ARISE: Towards Knowledge-Augmented Reasoning via Risk-Adaptive Search

Yize Zhang[1,2,3*] **Tianshu Wang**[5,7,8*†] **Sirui Chen**[1,6]
Kun Wang[4] **Xingyu Zeng**[4] **Hongyu Lin**[5]
Xianpei Han[5] **Le Sun**[5] **Chaochao Lu**[1,2‡]

[1]Shanghai AI Laboratory [2]Shanghai Innovation Institute [3]Shanghai Jiao Tong University
[4]SenseTime [5]Institute of Software, Chinese Academy of Sciences [6]Tongji University
[7]Hangzhou Institute for Advanced Study, [8]University of Chinese Academy of Sciences
ez220523@sjtu.edu.cn, tianshu2020@iscas.ac.cn, luchaochao@pjlab.org.cn

Abstract

Large language models (LLMs) have demonstrated impressive capabilities and are receiving increasing attention to enhance their reasoning through scaling test-time compute. However, their application in *open-ended, knowledge-intensive, complex reasoning* scenarios is still limited. Reasoning-oriented methods struggle to generalize to open-ended scenarios due to implicit assumptions of complete world knowledge. Meanwhile, knowledge-augmented reasoning (KAR) methods fail to address two core challenges: 1) error propagation, where errors in early steps cascade through the chain, and 2) verification bottleneck, where the explore–exploit trade-off arises in multi-branch decision processes. To overcome these limitations, we introduce ARISE, a novel framework that integrates risk assessment of intermediate reasoning states with dynamic retrieval-augmented generation (RAG) within a Monte Carlo tree search paradigm. This approach enables effective construction and optimization of reasoning plans across multiple maintained hypothesis branches. Experimental results show that ARISE significantly outperforms the state-of-the-art KAR methods by up to 23.10%, and the latest RAG-equipped large reasoning models by up to 25.37%. Our project page is at https://opencausalab.github.io/ARise.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities across a wide range of tasks (OpenAI, 2023; Zhao et al., 2023b; Bubeck et al., 2023). Despite their great success, LLMs still face fundamental challenges in complex reasoning scenarios, hindering their reliable application in real-world domains such as science, finance, and healthcare (Taylor et al., 2022; Li et al., 2023; Thirunavukarasu et al., 2023). To address this gap, recent research has increasingly focused on enhancing LLM reasoning by scaling test-time compute to emulate System 2 slow thinking, moving beyond System 1 fast responses (Kahneman, 2011; Snell et al., 2024). Extensive efforts have led to various approaches, including prompt-based (Yu et al., 2024a), search-based (Hao et al., 2023), and learning-based (OpenAI, 2024; DeepSeek-AI et al., 2025), showing great promise.

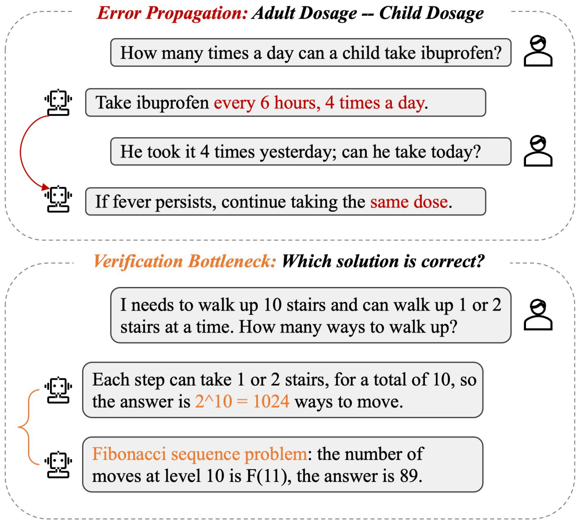

Figure 1: **Error propagation and verification bottleneck.** Prior knowledge-augmented reasoning works fail to address two core challenges: 1) error propagation, where errors in early steps cascade through the chain, and 2) verification bottleneck, where the explore–exploit trade-off arises in multi-branch decision processes.

However, reasoning-oriented methods struggle to generalize to open-ended scenarios (Valmeekam et al., 2022; Amirizaniani et al., 2024), primarily due to their implicit assumptions of complete world knowledge. While these solutions like large reasoning models (LRMs) have achieved expert or superhuman performance on tasks such as math and code, their success relies heavily on clear standards for search algorithms or reinforcement learn-

*Equal contribution.
†Work done during an internship at SenseTime.
‡Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10978–10995
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ing (Zhang et al., 2024a; Xu et al., 2025). Such an exclusive focus on enhancing LLM reasoning implicitly assumes that LLMs already possess all the knowledge necessary for reasoning, which is often lacking in open-ended or domain-specific contexts. For example, legal defense requires specialized jurisprudence knowledge, or medical diagnosis demands up-to-date clinical guidelines. In fact, reasoning is a dynamic process of integrating multiple knowledge to draw conclusions (Yu et al., 2024a; OpenAI, 2025), thus making knowledge acquisition an essential part of reasoning.

Meanwhile, current knowledge-augmented reasoning (KAR) methods, as illustrated in Figure 1, are hindered by *error propagation* and *verification bottleneck*, which undermine reasoning reliability. To acquire knowledge for reasoning, retrieval-augmented generation (RAG) has been shown to be an effective way of dynamically retrieving documents as intermediate results (Lewis et al., 2020; Liu et al., 2024). Prompt-based methods further extend KAR through chain-of-thought (CoT) prompting, which decomposes complex reasoning into sub-steps and iteratively retrieves relevant knowledge as reasoning proceeds (Zhao et al., 2023a; Yu et al., 2024b; Li et al., 2024). However, this approach is plagued by error propagation, where errors in early steps can cascade through the chain. While search-based methods can mitigate error propagation by maintaining multiple hypothesis branches, verification bottleneck limits the effective explore–exploit trade-off in KAR. Existing verification solutions remain unsatisfactory as they rely on error-prone self-verification (Stechly et al., 2024; Wang et al., 2023b; Zhang et al., 2024b), or on specific verifier training (Setlur et al., 2024; Zhang et al., 2024a).

To overcome these limitations, we present a novel framework, ARISE, towards knowledge-**A**ugmented **R**easoning via r**I**sk-adaptive **SE**arch. As shown in Figure 2, ARISE consists of three components: **reasoning state generation**, **Monte Carlo tree search** (MCTS), and **risk assessment**. Specifically, ARISE iteratively refines reasoning steps through decomposition, retrieval-then-reasoning to provide fine-grained knowledge for LLMs (§ 2.1). MCTS treats each step as a node in the search tree, expanding linear reasoning to mitigate error propagation by enabling focused exploration of promising reasoning states and allowing backtracking when necessary (§ 2.2). Risk assessment leverages Bayesian risk minimization to evaluate the uncertainty of each state, dynamically balancing explore–exploit trade-off to guide the search towards both reliable and novel reasoning directions (§ 2.3). In this way, ARISE enables robust and efficient complex reasoning by combining structured decomposition, knowledge retrieval, and risk-adaptive exploration in a unified framework.

