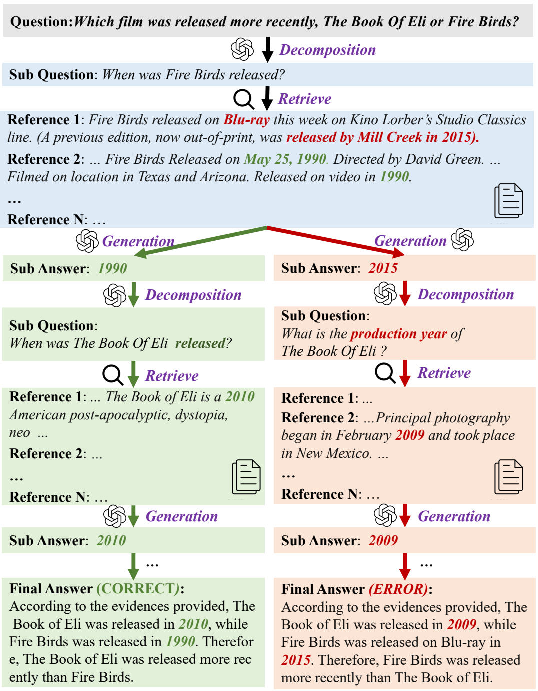
RISE: Reasoning Enhancement via Iterative Self-Exploration in Multi-hop Question Answering

Anonymous ACL submission

Abstract

Large Language Models (LLMs) excel in many areas but continue to face challenges with complex reasoning tasks, such as Multi-Hop Question Answering (MHQA). MHQA requires integrating evidence from diverse sources while managing intricate logical dependencies, often leads to errors in reasoning. Retrieval-Augmented Generation (RAG), widely employed in MHQA tasks, faces challenges in effectively filtering noisy data and retrieving all necessary evidence, thereby limiting its effectiveness in addressing MHQA challenges. To address these challenges, we propose **RISE: Reasoning Enhancement via Iterative Self-Exploration**, a novel framework designed to enhance models' reasoning capability through iterative self-exploration. Specifically, RISE involves three key steps in addressing MHQA tasks: question decomposition, retrieve-then-read, and self-critique. By leveraging continuous self-exploration, RISE identifies accurate reasoning paths, iteratively improving the model's capability to integrate evidence, maintain logical consistency, and enhance performance in MHQA tasks. Extensive experiments on multiple MHQA benchmarks demonstrate that RISE significantly improves reasoning accuracy and task performance.



Question: *Which film was released more recently, The Book Of Eli or Fire Birds?*

Sub Question: *When was Fire Birds released?*

Reference 1: *Fire Birds released on **Blu-ray** this week on Kino Lorber's Studio Classics line. (A previous edition, now out-of-print, was **released by Mill Creek in 2015).***

Reference 2: *... Fire Birds Released on **May 25, 1990**. Directed by David Green. ... Filmed on location in Texas and Arizona. Released on video in **1990**.*

...

Reference N: ...

Sub Answer: *1990*

Sub Question:
*When was The Book Of Eli **released**?*

Reference 1: ... *The Book of Eli is a **2010** American post-apocalyptic, dystopia, neo ...*

Reference 2: ...

...

Reference N: ...

Sub Answer: *2010*

Final Answer (CORRECT):
According to the evidences provided, The Book of Eli was released in *2010*, while Fire Birds was released in *1990*. Therefor e, The Book of Eli was released more rec ently than Fire Birds.

Sub Answer: *2015*

Sub Question:
*What is the **production year** of The Book Of Eli ?*

Reference 1: ...

Reference 2: *...Principal photography began in February **2009** and took place in New Mexico. ...*

...

Reference N: ...

Sub Answer: *2009*

Final Answer *(ERROR)*:
According to the evidences provided, The Book of Eli was released in ***2009***, while Fire Birds was released on Blu-ray in ***2015***. Therefore, Fire Birds was released more recently than The Book of Eli.

Figure 1: The upper part of the figure (blue) illustrates an Evidence Aggregation Error, where the Blu-ray release year of *Fire Birds* (2015) is mistaken for its theatrical release year. The lower part (green and red) shows a Reasoning Decomposition Error. The incorrect path formulates the sub-question as the production year of *The Book of Eli* (2009) instead of its release year (2010).

1 Introduction

Large language models (LLMs) demonstrate outstanding capabilities in natural language understanding and generation (Brown et al., 2020; Zhang et al., 2022; Zeng et al., 2022; Chowdhery et al., 2023; Touvron et al., 2023). However, LLMs still face challenges with complex Multi-Hop Question Answering (MHQA) tasks. MHQA requires models to integrate evidence from multiple sources and manage intricate logical relationships. This involves both retrieving and combining various pieces of evidence and constructing coherent reasoning chains. Prompt-based methods, such as Chain-of-Thought (CoT) (Wei et al., 2022b; Wang et al., 2023a; Yu et al., 2023), are employed to address MHQA by split complex problems into smaller, thereby harnessing the reasoning potential of LLMs. However, these methods often lack external knowledge, resulting in key evidence being overlooked and generate hallucinations (Rawte et al., 2023; Ji et al., 2023; Ye et al., 2023).

Retrieval-Augmented Generation (RAG) methods (Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022; Nakano et al., 2021; Asai et al., 2023; Ma et al., 2023; Yu et al., 2024; Shi et al., 2024a) have been proposed to address the aforementioned challenges. By incorporating external knowledge, RAG effectively mitigates hallucination phenomena and achieves significant results in MHQA tasks through multiple retrievals. However, RAG is constrained by the performance of the retrievers, inevitably introducing noise. Additionally, the multi-round retrieval process may lead to error propagation, resulting in two main types of errors: **Evidence Aggregation Errors** and **Reasoning Decomposition Errors**. As illustrated in Figure 1, Evidence Aggregation Errors occur when the model fails to accurately integrate evidence from multiple evidences, leading to hallucinations. Reasoning Decomposition Errors arise when problem decomposition phase generates sub-questions that do not align with original question's intent. These issues are particularly pronounced in smaller models with weaker reasoning capabilities.

