SMR on the BRIGHT benchmark, grouped by query type.

BRIGHT benchmark. Notably, SMR executes both the policy decision and the REFINE/RERANK steps using a single LLM via prompt-based interaction, without invoking any additional tools. As a result, the total token count serves as a practical estimate of real-world inference latency. Full results including all datasets are presented in Appendix A.9.

As shown in Figure 3, SMR (green bars) consistently consumes significantly fewer tokens across five representative datasets (Bio, Earth, Econ, Pay, Rob) compared to prior baselines including Rank1, Rank-R1, and O1-Pruner. On average, SMR reduces inference token usage by **74.4%** compared to the CoT-based baselines, outperforming even CoT compression methods such as O1-Pruner, which achieve only marginal reductions under **5%**.

While these baselines often perform redundant reasoning steps, SMR minimizes such redundancy by proposing structured states and terminating reasoning early when convergence is detected. This allows our method to achieve superior retrieval performance (see Table 1 and Table 2) while using substantially less computational resources.

To demonstrate this efficiency, we have conducted a cost analysis on an NVIDIA A6000 GPU with vLLM to provide a holistic view. The average number of REFINE actions per query was 2.97, with each additional retrieval pass taking approximately 1.5 seconds, resulting in an average overhead of $4.46 = 2.97 \times 1.5$ seconds. In contrast, SMR saves over 2,000 tokens per query versus baselines like Rank1. Generating these tokens with a 32B LLM on the same GPU takes 166 seconds. Since $166 \gg 4.46$, the token savings result in a substantial net gain in computational efficiency and a reduction in overall latency, confirming that the benefits of SMR's token efficiency win the minor cost of iterative retrieval.

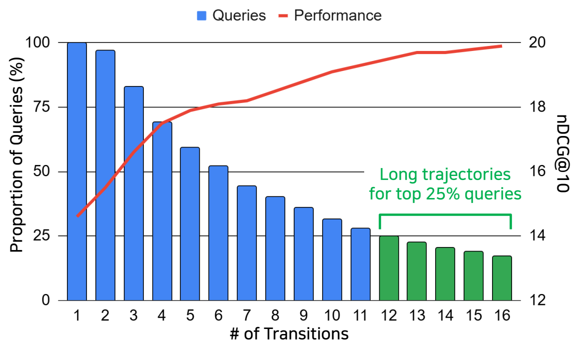

Figure 4: Transition statistics of SMR on the BRIGHT benchmark. The blue bars indicate the number of reasoning steps across all queries, computed cumulatively per transition depth. The red curve shows the average retrieval performance (nDCG@10) at each step.

Action Efficiency To evaluate how our policy model selects actions efficiently for queries of varying complexity, we analyze the distribution of action lengths across all queries in the BRIGHT benchmark, as shown in Figure 4. Each bar shows the cumulative number of queries that reached that reasoning step. For instance, a query that required 3 reasoning steps in total is counted in the bars for steps 1, 2, and 3.

We observe that **25%** of queries terminate within **3 steps**, and **50%** within **6 steps**, indicating that our system converges quickly. This validates our early stopping mechanism and shows that many queries can be resolved efficiently without overthinking.

At the same time, **25%** of queries (green bar) exhibit **12 or more** actions, especially for complex or ambiguous inputs, which benefit from additional refinement and reranking. This illustrates the flexibility of SMR which selectively allocates computation where needed. Examples of such queries are provided in Appendix A.10. The examples show how SMR successfully navigates these hard queries, while the baselines fail to handle them.

Notably, the red curve in Figure 4 shows the average retrieval performance (nDCG@10) after each transition step. Retrieval performance improves steadily as reasoning progresses, demonstrating that our policy model incrementally updates the state with meaningful gains. While we enforce a maximum of 16 steps due to computational constraints, the upward trend indicates that further improvements may be possible with extended actions, highlighting our capacity for continued enhancement when resources permit.