We conducted comprehensive experiments with multiple LLMs on three challenging multi-hop question answering (QA) benchmarks that require complex reasoning and knowledge integration. Experimental results demonstrate that ARISE significantly outperforms the state-of-the-art (SOTA) KAR methods, with an average of 23.10% and 15.52% improvement in accuracy and F1. In addition, when compared to the latest LRMs (DeepSeek-AI et al., 2025) equipped with RAG, ARISE also improve the average accuracy and F1 of 4.04% and 25.37%. These results verify the effectiveness of ARISE for open-ended, knowledge-intensive, complex reasoning tasks.

To summarize, our **contributions** are as follows:

- We propose a knowledge-augmented framework for open-ended complex reasoning and design a risk-adaptive MCTS algorithm to balance explore-exploit trade-off for reasoning.
- We conduct comprehensive experiments to verify the effectiveness of ARISE and to demonstrate that it outperforms the SOTA KAR methods and the latest LRMs equipped with RAG.
- We provide empirical insights that 1) search-based wide reasoning can explore more solutions than learning-based deep reasoning, and 2) ARISE progressively approaches optimal performance through model size scaling.

2 The ARISE Method

Our method, ARISE, utilizes risk-adaptive tree search to provide the model with more external knowledge, thereby effectively enhancing its reasoning capabilities. Our pipeline is illustrated in Figure 2 and comprises the following three parts:

• **Reasoning State Generation:** The single step of the policy model[1] consists of an action pair: decomposition and retrieval-then-reasoning. Each step serves as a node, encoding an intermediate reasoning state.

• **Monte Carlo Tree Search:** MCTS transforms a sequence of interconnected nodes into a

[1]We use "policy models" to refer to the LLMs employed during the inference phase.

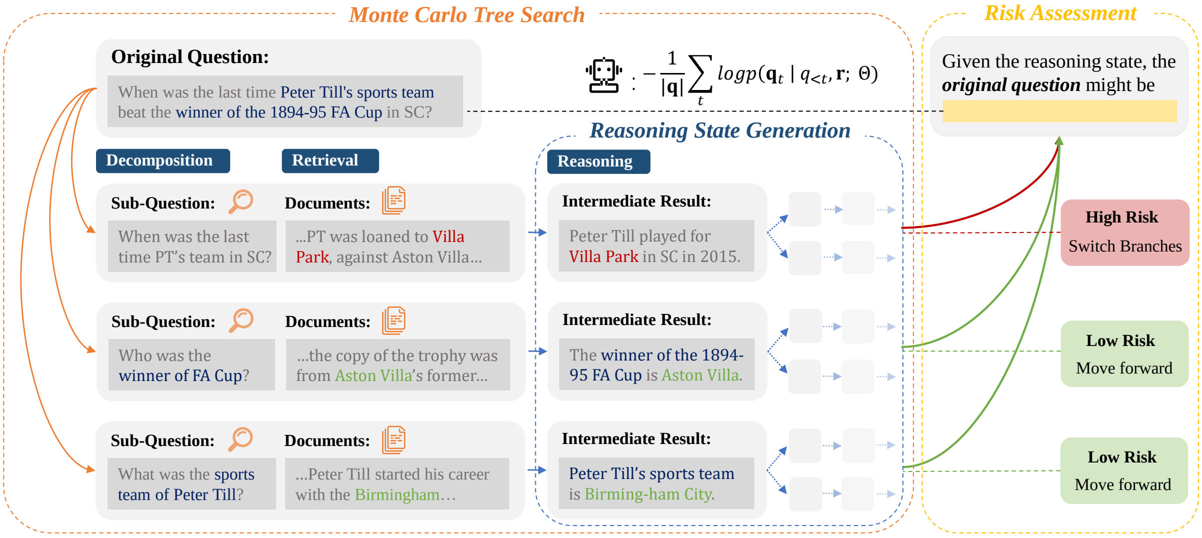

Figure 2: **Pipeline of ARISE.** ARISE iteratively refines reasoning steps through decomposition, retrieval-then-reasoning, providing fine-grained knowledge for LLMs (§ 2.1). MCTS treats each step as a node in the search tree, expanding linear reasoning to mitigate error propagation by enabling exploration of reasoning paths and allowing backtracking when necessary (§ 2.2). Risk assessment leverages Bayesian risk minimization to evaluate the quality of each state, dynamically optimizing action strategies to guide the search towards promising directions (§ 2.3).

tree structure. Each node can undergo a simulated rollout under guidance. The local value incorporating future reward can be updated without incurring the cost of taking an actual forward step.

- **Risk Assessment:** We designed the Risk-Value function to assess the risks of the intermediate reasoning states at each node. The policy model is capable of dynamically formulating and adjusting action strategies based on the actual risk associated with each branch.

2.1 Reasoning State Generation

To define steps more clearly and granularly, we prompt LLMs to perform problem decomposition and retrieval-then-reasoning in an interleaved manner. These two consecutive actions together constitute a single step. Intermediate results at each step are continuously appended to the entire reasoning state and serve as new inputs for subsequent steps, progressively approaching the final solution to complex tasks. This approach, where intermediate subtasks and their labels are concatenated to the original task's input to form a new input sequence, is widely applied in compounded tasks (Wies et al., 2023; Rajani et al., 2019; Cobbe et al., 2021). Specifically, at the i^{th} step[2], the input comprises the original problem $\mathbf{q}$ and the intermediate results $\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_{i-1}}$ from previous steps, with the latter forming the reasoning state $\mathbf{s_{i-1}} = \mathbf{r_1} \oplus \mathbf{r_2} \oplus \ldots \oplus \mathbf{r_{i-1}}$. The policy model then decomposes the problem into a subproblem $\mathbf{d_i}$, following the policy $\pi(\mathbf{d_i} \mid \mathbf{q}, \mathbf{s_{i-1}})$. Based on the subproblem $\mathbf{d_i}$ and the retrieved documents, the intermediate result $\mathbf{r_i}$ is then generated and appended to the reasoning state repeatedly. Each step encodes an $(\mathbf{s_{i-1}}, \mathbf{a_i})$ pair, where $\mathbf{s_{i-1}}$ represents the state, and $\mathbf{a_i}$ is a set $\{\mathbf{d_i}, \mathbf{r_i}\}$ that implicitly reflects the step's two actions, with $\mathbf{d_i}$ being the outcome corresponding to the decomposition and $\mathbf{r_i}$ to the retrieval-then-reasoning. A sequence of coherent steps, extending until the endpoint, collectively forms a complete trajectory.