Distillation and fine-tuning (Uesato et al., 2022; Luo et al., 2023; Shridhar et al., 2023) effectively enhance the reasoning capabilities of LLMs by leveraging large-scale models or high-quality, manually annotated data to improve performance. However, biases brought by human subjective annotations may undermine the performance of fine-tuning (Casper et al., 2023; Lightman et al., 2023), and these methods are costly, requiring substantial human or computational resources. Meanwhile, self-iteration methods (Yuan et al., 2024; Wang et al., 2024; Madaan et al., 2024) demonstrate tremendous potential in complex reasoning tasks. Unlike approaches that depend on large-scale models and manual annotations, self-iteration methods enable models to generate and learn from their own data, achieving outstanding results in complex tasks such as code generation and intelligent agents (Jiang et al., 2023; Ni et al., 2024; Qiao et al., 2024). Nevertheless, research on combination self-iteration methods with RAG remains limited. The integration of these two approaches has the potential to improve performance in complex reasoning tasks and leads to cost reduction.

In this paper, we introduce an innovative framework, **RISE** (**R**easoning Enhancement via **I**terative **S**elf-**E**xploration), which combines the paradigms of RAG and self-iteration to address key challenges in MHQA tasks. Specifically, RISE defines three core actions: question decomposition, retrieve-then-read, and self-critique. By repeatedly executing these actions, the model autonomously explores accurate reasoning paths for problems. During this process, RISE accumulates experience datasets for the three actions and updates the model based on this experience. Through multiple iterations, RISE significantly enhances the model's reasoning capabilities in MHQA tasks. Experimental results demonstrate that RISE outperforms baseline methods on several MHQA benchmark datasets, strongly validating its effectiveness in solving MHQA tasks while offering lower usage costs. Our main contributions are as follows:

- We propose RISE, which combines RAG and self-iteration to address two key challenges in MHQA tasks: Evidence Aggregation Errors and Reasoning Decomposition Errors.
- We design self-exploration mechanism, converts MHQA in RAG into multi-objective optimization problem, thus improving model's reasoning capability and reducing costs.
- We integrate self-iteration paradigm with RAG, bridging gap in applying self-iteration strategies within MHQA RAG framework.

2 Methods

2.1 Overview

In this section, we provide a detailed and comprehensive description of **RISE**. Traditional RAG frameworks typically rely on manual interventions or guidance from more advanced models to enhance model capabilities. In contrast, the RISE framework aims to fully exploit the model's intrinsic potential, enabling iterative self-exploration to achieve continuous capability improvement.

As illustrated in Figure 2, RISE begins with a dataset Q^i containing multi-hop questions as input. The model M^i performs self-exploration for each question q. This process is driven by the model's inherent capabilities and involves iterative execution of three core operations: question decomposition, retrieve-then-read, and self-critique. Through these operations, the model progressively explores answers to the given questions. Each result generated during this process is stored as historical experience $\mathcal{D}$. The detailed description of the self-exploration mechanism is provided in Section 2.2.

Upon completing the exploration of all questions, the accumulated independent yet interrelated experiences are used to synchronously optimize the three core capabilities of M^i, resulting in an enhanced model M^{i+1}. Subsequently, M^{i+1} performs question expansion based on seed question,

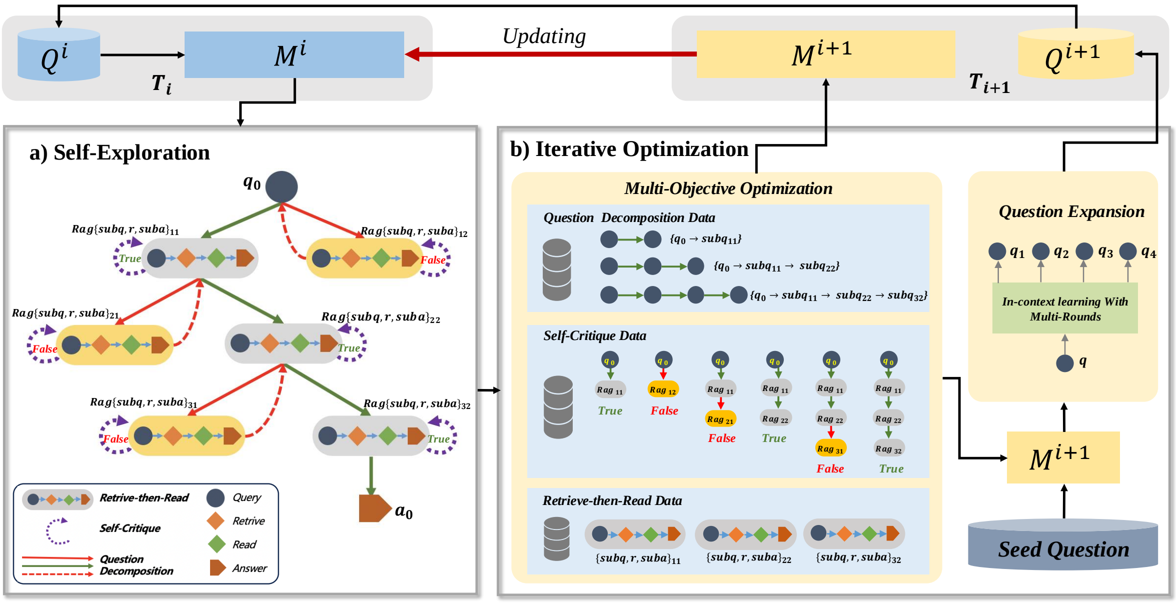


Figure 2: Overview of the RISE Framework. a) Self-Exploration: Model M^i decomposes complex questions q_0 into simpler sub-questions, generates sub-answers using retrieve-then-read, and evaluates the validity of each sub-question and sub-answer pair, culminating in a final answer a_0. Interactions are stored as historical data $\mathcal{D}$. b) Iterative Optimization: RISE optimizes M^i with $\mathcal{D}$, create an enhanced model M^{i+1}, which generate new questions Q^{i+1} for the next self-exploration cycle, iteratively improves model performance.

generating a new dataset Q^{i+1}, which serves as input to initiate a new round of self-exploration.