4.2.3 RQ3: Is SMR Generally Applicable?

	DBpedia	SciFact	FiQA	Avg
Retriever				
GPL	36.1	66.5	32.9	45.2
CoT Reasoning				
Rank1 (32B)	38.2	70.2	35.8	48.1
Rank-R1 (14B)	38.9	71.6	37.1	49.2
Compressed CoT Reasoning				
O1-Pruner (32B)	32.6	71.9	36.0	46.8
Ours				
SMR (Qwen2.5-32B)	37.6	**73.1**	**38.7**	**49.8**
SMR (QwQ-32B)	**39.5**	71.2	36.5	49.1

Table 4: Retrieval performance (nDCG@10) on BEIR datasets. All methods use GPL (Wang et al., 2021) as the retriever, varying only in reasoning strategy. Best scores per dataset are bolded.

Generalizability To assess the general applicability of SMR beyond reasoning-intensive scenarios, we evaluate its performance on the BEIR benchmark (Thakur et al., 2021), which covers a broad range of standard IR tasks without explicit reasoning demands. Table 4 presents the nDCG@10 results on three BEIR datasets (DBpedia, SciFact, and FiQA).

Table 4 shows that SMR achieves the highest average performance among all methods, outperforming both standard CoT and a compressed reasoning approach. This demonstrates that our structured reasoning approach is not specialized solely for reasoning-intensive scenarios. Instead, its core mechanism of identifying and refining information gaps via a structured state appears to be a fundamental benefit that transfers effectively to standard retrieval tasks.

Furthermore, Table 4 demonstrates that SMR delivers robust performance with both Qwen2.5-32B and QwQ-32B backbones. Both variants achieve results that are either comparable to or superior to the strongest baselines. This indicates that the performance gains stem from our structured reasoning framework itself, rather than from the capabilities of a particular underlying LLM.

5 Conclusion

We presented SMR, a structured reasoning for retrieval by modeling reasoning as transitions over discrete states, defined as queries and document rankings. To mitigate overthinking, unlike prior methods such as CoT prompting or ReAct relying on token-level generation or tool calls, SMR takes an action-driven approach constraining reasoning to a set of IR-specific actions and grounding each step in a well-defined state. SMR offers a compact and controllable alternative to token-level reasoning with three key benefits: (1) improved retrieval performance, (2) reduced token usage through early stopping and redundancy checks, and (3) robust generalization across retrievers and LLMs without task-specific tuning. Experiments on the BRIGHT and BEIR benchmarks validate these advantages, with SMR outperforming strong baselines in both complex and standard IR settings.

6 Limitation

While SMR offers a modular and token-efficient alternative to conventional CoT reasoning, our current state representation is limited to representing the query and the top-k retrieved documents, and does not incorporate user interaction signals such as click-through rates or engagement metrics. Incorporating such behavioral feedback could further enhance state fidelity and reasoning quality. Moreover, while we restrict reasoning actions to REFINE, RERANK, and STOP, the modular nature of our framework allows integration of additional operations such as domain-specific retrieval modules, document filters, or other tools without requiring architectural changes.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00414981) and the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2025-2020-0-01789) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

References

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. 2025. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill? *arXiv preprint arXiv:2504.06514*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Xiaomin Li, Zhou Yu, Zhiwei Zhang, Xupeng Chen, Ziji Zhang, Yingying Zhuang, Narayanan Sadagopan, and Anurag Beniwal. 2025a. When thinking fails: The pitfalls of reasoning for instruction-following in llms. *arXiv preprint arXiv:2505.11423*.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.

Jia Liu, Jie Shuai, and Xiyao Li. 2023. State machine of thoughts: Leveraging past reasoning trajectories for enhancing problem solving. *arXiv preprint arXiv:2312.17445*.

Ryan Liu, Jiayi Geng, Addison J Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L Griffiths. 2024. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*.