2.2 Monte Carlo Tree Search

The MCTS algorithm expands a single trajectory into a search tree structure. The whole process begins with the original problem as the root node, followed by iterative searches consisting of selection, expansion, simulation, and backpropagation. The four phases are detailed as follows:

Selection. Starting from the root node and traversing the existing tree structure, the algorithm selects the optimal child node in preparation for the next expansion phase. To balance exploration and exploitation, the well-known Upper Confidence Bounds (UCT) (Kocsis and Szepesvári, 2006) is used in the selection process, formulated as:

$$\mathrm{UCT}(\mathbf{s}, \mathbf{a}) = Q(\mathbf{s}, \mathbf{a}) + w\sqrt{\frac{\ln N(Pa(\mathbf{s}))}{N(\mathbf{s}, \mathbf{a})}},$$

[2]We use the **bold** notation for vectors and non-bold notation for scalars.

where $N(\mathbf{s}, \mathbf{a})$ and $N(Pa(\mathbf{s}))$ represent the visit counts of the current node and its parent node in previous searches, respectively. The initial value of $Q(\mathbf{s}, \mathbf{a})$ is calculated by the Risk-Value function (detailed in § 2.3) and is subsequently updated during the backpropagation phase.

Expansion. The model decomposes the original problem based on the reasoning state from different perspectives to generate new subproblems. Each subproblem and its corresponding result form a distinct child node, which is then appended to the selected node, thereby expanding the tree in both width and depth.

Simulation. The model initiates an imagined rollout from the selected node, proceeding until it reaches a leaf node. This phase assists in assigning the current node a more farsighted value that incorporates future rewards by completing the imagined reasoning trajectory without altering the tree structure. Within a single rollout, the model can still sample multiple times and greedily advance towards the leaf nodes.

Backpropagation. The backpropagation phase updates the values of all nodes along the selected reasoning branch. This process follows a bottom-up manner, where the parent node's value is determined by the values and visit counts of its child nodes. The mathematical formulation is as follows:

$$Q(\mathbf{s}, \mathbf{a}) = \frac{\sum_{\mathbf{c} \in \mathcal{C}(\mathbf{s}, \mathbf{a})} Q(\mathbf{c}) \cdot N(\mathbf{c})}{\sum_{\mathbf{c} \in \mathcal{C}(\mathbf{s}, \mathbf{a})} N(\mathbf{c})},$$

where $\mathcal{C}(\mathbf{s}, \mathbf{a})$ denotes all child nodes of $(\mathbf{s}, \mathbf{a})$.

After reaching the predetermined number of search iterations, the tree structure and node values stabilize. Ultimately, the model selects the optimal path by maximizing value at each step, following the greedy policy.

2.3 Risk Assessment

In this section, we delve into the Risk-Value function, which assesses the risks of reasoning states to guide the tree-search process. To begin with, for a composite problem $\mathbf{q}$, we treat its decomposition and retrieval-then-reasoning as a statistical decision of a probabilistic process (Zhai and Lafferty, 2006; Lafferty and Zhai, 2001). Specifically, given a set of decomposed subproblems $D = \{\mathbf{d^1}, \mathbf{d^2}, \ldots, \mathbf{d^k}\}$ and the corresponding set of intermediate results $R = \{\mathbf{r^1}, \mathbf{r^2}, \ldots, \mathbf{r^k}\}$[3], the quality of a node state can be evaluated using a relevance score $p(\mathbf{r} \mid \mathbf{q}), \mathbf{r} \in R$ (Sachan et al., 2022). We substitute the "problem generation likelihood" (Zhai and Lafferty, 2001; Ponte and Croft, 1998) as an alternative to the relevance score after applying the Bayes' rule:

$$\log p(\mathbf{r} \mid \mathbf{q}) = \log p(\mathbf{q} \mid \mathbf{r}) + \log p(\mathbf{r}) + c,$$

where $p(\mathbf{r})$ is the prior belief that $\mathbf{r}$ is relevant to any problem and is assumed to be uniform in this case. We can also drop c since it is the intermediate result-independent constant. The formula is then reduced to:

$$\log p(\mathbf{r} \mid \mathbf{q}) \propto \log p(\mathbf{q} \mid \mathbf{r}), \forall \mathbf{r} \in R,$$

where $p(\mathbf{q} \mid \mathbf{r})$ captures how well the intermediate results $\mathbf{r}$ fit the particular problem $\mathbf{q}$. We utilize the policy model to compute the average log-likelihood of generating the original problem tokens in order to estimate $\log p(\mathbf{q} \mid \mathbf{r})$ (Sachan et al., 2022; Yuan et al., 2024), and define the expected risk of a node $(\mathbf{s}, \mathbf{a})$ pointing to $\mathbf{r}$ as follows:

$$\text{Risk}((\mathbf{s}, \mathbf{a}) \to \mathbf{r} \mid \mathbf{q}) = -\frac{1}{|\mathbf{q}|} \sum_t \log p(q_t \mid \mathbf{q}_{<t}, \mathbf{r}; \Theta),$$

where $|\mathbf{q}|$ denotes the length of the original problem, q_t represents the t^{th} token in $\mathbf{q}$, and $\mathbf{q}_{<t}$ refers to the sequence of tokens preceding the t^{th} token in $\mathbf{q}$. Θ denotes the parameters of the policy model. Finally, the risk is scaled to the range (0, 1) through a sigmoid function in the opposite direction, serving as the node value:

$$Q(\mathbf{s}, \mathbf{a}) = 1 - \frac{1}{1 + e^{\alpha \cdot (\text{Risk} - \beta)}},$$

where α, β are the translation and scaling factors.

3 Experiments

3.1 Setup

Datasets. We use HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MusiQue (Trivedi et al., 2022) as test sets. These datasets span a wide range of topics, necessitating the retrieval and reasoning over multiple supporting documents. To balance computational efficiency and evaluation robustness, we conduct experiments on a subset of 200 randomly selected questions (Jiang et al., 2024; Feng et al., 2025). More details are available in § B.1.

[3] In this notation, the subscript of the symbol denotes the sequence number of the reasoning step, while the superscript indicates the identifier for different reasoning perspectives.