2.2 Self-Exploration Mechanism

The self-exploration mechanism constitutes the core of our framework, enabling the model to address complex problems through iterative reasoning. This mechanism encompasses three fundamental tasks: question decomposition, retrieve-then-read, and self-critique, which collectively form a structured pathway for exploration, as illustrated in Algorithm 1. The related prompts can be found in the Appendix A.1.1. By facilitating a fine-grained exploration process, the self-exploration mechanism systematically enhances the model's capability to handle complex problem-solving tasks.

Question Decomposition. Prior works(Press et al., 2023a; Li et al., 2024) have shown that models can iteratively generate simple sub-questions to solve complex questions. In this task, model incrementally decomposes the initial complex question into finer-grained sub-questions. Specifically, at the t-th exploration node, the model utilizes previously explored sub-questions and answers as historical information, denoted as $\mathcal{H} = \{(subq_1, suba_1), \cdots, (subq_{t-1}, suba_{t-1})\}$. The original question q_0 is combined with $\mathcal{H}$ and input into model M to generate next sub-question. If model determines that historical information is sufficient to answer the original question, it generates the final answer, marking the end of the exploration. Formally, this process is represented as Formula 1:

$$subq_t = \mathcal{F}_d(M, \mathcal{H}, q_0) \tag{1}$$

$$a_0 = M(q_0, \mathcal{H}), \quad \text{if } \mathcal{H} \text{ is sufficient.} \tag{2}$$

Additionally, all decomposition steps, including the original question and generated sub-questions, are recorded to form the dataset $\mathcal{D}_\text{d} = \left\{\{q_0, \mathcal{H}, subq\}_{i=1}^{n_\text{p}}\right\}^{N_\text{q}}$. By leveraging this fine-grained and structured dataset, the model learns the logical dependencies and relationships between questions and sub-questions, thereby improving its ability to decompose complex problems.

Retrieve-then-Read. This task adopts a standard RAG paradigm to provide evidence-based answers for generated sub-questions. At t-th exploration node, we utilize a retriever to obtain relevant fragments r_t based on sub-question. M then generates the corresponding answer with retrieved evidence:

$$suba_t = \mathcal{F}_g(M, subq_t^i, r_t) \tag{3}$$

Each sub-question and its answer form an exploration node $(subq_i, suba_i)$, which is added to the historical information $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{(subq_i, suba_i)\}$. All exploration nodes are recorded to construct the dataset: $\mathcal{D}_{\mathrm{r}} = \left\{\{subq, r, suba\}_{i=1}^{n_{\mathrm{p}}}\right\}^{N_{\mathrm{q}}}$. Training on this dataset enables the model to effectively integrate evidence into the reasoning process, improving the accuracy of generated answers and the reliability of the exploration process.

Self-Critique. In this task, the model's critique capability is incorporated into the exploration process, where it critiques each exploration node through a binary assessment. Specifically, after completing the question decomposition and retrieve-then-read tasks at the t-th exploration node, the model M critiques the relevance and utility of the node for solving the original question and outputs a binary decision. If the node is critiqued as True, it is retained, and the exploration proceeds to the next step. If critiqued as False, the node is temporarily stored, and the process reverts to the preceding valid node to generate a new node. Formally, this process is represented as Formula 4:

$$\sigma_t = \mathcal{F}_{\mathrm{c}}(M, subq_t, suba_t), \quad \sigma_t \in \{0, 1\} \tag{4}$$

By recording these critiques, the dataset $\mathcal{D}_c$ is constructed: $\mathcal{D}c = \left\{\{\langle subq, suba\rangle, \sigma\}_{i=1}^{n_{\mathrm{p}}}\right\}^{N_{\mathrm{q}}}$. This dataset is designed to enhance the model's self-critique capabilities, ensuring logical consistency and relevance within the exploration path.

2.3 Iterative Optimization

Multi-Objective Optimization. In Section 2.2, the three datasets, $\mathcal{D}_d$, $\mathcal{D}_r$, and $\mathcal{D}_c$, are interconnected and mutually influential. To reflect this interdependence in the model's capabilities, we employ a multi-objective optimization approach to train our model. The multi-objective optimization approach integrates the objectives of different training tasks into a unified objective. We posit that stronger decomposition capabilities can enable the model to generate more precise questions, thereby improving the accuracy of the generation task. Additionally, enhanced critique capabilities can assist the model in decomposing more relevant sub-questions. To achieve this, we defined three loss functions corresponding to the three tasks and integrated them into a unified objective. The overall loss function is formulated as Formula 5:

Algorithm 1 Self-Exploration Mechanism

```
Require: Model $M$, Retriever $R$
Initialize: History $\mathcal{H}$ = null
Input: Original question $q_0$
while Additional information $\mathcal{H}$ is needed to answer $q_0$ do
    $M$ generates a sub-question $subq$ based on the current $\mathcal{H}$
    $R$ retrieves relevant references $r$ from external knowledge using $subq$
    $M$ generates a sub-answer $suba$ for $subq$ using references $r$
    $M$ critiques the pair $(subq, suba)$ to produce a confidence score $\sigma$
    if $\sigma$ == 1 then
        Add $(subq, suba)$ to $\mathcal{H}$
    end if
end while
$M$ generates the final answer $a_0$ based on the accumulated $\mathcal{H}$
Output: Final answer $a_0$, History $\mathcal{H}$
```

$$\mathcal{L} = \alpha\mathcal{L}_d + \beta\mathcal{L}_r + \gamma\mathcal{L}_c \tag{5}$$

where α, β, and γ are the weights for the three tasks. Through experimentation, we observed that different datasets exhibit varying levels of dependency on the three capabilities. By adjusting the loss weights, the model's reliance on each capability can be fine-tuned accordingly.

Question Expansion and Iteration. After completing the multi-objective optimization, we use data from the previous questions as seed data for M^{i+1} to perform question expansion, enabling the acquisition of more training data for the next iteration. The question expansion method is inspired by (Wang et al., 2023c), which employs multiple rounds of in-context learning to enhance the diversity of the training dataset.