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*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

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.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*.

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*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073.

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*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577*.

Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai Wong, Heng Ji, and Kam-Fai Wong. 2025. Harnessing the reasoning economy: A survey of efficient reasoning for large language models. *arXiv preprint arXiv:2503.24377*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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*.

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 Details of Refine Action

We avoid fully replacing D_t with newly retrieved results. Doing so risks erasing valuable signals from earlier reasoning steps, which can mislead the reasoning trajectory. On the other hand, omitting retrieval altogether after a query refinement would be redundant, as the refined query is intended to improve document retrieval. By integrating retrieval directly into the REFINE action, we ensure consistency of reasoning and avoid unnecessary decomposition into separate actions.

A.2 Policy Prompt

We provide the full prompt used to guide action selection in our system, as shown in Table 5. The prompt is designed to simulate the behavior of a decision-making agent responsible for managing a search process through discrete reasoning steps.

It describes three possible actions with their input and output formats, decision conditions, and justifications. The prompt is instruction-based and free of task-specific training, enabling it to be applied across diverse retrieval scenarios without reconfiguration.

A.3 Details of Policy

LLM behavior during decoding is sensitive to temperature settings. A low temperature (e.g., $T = 0$) yields deterministic outputs, promoting stability, while higher temperatures increase diversity by allowing exploration of alternative generations.

We use the zero temperature to ensure consistency at default. However, if the model fails to produce a valid output, such as malformed JSON or empty responses, we incrementally raise the temperature by 0.1 and retry. This strategy expands the model's output space just enough to recover from failure without sacrificing control.

Importantly, our reasoning framework is designed to tolerate such exploratory decoding. Because each action operates within a strict interface and is validated at each step, the system remains robust even under increased temperature. This allows us to combine deterministic defaults with controlled exploration for improved reliability.

A.4 Implementation Details

Rank1-32B is derived from Qwen2.5-32B[1], and Rank-R1-14B is derived from Qwen2.5-14B[2]. Since the 32B version is not available in Rank-R1, we experiment with the 14B model. Since O1-Pruner is not originally designed for retrieval, we apply our prompt to perform query rewriting and document reranking using the model. O1-Pruner-32B is derived from QwQ-32B-preview[3]. To ensure a fair comparison given the model origin, we experiment with both particular non-reasoning and reasoning LLMs (Qwen2.5-32B[1], QwQ-32B[4]).

The token usage is computed by summing all generated output tokens across reasoning steps. Input tokens are excluded to isolate the generation cost. All experiments are run on a single NVIDIA A6000 GPU. We set batch size, LLM temperature, top-k retrieval, and the maximum number of reasoning steps for SMR as follows: `batch_size` = 8, `temperature` = 0.0, `k` = 10, `max_steps` = 16. These hyperparameters are held constant across all datasets and models unless otherwise stated.

A.5 Evaluation on Additional Metrics

To complement the main evaluation in terms of nDCG@10, we report additional metrics to further validate the robustness of our approach. Table 6, and Table 7 show results on MAP@10 and Recall@10, respectively, evaluated on the BRIGHT benchmark using BM25 as the base retriever. Across all metrics and datasets, SMR consistently achieves top performance compared to CoT and compressed CoT baselines.

A potential concern with our early stopping mechanism is that it may prematurely terminate the reasoning process, potentially reducing the recall of retrieved documents. However, our results show that SMR consistently achieves higher recall than all baselines. This indicates that the risk of under-retrieval is either negligible or outweighed by the benefits of our reasoning process.