Method	HotpotQA		2Wiki		MusiQue		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
Qwen2.5-14B-Instruct								
Vanilla	59.50	63.63	37.00	50.33	14.50	47.07	37.00	53.68
Prompt-based								
Query2Doc (Wang et al., 2023a)	61.00	67.65	38.00	53.40	22.00	55.79	40.33	58.95
Self-Ask (Press et al., 2023)	58.50	64.74	38.50	53.45	25.00	58.59	40.67	58.93
Verify-and-Edit (Zhao et al., 2023a)	62.50	70.16	38.50	57.68	22.00	55.30	41.00	61.05
Auto-RAG (Yu et al., 2024b)	68.00	66.64	53.00	55.13	35.50	59.05	52.17	60.27
Search-based								
CoT-SC (Wang et al., 2023b)	63.00	71.39	37.50	54.41	16.00	57.46	38.83	61.09
RATT (Zhang et al., 2024b)	64.50	73.91	43.00	57.48	24.00	63.76	43.83	65.05
ARISE (ours)	**73.50**	**75.39**	**56.50**	**62.61**	**40.50**	**65.87**	**56.83**	**67.96**
Qwen2.5-7B-Instruct								
Vanilla	54.50	63.63	36.50	50.33	11.00	47.07	34.00	53.68
Prompt-based								
Query2Doc (Wang et al., 2023a)	60.00	67.31	37.00	53.78	14.50	54.59	37.17	58.56
Self-Ask (Press et al., 2023)	44.00	61.43	27.50	48.86	22.00	57.48	31.17	55.92
Verify-and-Edit (Zhao et al., 2023a)	**67.00**	69.87	39.00	53.91	21.50	54.75	42.50	59.51
Auto-RAG (Yu et al., 2024b)	66.50	66.33	44.50	54.00	**29.50**	57.19	46.83	59.17
Search-based								
CoT-SC (Wang et al., 2023b)	60.50	70.66	38.00	54.01	15.00	56.72	37.83	60.46
RATT (Zhang et al., 2024b)	58.50	68.88	36.50	53.91	18.00	56.58	37.67	59.79
ARISE (ours)	66.50	**73.87**	**47.50**	**61.37**	29.00	**62.26**	**47.67**	**65.83**
Llama3.1-8B-Instruct								
Vanilla	55.50	63.63	31.50	50.33	14.00	47.07	33.67	53.68
Prompt-based								
Query2Doc (Wang et al., 2023a)	57.50	63.52	32.50	49.78	19.00	49.91	36.33	54.40
Self-Ask (Press et al., 2023)	57.00	62.10	40.00	51.26	20.50	52.14	39.17	55.17
Verify-and-Edit (Zhao et al., 2023a)	50.50	65.07	29.00	51.91	13.50	49.77	31.00	55.58
Auto-RAG (Yu et al., 2024b)	51.00	53.37	35.00	48.81	21.50	52.61	35.83	51.60
Search-based								
CoT-SC (Wang et al., 2023b)	**64.50**	71.40	45.00	58.02	22.00	59.43	43.83	62.96
RATT (Zhang et al., 2024b)	58.50	71.18	**46.00**	56.18	**29.50**	63.66	**44.67**	63.67
ARISE (ours)	63.00	**74.78**	34.50	**63.19**	24.50	**66.38**	40.67	**68.12**

Table 1: **Comparison of ARISE with a wide range of baselines.**

Baselines and Metrics. All baselines are incorporated with RAG. For prompt-based baselines, we compare ARISE with Query2Doc (Wang et al., 2023a), Self-Ask (Press et al., 2023), Verify-and-Edit (Zhao et al., 2023a) and Auto-RAG (Yu et al., 2024b). For search-based baselines, we use Self-Consistency (SC) (Wang et al., 2023b) and Retrieval-augmented-thought-tree (RATT) (Zhang et al., 2024b). We choose EM accuracy and F1 score as our evaluation metrics. The prediction is correct if the ground truth answer is exactly contained (Jiang et al., 2024; Feng et al., 2025). More details are available in § B.2.

Implementation Details. In the main experiments, we configure the number of iterations to 200, the exploration weight factor w in UCT to 1.4, and the temperature to 0.7. For the retrieval process, we employ BM25 as our retriever. Prompts for forward reasoning directly describe the task with zero-shot instructions. For verification prompts, we provide few-shot demonstrations. Further details and specific prompts are available in § B.4.

3.2 Main Results

***Finding 1:* ARISE demonstrates superior performance.** Table 1 presents the comprehensive experimental results on ARISE and various baselines. Specifically, on the Qwen2.5-14B-Instruct model, ARISE outperforms across all benchmarks, achieving an absolute improvement of 19.83% in

EM over the vanilla RAG method, 13.29% over prompt-based baselines, and 15.5% over search-based baselines. ARISE maintains robust performance on the Qwen2.5-7B-Instruct model with an absolute improvement of 13.67% in EM over the vanilla RAG method and overall surpasses various baselines. We observed that ARISE performs slightly worse on Llama models. To further analyze this, we identify another interesting phenomenon: Auto-RAG, which adopts the same interleaved decomposition and retrieval paradigm as ARISE, also exhibits a decline on Llama. This phenomenon suggests that the Llama model may not be well-suited for iterative problem decompositions. In contrast, continuous-step reasoning methods such as CoT and tree-of-thoughts show better results. Nevertheless, ARISE still maintains a notable F1 advantage on Llama, indicating its effectiveness in selecting more promising paths.

Finding 2: **ARISE demonstrates substantial potential on more challenging tasks.** We are excited to show that ARISE demonstrates superior performance on more challenging datasets. Based on average performance, the difficulty level increases progressively from HotpotQA to 2WikiMultihopQA, and then to MusiQue. In particular, on the 14B model, ARISE achieves a relative improvement of 23.53% in EM over the vanilla RAG method on HotpotQA, which surges to 52.70% and 179.31% on 2WikiMultihopQA and MusiQue, respectively. In contrast, a wide range of baselines show only average performance improvements of 5.74%, 11.94%, and 66.09%, respectively. The F1 score reflects the same trend, with a relative improvement of 18.48%, 24.40%, and 39.94%, corresponding to the three benchmarks.

3.3 Comparison to Learning-based LRMs

Finding 3: **Learning-based LRMs have not yet approached the point where they can effectively match or even replace search-based reasoning as ARISE.** Our empirical comparison between base models with ARISE and the DeepSeek-R1 distilled models reveals key insights into the effectiveness of test-time search. These learning-based LRMs extract similar reasoning patterns from DeepSeek-R1. As shown in Figure 3, ARISE exhibits a performance advantage over the LRMs, especially on the Qwen model series. While ARISE slightly underperforms in comparison to DeepSeek-R1-style reasoning pattern on Llama, it consistently outperforms it on the Qwen models. On average, ARISE shows a relative improvement of 4.03%, emphasizing the benefit of our search-based method. These results suggest that while learning-based LRMs like DeepSeek-R1 distilled models provide valuable insights, they have not yet demonstrated the same effectiveness as search-based reasoning.