To prevent the model from experiencing catastrophic forgetting, we record the decomposition task datasets $\mathcal{D}_{\mathrm{d}}$ after each iteration. During subsequent iterations, these datasets are incorporated as review data $\mathcal{D}_{review}$ to reinforce the model's reasoning ability for complex problems. In summary, the formula for each iteration is Formula 6:

$$M^{i+1} = \mathcal{F}_{update}\left(M^i, \mathcal{D}_{\mathrm{d}}^i, \mathcal{D}_{\mathrm{r}}^i, \mathcal{D}_{\mathrm{c}}^i, \mathcal{D}_{\text{review}}\right) \tag{6}$$

3 Experiments Setup

Datasets: For the main experiments, we use three QA datasets: 2WikiMultiHopQA (2WIKI) (Ho et al., 2020), HotpotQA (Hotpot) (Yang et al., 2018), and MuSiQue (MSQ) (Trivedi et al., 2022), which provide diverse reasoning challenges to evaluate the robustness of our framework. Additionally, for the analysis experiments, we include Natural Questions (NQ) (Kwiatkowski et al., 2019), Web Questions (WebQ) (Berant et al., 2013) and TriviaQA (Joshi et al., 2017) to assess the model's performance on open-domain Question Answering tasks, further extending the evaluation scope.

Models and Methods: In our experiments, we use LLaMA-3.1-8B (Dubey et al., 2024) as the base model for our method in main experiments. Similarly, most of the reproduced methods are also implemented using LLaMA-3.1-8B. Additionally, based on the characteristics of MHQA tasks, we select and reproduce a variety of methods, categorized into non-retrieval-based methods and retrieval-based methods. Non-retrieval-based methods include Naive LLM (LLaMA-3.1-8B, GPT-4-turbo, GPT-3.5-turbo), CoT (Wei et al., 2022b), CoT-SC (Wang et al., 2023a) and GenRead (Yu et al., 2023), while the retrieval-based methods consist of Naive RAG, Self-Ask (Press et al., 2023b), WebGLM (Liu et al., 2023), Self-RAG (Asai et al., 2023), RRR (Ma et al., 2023), and GenGround (Shi et al., 2024a). In the analysis experiments, we employ GPT-4o[1] as the evaluation model, combining subjective analysis with specific metrics to comprehensively assess model performance.

Retrieval: We adopt a two-stage retrieval framework (Liu et al., 2023), consisting of coarse-grained web search (via Chrome) followed by fine-grained LLM-enhanced retrieval. We consistently use the same retrieval method to reproduce results for other approaches that incorporate retrievers.

Evaluation Metrics: We assess performance using Accuracy (Acc), F1 score (F1), and Exact Match (EM) to evaluate QA quality. Additionally, we evaluate the quality of the reasoning chains from the perspectives of chain length and four subjective dimensions: conciseness, rationality, sequencing, and goal orientation.

We provide comprehensive experimental details in Appendix A.2, including implementation details, datasets, and other relevant information.

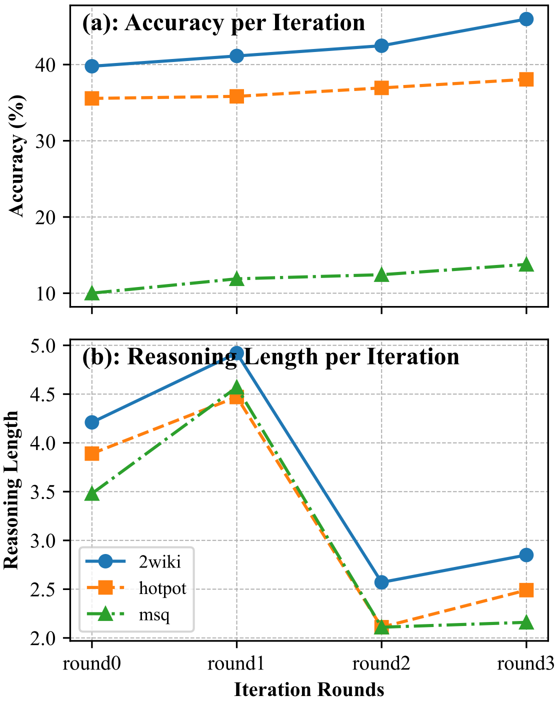


Figure 3: Figure illustrates the changes in model accuracy (a) and reasoning length (b) across multiple datasets after three iterations. Accuracy consistently improves across all datasets, while reasoning length, despite some fluctuations, shows an overall decreasing trend. Notably, the average reasoning length remains below 3, with accuracy continuing to improve, demonstrating the effectiveness of the RISE.

4 Results and Analysis

In this section, we evaluate RISE from three aspects. First, we validate effectiveness of multiround self-iterative and compare RISE with mainstream MHQA methods. Second, we conduct an in-depth analysis of the performance of question decomposition, retrieve-then-read, and self-critique using objective metrics and AI-based evaluations. Finally, we conduct ablation studies to verify the importance of different tasks in enhancing performance.

4.1 Overall Performance

RISE Outperforms Other Methods: Table 1 presents the experimental results across three MHQA datasets. We observe that retrieval enhancement is crucial for MHQA tasks, without external knowledge support, non-retrieval methods generally achieve lower accuracy compared to most RAG approaches under the same model. RISE outperforms other methods on most datasets, even surpassing CoT-SC and RRR with GPT-3.5. Notably, RISE achieves 45.97% accuracy on 2WIKI, demon-

[1] We use GPT models accessed via the OpenAI API: `https://openai.com/api/`.