A.6 Action Distributions

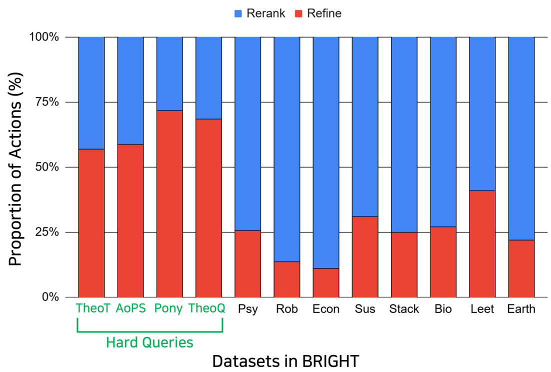

Figure 5: Distribution of reasoning actions selected by SMR on the BRIGHT benchmark. Red bars indicate the proportion of REFINE actions, and blue bars indicate RERANK actions.

To further investigate the behavior of SMR, Figure 5 presents the distribution of selected actions across different datasets in the BRIGHT benchmark. Each bar represents the relative frequency of the REFINE and RERANK actions issued during reasoning. The x-axis represents each dataset, sorted in ascending order of initial retriever performance, to examine how action distributions correlate with retrieval effectiveness.

We observe that SMR does not follow a fixed pattern or static heuristic. Instead, action selection varies across tasks, reflecting the differing reasoning needs of each dataset. In domains where the initial retrieval results are relatively informative (e.g., *Bio*, *Earth*, *Econ*), the model predominantly uses RERANK to adjust document ordering while keeping the original query intact. In contrast, in theorem and coding domains where the initial retrieval results are the lowest (green subset) such as

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

TheoT, *AoPS*, *Pony*, *TheoQ*, and they exhibit much higher proportions of REFINE actions, up to **70%** in some cases.

A.7 Ablation Study for Action Selection Policy

We conduct an ablation study comparing SMR with several fixed or naive strategies in Table 8, demonstrating the effectiveness of our prompt-based policy design. Specifically, we compare SMR with three simplified variants: one that removes the REFINE action ([RERANK or STOP]), one that removes the RERANK action ([REFINE or STOP]), and one that randomly selects an action at each step. We do not ablate STOP since it is essential for termination, but instead test whether random policy decisions degrade performance. SMR consistently achieves the best performance among these, indicating that our prompt-based policy design effectively balances refinement and reranking to support more accurate and efficient reasoning.

A.8 Evaluation of Query Intent Preservation During Refinement

To address concerns about potential query intent drift during iterative refinement, we conducted an automatic evaluation to quantify how well each rewritten query preserves the user's original intent. Specifically, we evaluated queries generated during each reasoning step of REFINE using the BRIGHT benchmark.

We employed GPT-4o-mini as a reference evaluator. For each reasoning step the model was shown the original query, the current rewritten query, and the retrieved documents used to generate. The model was instructed to assign a continuous score between 0 and 1, where 1 indicates perfect preservation of the original intent and 0 indicates complete divergence. The exact prompt used for this evaluation is included in Table 9.

Table 10 shows the average alignment score across all datasets within BRIGHT and over the full trajectory of refinement steps (up to 16 per query). The result shows that the average intent alignment score remained above **0.9** at every step. This suggests that, despite iterative rewriting, the refined queries remain closely aligned with the initial user intent. We attribute this stability to the fact that each refinement is grounded in both the immediately preceding query and its associated retrieval context, which anchors the model's generation. This evaluation provides quantitative evidence that our modular REFINE step does not introduce meaningful semantic drift, supporting the claim that our reasoning procedure maintains alignment with user goals throughout the refinement trajectory.

A.9 Token Efficiency

SMR achieves substantial reductions in inference token usage compared to all baseline methods across the full BRIGHT benchmark. Table 11 shows the total number of tokens consumed by each method across individual tasks, using BM25 as the underlying retriever.

Our method consistently outperforms Rank1, Rank-R1, and O1-Pruner by large margins. While baselines often produce long and redundant reasoning chains, SMR avoids such inefficiency through its structured state representation and early stopping mechanism. This result further supports our claim that SMR can deliver better performance with fewer computational resources.