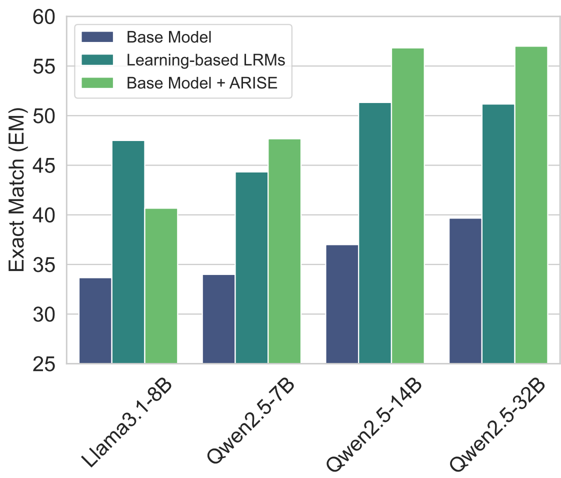

Figure 3: **Search-based reasoning vs. learning-based LRMs.** Learning-based LRMs like DeepSeek-R1 distilled models have not yet approached the point where they can effectively match or even replace search-based reasoning methods in terms of performance.

3.4 Model Scale

Finding 4: **ARISE progressively approaches the optimal performance upper bound as model scale increases, unlocking the potential of larger models.** We conducted experiments on the Qwen2.5 series models, spanning a range of parameter scales from 0.5B to 32B, as illustrated in Figure 4. To realistically and prominently reflect the related trends, we selected two more challenging test sets: 2WikiMultihopQA and Musique. We employ Pass@N as the metric to evaluate the upper bound of the success rate, where a problem is considered solved as long as a single surviving node in the tree leads to the correct answer. Pass@1, on the other hand, represents the success rate of the optimal path selected under the guidance of ARISE. The results show that Pass@N and the vanilla method exhibit similar trends as model parameters scale, but there is an average of 26.85% significant room for improvement. This indicates that appropriately scaling up inference computation offers substantial potential for enhancing performance. We observed that after the model size

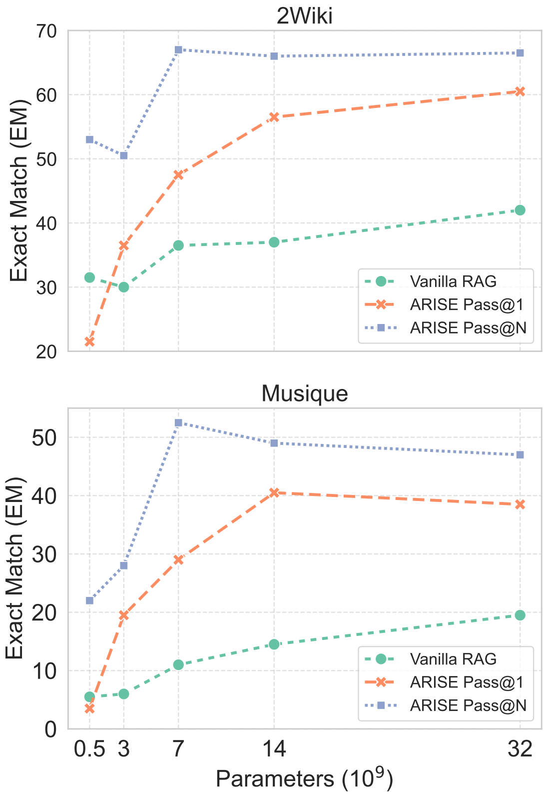

Figure 4: **Performance vs. model scales.** Although scaling up the model size shows diminishing returns for vanilla approaches, ARISE better harnesses the potential of larger models in solving complex tasks.

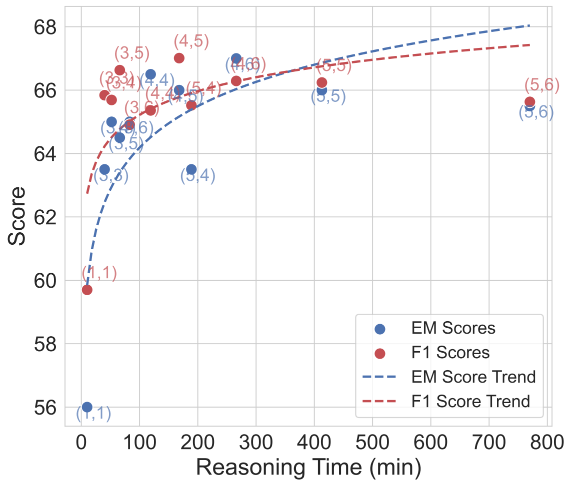

Figure 5: **Computational overhead in relation to the search space.** As the search space expands in depth and width, performance improves but with diminishing returns and significantly increased reasoning time. The (4,5) setting achieves a good trade-off and is adopted in our main experiments.

reaches 7B parameters, the performance of both the upper bound and naive retrieval tends to saturate, suggesting diminishing returns with further scaling of model parameters. In contrast, ARISE demonstrates consistent improvement as model scale increases. The accuracy of the optimal path selection gradually approaches the upper bound, with the success rate gap between Pass@1 and Pass@N decreasing from 25.00% to 7.25%.

3.5 Computational Overhead

The computational overhead is measured in terms of reasoning time (in minutes).

3.5.1 Overhead in Relation to Search Space.

***Finding 5:* Moderate search space expansion boosts performance over minimal settings, but returns diminish rapidly as overhead surge.** We conducted experiments on the HotpotQA dataset using the Llama3.1-8B-Instruct model. The results are under different search space configurations—defined by search depth and width. As shown in Figure 5, expanding the search space leads to performance improvements (in EM and F1 scores), especially when increasing from a vanilla setting (depth 1, width 1) to moderate configurations. However, these improvements diminish as the depth and width increase further, while the computational overhead escalates rapidly. For instance, increasing the depth from 3 to 4 and the width from 4 to 6 results in a reasoning time increase from 52 to 266 minutes, with only marginal performance gain. Based on this observation, we adopted a configuration of depth = 4 and width = 5 for our main experiments, which strikes a practical balance between accuracy and computational cost. It is worth noting that the optimal search space configuration may vary across instances. Different questions demand varying depths of reasoning and degrees of knowledge integration, often corresponding to different numbers of reasoning hops. Thus, adaptively determining the search space based on task complexity remains an open research problem. Even state-of-the-art reasoning models still face the challenge of choosing between wider and deeper reasoning paths. We discuss this further in § 5 and identify it as a promising direction for future work.