Method	Model	2WIKI			HotpotQA			MuSiQue		
		Acc	F1	EM	Acc	F1	EM	Acc	F1	EM
Without Retrieval										
Naive LLM	LLaMA-3.1-8B	32.26	7.13	0.00	26.94	5.28	0.28	7.30	2.01	0.00
	GPT-3.5-turbo-0125	44.62	12.88	0.54	41.67	17.50	5.28	15.68	6.52	0.00
	GPT-4-turbo-2024-0409	58.87	7.40	0.00	53.61	7.24	0.00	30.00	3.73	0.00
CoT (Wei et al., 2022b)	LLaMA-3.1-8B	35.75	2.74	0.00	37.78	2.39	0.00	15.14	1.39	0.00
CoT-SC* (Wang et al., 2023a)	GPT-3.5-turbo-0125	20.97	24.31	17.20	30.56	39.59	29.72	8.92	15.36	7.30
GenRead (Yu et al., 2023)	LLaMA-3.1-8B	15.59	17.22	12.10	28.06	34.81	25.83	6.22	10.38	4.32
With Retrieval										
Naive RAG	LLaMA-3.1-8B	37.90	5.62	0.27	37.50	12.42	4.44	11.89	3.26	0.54
Self-Ask (Press et al., 2023b)	LLaMA-3.1-8B	21.77	23.58	15.59	31.39	38.45	26.11	10.00	15.78	6.76
WebGLM (Liu et al., 2023)	LLaMA-3.1-8B	38.17	9.37	0.00	38.05	7.78	0.00	10.27	3.04	0.00
Self-RAG* (Asai et al., 2023)	LLaMA2-7B	31.99	18.55	7.80	32.22	23.93	9.72	8.65	6.79	0.81
	LLaMA2-13B	29.03	19.15	9.14	31.11	21.47	6.94	9.19	7.16	1.08
RRR (Ma et al., 2023)	LLaMA-3.1-8B	20.43	4.80	0.00	7.50	2.38	0.00	0.27	1.32	0.00
	GPT-3.5-turbo-0125	28.23	13.70	3.23	29.72	16.76	2.22	8.65	6.08	0.27
GenGround (Shi et al., 2024a)	LLaMA-3.1-8B	38.98	36.30	26.08	33.61	35.54	23.33	8.11	10.94	5.41
RISE (Ours)	LLaMA-3.1-8B	**45.97**	**41.15**	**31.18**	**38.06**	**40.47**	**28.06**	**13.78**	**15.46**	**7.30**

Table 1: Comparison of RISE with other methods on the 2WikiMultiHopQA, HotpotQA, and MuSiQue, including no-retrieval and retrieval-based approaches. Methods marked with asterisk (*) involve specific considerations: CoT-SC uses GPT-3.5 due to LLaMA-3.1's limitations in adhering to instructions, and Self-RAG employs publicly released model weights because its dataset is unavailable. All other methods are reproduced with LLaMA-3.1-8B.

strating performance comparable to GPT-3.5-turbo. Furthermore, our method excels in F1 and EM metrics, demonstrating both accuracy and efficiency. In summary, RISE performs exceptionally well in MHQA tasks.

Steady Performance Improvement: Meanwhile, as shown in Figure 3 *(a) Accuracy per Iteration*, we illustrate how the model's accuracy evolves over three iterations on multiple datasets. The results demonstrate a consistent upward trend in accuracy with each iteration, further validating the effectiveness of our proposed self-training method in improving the model's overall performance.

4.2 Analysis Experiments

Question Decomposition Capability: To evaluate improvement in the model's decomposition capability for MHQA tasks, we analyze the changes in reasoning length. As shown in Figure 3 *(b) Reasoning Length per Iteration*, accuracy steadily improves, while reasoning length initially increases and then decreases, ultimately showing downward trend. This trend reflects model's decomposition ability progressively improves over iterations.

To further analyze changes in decomposition ability, we conduct second set of experiments using GPT-4o as a judge to evaluate the model's query decomposition across four dimensions (including conciseness, rationality, sequencing and goal orientation, see Appendix A.1.2 for more details.). As illustrated in Figure 5, we compare the performance of the model across iterations and observe newer model consistently outperforms the previous iteration. These findings demonstrate that self-training not only improves reasoning paths but also significantly enhances the rationality of decomposition.

Retrieve-then-Read Capability: In MHQA tasks, models often struggle to integrate logical information from extensive evidence, especially in filtering irrelevant content. To evaluate the changes in the model's summarization capability over iterations, we disable the decomposition functionality and instead allow the model to perform single-round retrieval and direct question-answering. To ensure robustness in the experiments, we introduce relatively simpler datasets such as NQ, WebQ, and TriviaQA (Figure 5 *(a) Simple Questions*) while retaining the more complex datasets from the main experiments (Figure 5 *(b) Complex Questions*). The experimental results show that, as iterations progress, RISE consistently improves its performance across six datasets, including simple and complex tasks. This demonstrates the significant advantage of RISE in MHQA tasks and its effectiveness in conventional QA tasks, further validating its generalizability.

Self-Critique Capability: To evaluate the changes in the model's self-critique capability, we designed a third set of experiments. In this experiment, both our model and GPT-4o assess the same set of decomposition results, with GPT-4o serving as a reference. By analyzing the consistency between our model and GPT-4o evaluations, we measure

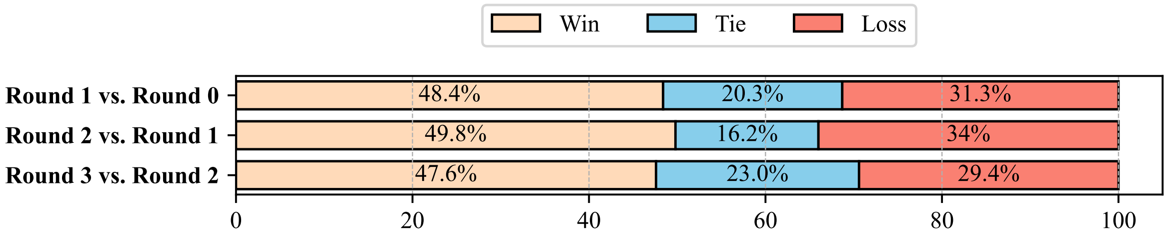


Figure 4: Evaluating the win rates between the current and previous iterations using GPT-4o to assess model's question decomposition capability. Results indicate that each new iteration consistently outperforms the previous one in subjective effectiveness, demonstrating RISE's continuously enhance the model's decomposition capability.