A.10 Examples of Hard Queries

To concretely illustrate how SMR addresses the two central challenges identified in Section 1, we present an example drawn from the BRIGHT benchmark involving a complex query. This case also corresponds to the **hard queries** analyzed in Figure 4, where longer reasoning is needed to recover relevant evidence.

The query asks for the angular speed of a door after a perfectly inelastic collision with a metal ball. Although the problem is detailed and well-structured, it includes distractor terms such as *game*, *toy*, and *door*, which cause initial retrieval to return irrelevant content, including passages on chess and probability. This scenario serves as a testbed for evaluating reasoning robustness under lexical noise and semantic mismatch.

Rank1 (Redundant Reasoning) Rank1 (Table 12) generates token-level CoT traces, and the model enters a redundant loop. It repeatedly justifies why the chess passages are irrelevant, consuming over 2,500 tokens without progressing the retrieval state. Although its judgment is eventually correct, the system lacks a mechanism to recognize semantic equivalence between states and continues reasoning far beyond necessity. This exemplifies the first challenge, **redundant trajectories**, where the model revisits similar states without gaining new information.

O1-Pruner (Misguided Reasoning) O1-Pruner (Table 13) aggressively compresses the reasoning trajectory using reinforcement learning. It quickly rewrites the query into a concise fragment: “calculate angular speed after a collision using conservation of angular momentum.” While technically correct, this abstraction removes too much context from the original problem, leading the retriever to return generic physics content that fails to address the door scenario. This case illustrates the second challenge, **misguided reasoning**, where brevity is achieved at the cost of alignment with the intent of the user.

SMR (Ours) By contrast, SMR (Table 14) successfully navigates this scenario using structured state transitions. It first refines the query to include all necessary physics parameters, and then reranks documents to surface the most relevant passages. Each updated state is checked for semantic equivalence to prevent drift. Because actions are discrete and grounded in context, SMR avoids misalignment and redundant loops, achieving efficient reasoning in IR.

A.11 Usage of AI Assistants

ChatGPT was employed to enhance the clarity and grammatical accuracy of the text, offering suggestions for sentence rephrasing and correction of grammatical errors.

````
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>",
"retrieved": [
    ("<docid>", "document contents"),
    ("<docid>", "document contents"),
    ...
]
}
```

### Decision policy (check in order):

1. Query Refinement
   Choose "refine query." 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. Reranking
   Only if the query already looks good and at least one retrieved document seems on topic.

3. 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_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": "re-rank",
"reranked": ["<docid>", "<docid>", ...],
"reason": "<reason for this action>"
}
```

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

```
{
"action": "stop"
}
```
````

Table 5: Full prompt used in our system.

	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 Reasoning													
Rank1 (32B)	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
Rank-R1 (14B)	17.7	25.0	10.3	10.7	14.1	13.1	16.1	20.2	2.1	4.0	10.8	7.7	12.7
Compressed CoT Reasoning													
O1-Pruner (32B)	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
Ours													
SMR (Qwen2.5-32B)	21.8	25.6	**13.9**	14.5	**16.1**	14.4	14.3	17.8	1.1	3.6	**14.6**	**16.5**	14.5
SMR (QwQ-32B)	**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**

Table 6: Retrieval performance (MAP@10) on BRIGHT benchmark using **sparse retriever (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 Reasoning													
Rank1 (32B)	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
Rank-R1 (14B)	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
Compressed CoT Reasoning													
O1-Pruner (32B)	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
Ours													
SMR (Qwen2.5-32B)	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**
SMR (QwQ-32B)	**25.7**	31.9	18.6	21.5	20.5	24.3	**23.9**	**29.5**	4.0	**6.3**	18.7	19.0	20.3