3.5.2 Overhead Comparison with Baselines.

***Finding 6:* ARISE achieves the best performance but incurs higher computational overhead due to its search-based paradigm.** We further evaluate the computational overhead of our method, ARISE, in comparison with a set of com-

Method	EM	F1	Time (min)
Vanilla	14.50	47.07	10
Query2Doc	22.00	55.79	16
Self-Ask	25.00	58.59	18
Verify-and-Edit	22.00	55.30	21
Auto-RAG	35.50	59.05	26
CoT-SC	16.00	57.46	69
RATT	24.00	63.76	155
ARISE (ours)	**40.50**	**65.87**	**160**

Table 2: **Computational overhead comparison with baselines.** ARISE achieves the best performance in both EM and F1 scores but incurs higher reasoning time due to its search-based multi-step reasoning. Practical deployment should consider the trade-off between computational resources and desired performance.

petitive baselines on the Musique dataset using the Qwen2.5-14B-Instruct model. The results are summarized in Table 2. As shown, ARISE achieves the highest EM and F1 scores, demonstrating the effectiveness of our search-based reasoning strategy. However, this comes at a higher computational cost, primarily due to the enlarged search space and multi-step reasoning process. In practical applications, the trade-off between performance and reasoning time should be contextualized by the nature of the task at hand. In high-stakes domains such as finance, healthcare, and law, sacrificing additional computational time for improved accuracy is often a worthwhile investment.

3.6 Ablation Studies

3.6.1 Risk Assessment

***Finding 7:* The Risk-Value function effectively risk-assess reasoning states to guide the tree search process.** We conducted ablation studies to evaluate the effectiveness of the Risk-Value (R-V) function in guiding Monte Carlo Tree Search (MCTS). Table 3 present experimental results compared to vanilla MCTS and MCTS with LLM-as-Judge[4] baselines. The incorporation of the R-V function resulted in improvements across all datasets. Specifically, it achieved an average relative performance gain of 10.71% over the vanilla MCTS baseline. The function's impact was even more pronounced in more challenging tasks, with improvements reaching up to 17.39% on MusiQue. This demonstrates the function's capacity to better navigate and prioritize lower-risk paths, ensuring more efficient exploration and exploitation within the search space. In comparison, MCTS with LLM-as-Verifier showed marginal improvements over the vanilla approach. While pretrained LLMs can provide meaningful context during verification, they are not specifically tuned for evaluating the quality of reasoning states in a dynamic environment. This suggests that pretrained LLMs are insufficient as standalone verifiers in path planning, underscoring the critical role of specialized functions like Risk-Value in guiding the search process.

3.6.2 Iterations of MCTS

***Finding 8:* ARISE with dynamic risk assessment achieves near-optimal solutions with a relatively low inference cost.** We conduct empirical experiments on the trade-off between inference cost and solution quality. Specifically, we examined the performance of ARISE as the number of MCTS iterations increased from 1 to 200. Figure 6 illustrates the efficiency of ARISE in reaching near-optimal solutions with a relatively low inference cost. As

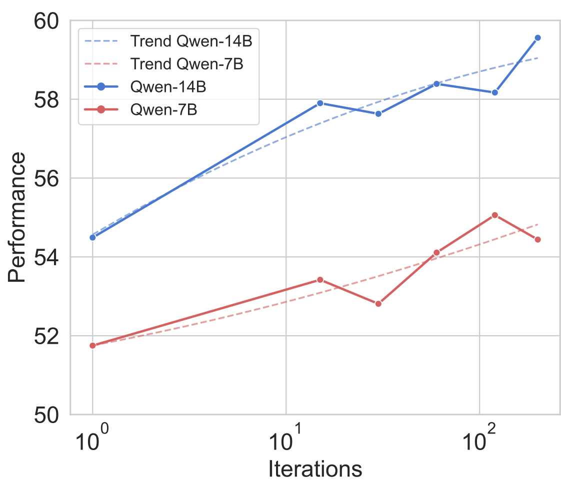

Figure 6: **Ablation on iterations.** We evaluate performance using the average of EM accuracy and F1 score.

the number of MCTS iterations increases, the performance improves, but the rate of improvement diminishes with higher iteration counts. At the initial stages of the search process (from 1 to 30 iterations), the performance shows rapid improvements. The initial exploration of potential moves encourages the model's better understanding of the search space. The subsequent increase in the number of additional paths during this phase contributes meaningfully to the quality of the solution. Beyond 30 iterations, the improvements in performance begin to level off. For instance, between 30 and 60

[4]To ensure experimental fairness, we employed the same LLM (policy models) for methods involving LLM-as-verifier.

Method	HotpotQA		2Wiki		MusiQue		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
MCTS (R-V function)	**73.50**	**75.39**	**56.50**	**62.61**	**40.50**	**65.98**	**56.83**	**67.96**
MCTS (vanilla)	71.00	73.58	48.50	60.47	34.50	63.34	51.33	65.80
MCTS (LLM-as-verifier)	69.00	73.85	53.50	60.98	34.00	64.30	52.17	66.38

Table 3: **Ablation on the Risk-Value function.** The incorporation of the Risk-Value function resulted in improvements across all datasets. The Risk-Value function effectively assesses reasoning states, guiding the tree search process; however, pretrained LLMs are not adequate as verifiers.

iterations, the performance increases slightly by a relative 1.32%, whereas during the initial phase, the increase was 5.76%. This suggests that further exploration of the search space yields diminishing returns as the algorithm begins to converge toward the optimal decision. This phenomenon can be attributed to that ARISE provides dynamic step-level Risk-Value, without requiring the model to wait for outcome verification to guide the next iteration. The Risk-Value function efficiently narrows down the promising branches at a rapid pace. The performance stabilizes after approximately 100 iterations. Further iterations may lead to incremental improvements but are less impactful in a highly explored search space.