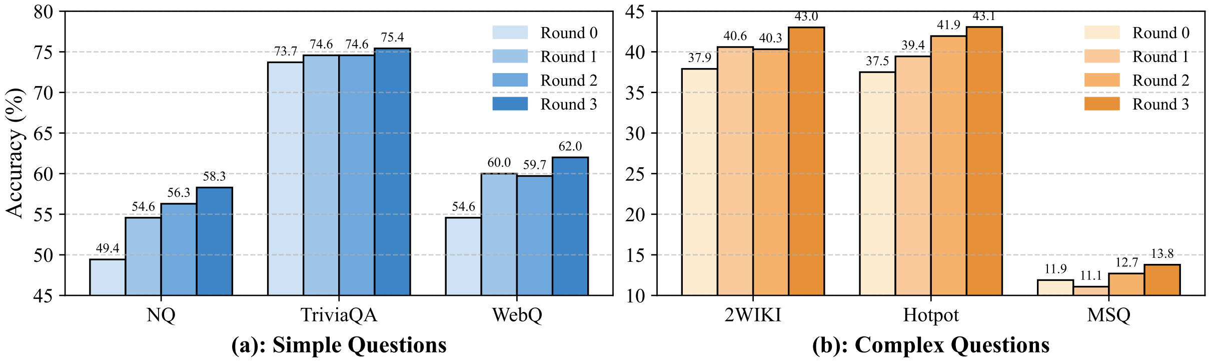


Figure 5: Changes in the model's retrieve-then-read capability. (a) Results on simpler datasets (NQ, TriviaQA, WebQ), (b) Results on more complex datasets (2Wiki, HotpotQA, MSQ), where accuracy shows consistent growth with each iteration, even in challenging scenarios.

the improvement in the model's self-critique capability. As shown in Table 2, the consistency between our model and GPT-4o steadily increases with each iteration. This indicates that the iterative process in RISE effectively enhances the model's self-criticism capability. (For more experiment details see Appendix A.2.3.)

4.3 Ablation Study

To evaluate the impact of each synthesized training dataset on the model's performance, we conduct an ablation study. As shown in Table 3, the experiment uses the same three MHQA datasets as before and the three training datasets generated in the round1, with accuracy as the primary evaluation metric.

Removing the question decomposition dataset leads to accuracy drop of 3.5% on 2Wiki, highlighting its importance in enabling effective multi-hop reasoning. Excluding the retrieve-then-read dataset causes accuracy declines on HotpotQA (2.77%) and Musique (2.43%), highlighting the importance of this dataset in synthesizing evidence from diverse sources to mitigate the impact of noise. Furthermore, the removal of the self-critique dataset results in consistent accuracy reductions across all three datasets, emphasizing its pivotal function in refining reasoning paths processes. These results demonstrate the complementary and indispensable contributions of the question decomposition, retrieve-then-read, and self-critique datasets to the model's overall performance.

5 Related Works

Multi-hop Question Answering: MHQA tasks address questions that require integrating information from multiple sources and performing multi-step reasoning to produce a complete answer (Zhang et al., 2024; Li and Du, 2023). Question decomposition has been a pivotal approach for understanding and solving multi-hop questions, some works (Wei et al., 2022a; Wang et al., 2023b; Zhou

	Consistency with GPT-4o (%)		
	2WIKI	HotpotQA	MSQ
Round 1	74.30	64.70	60.00
Round 2	72.67	66.30	76.00
Round 3	**79.67**	**77.33**	**79.33**

Table 2: Consistency analysis with GPT-4o across three rounds training on datasets (2WIKI, HotpotQA, and MSQ). The results show progressive improvements in consistency with GPT-4o, highlighting the model's enhanced self-critique ability through iterative training.

	2WIKI	Hotpot	MSQ
	Acc	Acc	Acc
w/o Decomp	37.63	33.89	11.08
w/o R-t-R	40.59	33.06	9.46
w/o Critique	38.98	33.89	10.27
RISE	**41.13**	**35.83**	**11.89**

Table 3: Ablation study results on 2WIKI, HotpotQA, and MSQ, showing the impact of removing Question Decomposition (w/o Decomp), Retrieve-then-Read (w/o R-t-R), and Self-Critique (w/o Critique) datasets.

et al., 2023; Shi et al., 2024b) leverage LLMs to divide complex questions into simpler single-hop sub-questions that are solved sequentially. Self-Ask (Press et al., 2023b) uses LLMs to generate and resolve follow-up sub-questions with an external search engine. However, the effectiveness of these approaches depends significantly on LLM's inherent question decomposition capabilities, and is further constrained by hallucinations.

Retrieval-Augmented Generation for MHQA: RAG (Guu et al., 2020; Lewis et al., 2020; Izacard et al., 2022; Nakano et al., 2021; Asai et al., 2023; Ma et al., 2023; Yu et al., 2024; Shi et al., 2024a) integrates retrieval with generation to solve knowledge-intensive tasks (Zhu et al., 2024; Feng et al., 2024). The original RAG framework excels at single-hop QA but faces significant challenges in handling multi-hop QA and complex reasoning tasks (Lewis et al., 2020; Xu et al., 2024).

To address these challenges, various methods have been proposed. Chain of Thought (CoT) (Wei et al., 2022b) and Tree of Thought (ToT) (Yao et al., 2024) are integrated with RAG to enable multi-step reasoning and iterative retrieval (Press et al., 2023b; Yao et al., 2023; Zhou et al., 2023; Khattab et al., 2023), allowing the model to incorporate a broader range of external knowledge and improve its reasoning capabilities. However, existing retrieval-augmented systems are inevitably affected by the limitations of retrievers, often introducing irrelevant or noisy information (Yin et al., 2023; Xu et al., 2024; Ma et al., 2023). Enhancing the model's reasoning capabilities to filter noise and focus on critical evidence is essential for accurate summaries, which our method achieves through reasoning decomposition, improving both logical reasoning and QA performance.

Self-Improvement in Large Language Models: Self-improvement refers to the process by which models generate and utilize their own output data to enhance performance (Zelikman et al., 2024; Singh et al., 2024; Gülçehre et al., 2023). Existing approaches, such as self-training (Du et al., 2021) and self-play (Yuan et al., 2024; Chen et al., 2024), leverage pseudo-label generation and iterative policy optimization to improve the utilization of unlabeled data and enhance decision-making capabilities. Self-Rewarding (Yuan et al., 2024) employs the LLM-as-Judge paradigm to strengthen reasoning abilities, while Self-Refine (Madaan et al., 2024) iteratively optimizes generated outputs through self-feedback mechanisms.

In complex tasks like code generation and agent-based learning, self-improvement proves effective. Methods such as Self-Evolve (Jiang et al., 2023), NExT (Ni et al., 2024), and AutoAct (Qiao et al., 2024) leverage self-feedback, self-guided tracking, and self-planning to enhance performance. However, the application of self-iterative techniques in RAG scenarios remains underexplored. Our method addresses this gap by integrating self-exploration into RAG to generate diverse training data, enabling continuous model evolution and enhancing performance in complex tasks.