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

	Bio	Earth	Econ	Psy	Rob	**Avg**
SMR (Qwen2.5-32B)	**28.7**	**34.9**	**20.4**	20.7	**20.9**	**25.1**
[RERANK or STOP]	23.9	33.6	17.7	15.8	19.1	22.0
[REFINE or STOP]	26.8	29.5	14.4	**20.9**	12.4	20.8
[Randomly Selecting Action]	22.6	28.8	16.4	16.1	15.0	19.8

Table 8: Ablation study of policy choices for SMR (Qwen2.5-32B), measured by nDCG@10 across the first five domains.

```
Please compare the second query aligns with the any intent of the first query and give a score from 0 to 1.
The first query is the original query, and the second query is the modified query.
The score should be based on how similar the two queries are. You must output the floating number score only
    .
The first query is: "{query_original}".
The second query is: "{query}".
```

Table 9: The prompt used to get alignment score between the original and refined queries.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alignment Score	0.98	0.97	0.95	0.95	0.95	0.94	0.94	0.93	0.93	0.93	0.92	0.92	0.92	0.92	0.91	0.91

Table 10: Average alignment scores across all datasets within BRIGHT.

	Bio	Earth	Econ	Psy	Rob	Stack	Sus	Leet	Pony	AoPS	TheoQ	TheoT	**Avg**
CoT Reasoning													
Rank1 (32B)	2,772	2,775	3,115	2,688	3,548	3,554	2,691	3,248	2,570	3,390	3,135	3,281	3,064
Rank-R1 (14B)	2,924	2,842	2,951	2,835	3,130	3,065	2,895	3,020	3,383	2,674	2,680	3,379	2,982
Compressed CoT Reasoning													
O1-Pruner (32B)	2,259	2,595	2,771	2,457	2,568	2,818	2,492	6,187	2,870	4,126	2,520	2,375	3,003
Ours													
SMR (Qwen2.5-32B)	**551**	**666**	**663**	**741**	**552**	**794**	**598**	982	**780**	1,063	914	996	**775**
SMR (QwQ-32B)	1,891	1,167	1,594	2,114	1,465	1,144	2,034	**672**	1,094	**897**	**811**	**805**	1,307

Table 11: Total number of tokens generated during inference on the BRIGHT benchmark. The lowest token counts, indicating higher inference efficiency, are marked in bold.

Iter	**Field**	**Content**
Iter 0	Query (q_0)	Imagine you're playing a game where you have a toy cannon that shoots small metal balls. You decide to aim at a lightweight door that can swing open on its hinges. The door is 1 meter wide and weighs 15 kilograms. You shoot a metal ball weighing 10 grams at a speed of 400 meters per second straight at the middle of the door, and the ball sticks to it. What is the new angular speed of the door after the ball sticks to it? (Unit: rad/s)
	Document (D_0)	$[d_1]$, $[d_2]$(about chess), $[d_3]$(about Einstein's equation) $\ldots$, $[d_k]$
Iter 1	Action	RERANK
	Query (q_1)	Imagine you're playing a game where you have a toy cannon that shoots small metal balls. You decide to aim at a lightweight door that can swing open on its hinges. The door is 1 meter wide and weighs 15 kilograms. You shoot a metal ball weighing 10 grams at a speed of 400 meters per second straight at the middle of the door, and the ball sticks to it. What is the new angular speed of the door after the ball sticks to it? (Unit: rad/s)
	Document (D_1)	$[d_1]$, $[d_k]$, ... $[d_2]$(about chess), $[d_3]$(about Einstein's equation)
	Reason	$[d_1]$: ... $[d_2]$: Okay, let's see. The user is asking whether the passage is relevant to calculating the angular speed ... **There's no overlap in subject matter here.** ... Even though both involve calculations, **the context and required knowledge are completely different.** ... **The chess passage is about combinatorics and game theory, while the query is a physics problem.** ... **They cover completely different topics with no overlap in relevant concepts.** ... Therefore, the passage isn't relevant to the query. ... $[d_3]$: ... ... *[number of tokens: 2542]*

Table 12: A qualitative example of redundant reasoning in Rank1, where the model repeats semantically equivalent steps.