4 Related Works

Test-Time Compute. Scaling test-time compute has become a popular topic recently in many research efforts (Brown et al., 2024; Snell et al., 2024), aiming to shift the reasoning paradigm of LLMs from the fast but error-prone system 1 thinking to the more deliberate system 2 thinking (Kahneman, 2011; Snell et al., 2024). Prior works mainly include prompt-based (Yu et al., 2024a), search-based (Hao et al., 2023), and learning-based (OpenAI, 2024; DeepSeek-AI et al., 2025) methods. Prompt-based reasoning utilize chain-of-thought (CoT) (Wei et al., 2022) prompting to break down complex reasoning into sub-steps, gradually approaching a more accurate final answer through the generation of additional intermediate tokens (Zhao et al., 2023a; Yu et al., 2024b; Li et al., 2024). Search-based reasoning allows LLMs to improve performance by generating multiple samples, and tree-based methods have further integrated planning and exploration (Yao et al., 2023; Besta et al., 2024; Hao et al., 2023). While the multiple and redundant rollouts significantly burden the inference spend, verifiers for solution-selection is essential for ensuring efficiency. Learning-based methods aim to inject the deep reasoning patterns of large and complicated models into smaller models through post-training (OpenAI, 2024; DeepSeek-AI et al., 2025).

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) merges the intrinsic knowledge of LLMs with a vast, dynamic repository of external databases, mitigating the issues of language model hallucination and outdated knowledge to some extent (Lewis et al., 2020; Gao et al., 2023). Recent studies (Wang et al., 2023a; Press et al., 2023; Yu et al., 2024b; Zhao et al., 2023a) have proposed some prompting-based strategies for LLMs to better harness the potential of RAG, essentially integrating it into the intermediate reasoning process (*e.g.,* chain-of-thought (CoT) (Wei et al., 2022)). In these methods, the interaction between LLMs and retrieval actions breaks down the reasoning process into discontinuous smaller steps, which helps produce more authentic intermediate results and reduces the instability inherent in autoregressive token generation.

5 Conclusion

In this work, we introduce ARISE, a novel framework that addresses the challenges of error propagation and verification bottleneck in open-ended, knowledge-intensive, and complex reasoning tasks. By integrating Monte Carlo Tree Search with risk-adaptive exploration, ARISE enables dynamic and effective reasoning through iterative problem-decomposition and retrieval-then-reasoning steps. Our experiments demonstrate that ARISE outperforms a wide range of state-of-the-art methods, and also surpasses the performance of the latest learning-based large reasoning models (LRMs) equipped with RAG. These results highlight the strong potential of ARISE in advancing open-ended, knowledge-intensive, and complex reasoning tasks across various real-world applications.

Limitations

Although our method ARISE demonstrates strong performance for knowledge-intensive, and complex reasoning tasks, several limitations remain open for improvement. Our experiments are currently confined to multi-hop question-answering (QA) tasks. The applicability of ARISE to other reasoning tasks, such as mathematical problem-solving, code generation, or complex decision-making, remains to be explored. Extending our method to a broader range of reasoning tasks is an important direction for future work. There is also a need for more systematically designed prompts to ensure generalization and robustness across diverse scenarios. Moreover, existing search-based paradigms mainly rely on post-trained reward models for verification. These models are trained on specially annotated data to learn how to score reasoning states. While this approach has shown progress in mathematics (a relatively closed domain) (Lightman et al., 2024; Luo et al., 2024; Wang et al., 2024), static, post-trained reward models struggle with open-ended, knowledge-intensive tasks, where external knowledge is dynamically involved. Such models cannot accurately assess new knowledge unseen during post-training, limiting their generalization. We argue that a generalizable design of rewards is crucial for future developments. Our work presents an initial attempt in this direction, and we will further explore this line of research. Last but not least, the search space for specific questions is predefined, lacking the ability to adapt efficiently to varying reasoning complexity or knowledge density. This points to an open challenge in the study of efficient reasoning: how to achieve an effective trade-off between broad and deep search, and how to reduce redundancy in the reasoning process. We leave this as an avenue for future work.

Acknowledgement

We thank all the anonymous reviewers and area chair for their valuable feedback throughout the review process. This work is supported by Shanghai Artificial Intelligence Laboratory.

References

Maryam Amirizaniani, Elias Martin, Maryna Sivachenko, Afra Mashhadi, and Chirag Shah. 2024. Can llms reason like humans? assessing theory of mind reasoning in llms for open-ended questions. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pages 34–44. ACM.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.

Bradley C. A. Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Jingyi Song, and Hao Wang. 2025. Airrag: Activating intrinsic reasoning for retrieval augmented generation via tree-based search. *Preprint*, arXiv:2501.10053.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.

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, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Jinhao Jiang, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2024. Rag-star: Enhancing deliberative reasoning with retrieval augmented verification and refinement. *CoRR*, abs/2412.12881.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer.

John D. Lafferty and ChengXiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 111–119. ACM.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *4th ACM International Conference on AI in Finance, ICAIF 2023, Brooklyn, NY, USA, November 27-29, 2023*, pages 374–382. ACM.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jingyu Liu, Jiaen Lin, and Yong Liu. 2024. How much can RAG help the reasoning of llm? *CoRR*, abs/2410.02338.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *CoRR*, abs/2406.06592.

OpenAI. 2023. GPT-4 technical report.

OpenAI. 2024. Learning to reason with llms.

OpenAI. 2025. Introducing deep research.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 275–281. ACM.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4932–4942. Association for Computational Linguistics.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3781–3797. Association for Computational Linguistics.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for LLM reasoning. *CoRR*, abs/2410.08146.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. On the self-verification limitations of large language models on reasoning and planning tasks. *CoRR*, abs/2402.08115.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature Medicine*, 29(8):1930–1940.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Karthik Valmeekam, Alberto Olmo Hernandez, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (A benchmark for llms on planning and reasoning about change). *CoRR*, abs/2206.10498.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9414–9423. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Noam Wies, Yoav Levine, and Amnon Shashua. 2023. Sub-task decomposition enables learning in sequence to sequence tasks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2024a. Natural language reasoning, A survey. *ACM Comput. Surv.*, 56(12):304:1–304:39.

Tian Yu, Shaolei Zhang, and Yang Feng. 2024b. Autorag: Autonomous retrieval-augmented generation for large language models. *CoRR*, abs/2411.19443.

Xiaowei Yuan, Zhao Yang, Yequan Wang, Jun Zhao, and Kang Liu. 2024. Improving zero-shot LLM reranker with risk minimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17967–17983. Association for Computational Linguistics.

ChengXiang Zhai and John D. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 334–342. ACM.

ChengXiang Zhai and John D. Lafferty. 2006. A risk minimization framework for information retrieval. *Inf. Process. Manag.*, 42(1):31–55.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: LLM self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. 2024b. RATT: A thought structure for coherent and correct LLM reasoning. *CoRR*, abs/2406.02746.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023a. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5823–5840. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023b. A survey of large language models. *CoRR*, abs/2303.18223.