6 Conclusion

We propose RISE, a framework that addresses two key errors in MHQA tasks: Evidence Aggregation and Reasoning Decomposition. Through self-exploration, RISE continuously enhances reasoning capabilities. Additionally, RISE integrates self-iterative paradigm with RAG framework, bridging the gap in applying self-iterative strategies to MHQA scenarios without requiring manual intervention or reliance on large models, thereby offering a cost-effective solution. Experimental results on MHQA benchmarks demonstrate significant improvements in reasoning accuracy and task performance, highlighting RISE's robustness and adaptability in tackling complex reasoning challenges.

Limitation

While RISE achieves strong performance in complex reasoning tasks, there remain opportunities for further enhancement. The current framework relies on external retrieval mechanisms without explicit optimization, which may limit the quality of evidence for downstream reasoning. Future work could explore self-improvement across the entire pipeline—spanning question decomposition, retrieval, generation, and reflection—to achieve more seamless integration and efficiency.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. In *Forty-first International Conference on Machine Learning*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2024. Retrieval-generation synergy augmented large language models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11661–11665.

Çaglar Gülçehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *CoRR*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Shuyang Jiang, Yuhao Wang, and Yu Wang. 2023. Self-evolve: A code evolution framework via large language models. *Preprint*, arXiv:2306.02907.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *Preprint*, arXiv:2212.14024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6779–6789, Singapore. Association for Computational Linguistics.

Xiang Li, Shizhu He, Fangyu Lei, JunYang JunYang, Tianhuang Su, Kang Liu, and Jun Zhao. 2024. Teaching small language models to reason for knowledge-intensive multi-hop question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7804–7816, Bangkok, Thailand. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4549–4560.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *Preprint*, arXiv:2308.09583.

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, Singapore. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ansong Ni, Miltiadis Allamanis, Arman Cohan, Yinlin Deng, Kensen Shi, Charles Sutton, and Pengcheng Yin. 2024. Next: Teaching large language models to reason about code execution. In *Forty-first International Conference on Machine Learning*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023a. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023b. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Huajun Chen, et al. 2024. Autoact: Automatic agent learning from scratch for qa via self-planning. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024a. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024b. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353, Bangkok, Thailand. Association for Computational Linguistics.

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

Avi Singh, John D Co-Reyes, and Rishabh Agarwal. 2024. Beyond human data: Scaling self-training for problem-solving with language models. In *ICLR*

2024 Workshop on Navigating and Addressing Data Problems for Foundation Models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *Preprint*, arXiv:2211.14275.

Tianduo Wang, Shichen Li, and Wei Lu. 2024. Self-training with direct preference optimization improves chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11917–11928.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022a. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1362–1373, New York, NY, USA. Association for Computing Machinery.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

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

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. ALCUNA: Large language models meet new knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1397–1414, Singapore. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, S Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference on Learning Representations*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Eric Zelikman, Eliana Lorch, Lester Mackey, and Adam Tauman Kalai. 2024. Self-taught optimizer (stop): Recursively self-improving code generation. In *OPT 2023: Optimization for Machine Learning*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1718–1731, Mexico City, Mexico. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2024. Large language models for information retrieval: A survey. *Preprint*, arXiv:2308.07107.

A Appendix

A.1 Prompts

A.1.1 Self-Exploration Prompts

We designed detailed prompts for the three tasks in the self-exploration phase: question decomposition 6, retrieve-then-read 7, and self-critique 8. The examples used in the decomposition prompt are inspired by self-ask (Press et al., 2023b).

A.1.2 Self-Decomposition Evaluation Prompt

In this paper, the evaluation of the question decomposition capability is conducted using GPT-4o with prompt as shown in Figure 9. The analysis involves assessing and scoring the decomposition results of different iterations across multiple dimensions, ultimately leading to a comparative analysis of the two models. The dimensions of the analysis include:

- **Conciseness**: Whether the decomposition avoids redundancy while ensuring comprehensiveness.
- **Rationality**: Whether the decomposed subproblems are closely related to the original problem.
- **Sequencing**: Whether the decomposition of sub-problems follows a logical order and facilitates the problem-solving process.
- **Goal Orientation**: Whether the decomposition is clearly centered around addressing the main problem's objective. Are the subproblems closely aligned with the core goal of the main problem? Does it avoid redundant issues that deviate from the primary objective?

A.2 Experiment detail

A.2.1 Implementation Details

We conduct all experiments on a server equipped with four NVIDIA A800 80G GPUs. For the experimental setup, we use the following hyperparameters: learning rate of 1×10^{-4}, batch size of 64,and cut-off length of 8192. Furthermore, for the weighting parameters α, β, andγ in the overall loss function, values of 1 were uniformly adopted in this research.

A.2.2 Datasets

The cold-start dataset Q^0 consists of 800 randomly sampled instances from the training sets of 2WikiMultiHopQA, HotpotQA, and MuSiQue, totaling 2,400 cold-start samples. Table 4 provides detailed information on the training datasets constructed during each round of self-exploration.The evaluation datasets comprise 372 examples from 2WikiMultiHopQA, 360 examples from HotpotQA, and 370 examples from MuSiQue.

Datasets	$\mathcal{D}_d$	$\mathcal{D}_r$	$\mathcal{D}_c$
Round1	3276	2501	3925
Round2	8309	6311	8074
Round3	4858	2106	2312

Table 4: Number of samples in datasets $\mathcal{D}_d$, $\mathcal{D}_r$, and $\mathcal{D}_c$ for each training iteration round.

	2WIKI	HotpotQA	MSQ
Round1	223	194	180
Round2	218	199	228
Round3	239	232	238
Total	300	300	300

Table 5: Number of instances in each round's self-critique capability evaluation that aligned with GPT-4o

A.2.3 Self-Critique Capability Experiments Details

To demonstrate the improvement in the self-critique capability of the model across iterations, we sampled 300 instances from the generated $\mathcal{D}_c$ at each round and compared them with GPT-4o. The responses from GPT-4o were used as ground truth to calculate the self-critique accuracy of our model. In Table 5, we present the number of instances in each round's self-critique capability evaluation that aligned with GPT-4o.