Iter	Field	Content
Iter 0	Query (q_0)	Imagine you're playing a game where you have a toy cannon that shoots small metal balls. You decide to aim at a lightweight door that can swing open on its hinges. The door is 1 meter wide and weighs 15 kilograms. You shoot a metal ball weighing 10 grams at a speed of 400 meters per second straight at the middle of the door, and the ball sticks to it. What is the new angular speed of the door after the ball sticks to it? (Unit: rad/s)
	Document (D_0)	$[d_1]$, $[d_2]$(about chess), $[d_3]$(about Einstein's equation)…, $[d_k]$
Iter 1	Action	REFINE
	Query (q_1)	Calculate angular speed after a collision using conservation of angular momentum
	Reason	Alright, I've got this problem here about a toy cannon shooting a metal ball at a door, and I need to find the new angular speed of the door after the ball sticks to it. … It's about angular speed after a collision. … Specifically, it's about conservation of angular momentum. … Again, I need the angular speed once the ball sticks to the door. … Again, conservation of angular momentum governs the result. … **Perhaps phrased as "calculate angular speed after a collision using conservation of angular momentum."** … *[number of tokens: 1153]*
Iter 2	Action	RERANK
	Query (q_2)	Calculate angular speed after a collision using conservation of angular momentum
	Document (D_1)	$[d_1]$, $[d_k]$, …, $[d_2]$(about chess), $[d_3]$(about Einstein's equation)
	Reason	$[d_1]$: Relevant ... $[d_2]$: Not relevant ... $[d_3]$: Not relevant ... ... *[number of tokens: 992]* *[total number of tokens: 2143]*

Table 13: A qualitative example of misguided reasoning in O1-Pruner, where compression produces a concise but semantically misaligned query.

Iter	Field	Content
Iter 0	Query (q_0)	Imagine you're playing a game where you have a toy cannon that shoots small metal balls. You decide to aim at a lightweight door that can swing open on its hinges. The door is 1 meter wide and weighs 15 kilograms. You shoot a metal ball weighing 10 grams at a speed of 400 meters per second straight at the middle of the door, and the ball sticks to it. What is the new angular speed of the door after the ball sticks to it? (Unit: rad/s)
	Document (D_0)	$[d_1]$, $[d_2]$(about chess), $[d_3]$(about Einstein's equation) …, $[d_k]$
Iter 1	Action	REFINE
	Query (q_1)	How to calculate angular speed (rad/s) of a door after collision with metal ball using conservation of angular momentum? Door width=1m, mass=15kg; ball:mass=0.01kg, velocity=400 m/s hits center and sticks.
	Reason	The original query is missing key physics concepts like **'conservation of angular momentum' which are essential for solving the problem.** The retrieved documents include unrelated topics (e.g., Einstein's equation, chess), so refining to focus on angular momentum will improve precision. *[number of tokens: 47]*
• • •		
Iter 5	Action	RERANK
	Query (q_5)	How to calculate angular speed (rad/s) of a door after collision with metal ball using conservation of angular momentum? Door width=1m, mass=15kg; ball:mass=0.01kg, velocity=400 m/s hits center and sticks.
	Document (D_5)	$[d_1]$, $[d_k]$, …, $[d_2]$(about chess), $[d_3]$(about Einstein's equation)
	Reason	The retrieved document ($[d_1]$) include foundational content on **angular momentum, units of mass and length** relevant to the problem's parameters. Document $[d_k]$ discusses **angular velocity as a vector quantity which is essential for calculating direction-dependent quantities like angular speed.** **The other documents either discuss unrelated topics** such as chess move counts or are less directly related. … *[number of tokens: 67]*
Iter 6	Action	STOP *[total number of tokens: 252]*

Table 14: A qualitative example of SMR (QWQ-32B), demonstrating efficient reasoning in IR.