Decomposition Prompt

Instruction: Please answer the following questions according to the given format. Strictly follow each format specification, as this will ensure consistency and clarity in your response.

- Only add follow-up questions if additional details are needed to arrive at the final answer.
- For each follow-up question, use exactly this format: ’Follow up: question’
- Ensure each follow-up question is direct and structured to be easily searchable, focusing on key information for efficient search engine retrieval.
- For each answer to a follow-up question, use exactly this format: ’Intermediate answer: answer’
- Do not repeat or alter any previously generated follow-up questions or intermediate answers.
- Conclude with the final answer using this exact format: ’So the final answer is: final answer’ if no further questions are needed.

Use the examples below to understand the expected structure, and follow this format without deviating from these instructions.

Question: Who lived longer, Muhammad Ali or Alan Turing?
Are follow up questions needed here: Yes.
Follow up: How old was Muhammad Ali when he died?
Intermediate answer: Muhammad Ali was 74 years old when he died.
Follow up: How old was Alan Turing when he died?
Intermediate answer: Alan Turing was 41 years old when he died.
So the final answer is: Muhammad Ali.

Question: When was the founder of craigslist born?
Are follow up questions needed here: Yes.
Follow up: Who was the founder of craigslist?
Intermediate answer: Craigslist was founded by Craig Newmark.
Follow up: When was Craig Newmark born?
Intermediate answer: Craig Newmark was born on December 6, 1952.
So the final answer is: December 6, 1952.
... ...
—
Now, **continue the response** using the following question and information provided below.
Only add follow-up questions if necessary to reach the final answer.
Ensure all follow-up questions are optimized for search engine queries, making each question concise, direct, and easily searchable. Avoid modifying or repeating any existing content.
—

Question (ORIGINAL): {question}
Are follow up questions needed here: Yes.

Figure 6: Decomposition prompt template.

Generation Prompt

#Question-Answering-in-Reference-Task#

Instruction:
- Use the references provided to answer the question as specifically and completely as possible.
- If the references do not directly answer the question, combine relevant information from multiple references to create a well-supported answer.
- When references are Null or insufficient, use your own knowledge to provide a clear and relevant answer.
- When a direct answer cannot be determined, list any information in the references that could be relevant or provide partial insights related to the question. Avoid responses such as 'I don't know' or 'more information is needed.'
- Always prioritize specificity and relevance in your answer, providing helpful context or details that approach a complete answer.

Reference [1]
Reference [2]
...

Question: {question}

Figure 7: Generation prompt template.

Self-Critique Prompt

Main Question: {question}
Below is a list of previously generated subquestions and their intermediate answers, created as part of a multi-step reasoning process to answer the main question.
Your task is to evaluate whether the information in the current subquestion is necessary and contributes incrementally towards solving the main question.

Previously generated subquestions and answers:
{previous subquestions}

Current subquestion and answer candidate:
{subquestion and intermediate answer }

Instruction:
- Step 1: Check for Redundancy. Check if the current subquestion or answer repeats information already provided in previous subquestions. If it does, return 'flag = False' as this information is redundant.
- Step 2: Assess Relevance. If the information is not a duplicate, analyze its relevance to the main question. Determine whether it provides new, relevant information that helps move closer to solving the main question, even if it only provides indirect context or background.
Note that information does not need to directly answer the main question to be considered relevant; it can also support understanding or provide necessary context. Mark it as 'flag = True'.
- Step 3: Based on your analysis, provide a final judgment in the following format:

Final Judgment: [flag = True or flag = False]

Examples:

Main Question: "Who lived longer, Muhammad Ali or Alan Turing?"
• Follow up: "How old was Muhammad Ali when he died?" (Flag = True, relevant for lifespan comparison.)
• Follow up: "How old was Alan Turing when he died?" (Flag = True, completes lifespan comparison.)
• Redundant Example: "How old was Muhammad Ali when he passed?" (Flag = False, redundant with earlier subquestion.)
Main Question: "Are both the directors of Jaws and Casino Royale from the same country?"
• Follow up: "Who directed Jaws?" (Flag = True, needed for director identification.)
• Follow up: "Where is Steven Spielberg from?" (Flag = True, relevant to nationality check.)
• Irrelevant Example: "What is Steven Spielberg's favorite genre?" (Flag = False, not relevant to nationality.)

Reminder: Use "flag = True" for any subquestion that provides useful information or context toward solving the main question, even if indirectly. Set "flag = False" only if it is redundant or entirely irrelevant.

Figure 8: Self-Critique prompt template.

Self-Decomposition Evaluation Prompt

```
You are given two problem decomposition results for the same complex problem. Your task is to compare these results from Conciseness, Rationality, Sequencing and Goal Orientation. Analyze the two decomposition results using the criteria above. Clearly explain which approach is more effective for solving the problem and why, while highlighting the strengths and weaknesses of each approach in detail.

# Scoring Criteria:
- Score each dimension on a scale of 1-5, where:
- 1: Poor
- 2: Needs Improvement
- 3: Average
- 4: Good
- 5: Excellent
# The output follows the format below. Do not add any additional text: {
"Conciseness": {
"Result 1 Score": X,
"Result 2 Score": Y,
"Explanation": "How effectively does each decomposition avoid unnecessary complexity while still addressing all relevant aspects of the problem? Is the explanation clear and straightforward?"
},
"Rationality": {
"Result 1 Score": X,
"Result 2 Score": Y,
"Explanation": "Are the identified components logical and directly related to the problem? Do the solutions align well with the identified components?"
},
"Sequencing": {
"Result 1 Score": X,
"Result 2 Score": Y,
"Explanation": "Is the order of steps or components logical and easy to follow? Does the sequence facilitate efficient problem-solving?"
},
"Goal Orientation": {
"Result 1 Score": X,
"Result 2 Score": Y,
"Explanation": "Do the sub-questions stay aligned with the core goal of the main problem? Are there any redundant sub-questions that deviate from the primary objective?"
},
"Result": "Decomposition Results 1 Decomposition Results 2 Tie"
}
# Problem:
{problem} # Decomposition Results to Compare:
- Decomposition Results 1:
{result1}
- Decomposition Results 2:
{result2} # Output:
```

Figure 9: GPT-4o decomposition prompt template.