A Impact of Retrieval System Quality

To evaluate how sensitive ARISE is to the quality of the underlying retrieval system, we examine two key factors: the type of retriever and the number of retrieved documents.

Dataset	BM25 EM	BM25 F1	BGE EM	BGE F1
Musique	40.50	65.87	44.50	70.47
HotpotQA	73.50	75.39	74.00	75.41
2Wiki	56.50	62.61	72.00	67.36

Table 4: **Impact of Retriever Type.** The dense retriever (BGE) outperforms the sparse retriever (BM25) across all datasets, with more notable gains on challenging benchmarks.

Choice of Retriever. We compare the performance of ARISE when paired with a sparse retriever (BM25) versus a dense retriever (BGE). As summarized in Table 4, dense retrieval consistently outperforms sparse retrieval across all three datasets. On average, BGE improves EM and F1 scores by 11.7%, with particularly substantial gains observed on more challenging datasets such as 2Wiki. This indicates that while ARISE exhibits some robustness to retrieval quality on simpler tasks (*e.g.,* HotpotQA), it substantially benefits from higher-quality evidence in more complex scenarios. These results suggest that although the verifier-guided reasoning in ARISE can partially mitigate retrieval noise, it still relies on access to relevant and precise information to perform effective multi-hop reasoning.

#Doc	EM	F1
1	28.50	44.14
2	40.50	65.87
3	37.00	63.97

Table 5: **Effect of the Number of Retrieved Documents.** Performance peaks at retrieving two documents, suggesting a balance between sufficient evidence and retrieval noise.

Number of Retrieved Documents. We further assess the impact of varying the number of retrieved documents on the Musique dataset. As shown in Table 5, retrieving two documents yields the best performance. Adding a third document does not lead to further improvements and may even slightly degrade performance. This can be attributed to two factors. First, the knowledge density of the task affects how much additional context is beneficial. Second, due to ARISE's step-by-step reasoning design, the overall problem is decomposed into smaller, more manageable subproblems, each requiring less contextual information. In a nutshell, ARISE is more sensitive to missing critical information than to receiving redundant or noisy evidence, exhibiting a degree of robustness to retrieval noise.

B Further Details

B.1 Further Details for Datasets

We use HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MusiQue (Trivedi et al., 2022) as the test set, which are three representative benchmarks for open-ended, knowledge-intensive, and complex reasoning tasks. The questions in these datasets require retrieving and reasoning over multiple supporting documents to answer, and they cover a wide range of topics without being constrained by any existing knowledge base or schema. We performed preprocessing on the dataset. Specifically, considering the limitations of computational resources, we randomly sampled 200 questions from each dataset as the final test set. Each instance includes the original question and its answer, along with possible reference documents and necessary supporting documents. During the testing phase, we treated the possible reference documents for each question as its external knowledge base and employed BM25 as our retriever, with each retrieval returning the top two documents. Based on the original dataset, the majority of questions involve complex reasoning with three or more hops (over 80%) (Feng et al., 2025). Among them, Musique has the highest reasoning difficulty, with a notable higher proportion of multi-hop questions. We fixed the search depth to four layers to align with the number of sub-question decompositions required for problem-solving, thereby reducing unnecessary reasoning overhead. Additionally, we set the initial maximum number of expandable child nodes to 5 to cover hop counts comprehensively, guaranteeing diversity in the model's decomposition of questions. As the search depth increases, the diversity of decomposition perspectives for original questions gradually decreases. Therefore, we also progressively reduced the number of expandable child nodes, which ensures search efficiency.

B.2 Further Details for Evaluations

We choose EM accuracy and F1 score as the evaluation metrics. EM accuracy measures the success rate of the results, while F1 score evaluates the reliability of the reasoning process. Specifically, for EM, we adopt the covered EM calculation method. We preprocess both the model's predictions and the ground truth answers by converting them to a uniform case format. If the ground truth answer is a substring of the predicted result, we consider the prediction correct. This approach aims to genuinely reflect the method's performance. Additionally, we employ concise response prompts to ensure the model's final output is not overly verbose, thereby avoiding false-positive cases. For the calculation of F1 score, we utilize the top two documents related to the entire reasoning path and original question. At the final step, we perform an additional retrieval based on the complete reasoning state to ensure the documents reflect the context of the entire path. To prevent redundant retrievals from obscuring the comparison of F1 scores, we limit each retrieval to return only the top two documents.

B.3 Further Details for LLMs

Throughout the experiments, we primarily utilized the Qwen series (Team, 2024) and Llama series models (Dubey et al., 2024). The Qwen series includes models with scales of 0.5B, 3B, 7B, 14B, and 32B parameters, while the experiments with Llama were mainly conducted on the 8B parameter model. To ensure fairness in the experiments, for any tasks involving risk assessment or state evaluation, we consistently employed the corresponding policy models. In addition, experiments also involved DeepSeek-R1 distilled models (DeepSeek-AI et al., 2025), including Qwen-7B, Qwen-14B, Qwen-32B, and Llama-8B. The distilled small model extracts reasoning patterns from DeepSeek-R1 and demonstrates superior performance compared to the inference patterns obtained through reinforcement learning (DeepSeek-AI et al., 2025).

B.4 Further Details for Prompts

We list the full details of all prompts employed in ARISE as follows. The prompts used for forward inference follow a zero-shot instruction to directly describe the task. We provide few-shot demonstrations in the prompts for risk assessment.

Problem Decomposition

Your task is to decompose the original question into one smaller sub-question based on the Intermediate answer and Observation.
The decomposed process is encouraged to be done from multiple perspectives.
Output a thought to reason the original question, and output one sub-question that you think is appropriate to solve next.
DO NOT REPEAT the question and **DO NOT** try to answer the question.
The output format is limited to:
Thought: ...
Sub-question: ...
Here, the "..." indicates omitted output information that you need to fill in.

Original question: {original question}
Intermediate answers: {reasoning state}
Observation: {retrieved documents}
Output:

Intermediate Answer Generation

Your task is to answer the following question using provided supporting facts.
The output answer should be a complete declarative sentence, rather than directly outputting phrases or words.
DO NOT use pronouns in the sentence.
Specially, if no provided supporting facts, just output "No directly relevant facts found." and nothing else.

Question: {sub-question}
Supporting facts: {retrieved documents}
Output:

Final Answer Generation

Your task is to answer the original question based on the intermediate answers.
Output the final answer directly and nothing else.

Original question: {original question}
Intermediate answers: {reasoning state}
Output:

Risk Assessment

Given intermediate answer containing the facts about the original question, which is unknown, your task is to infer what the original question might have been.
Output the most likely original question directly and nothing else.
Example 1:
Intermediate answer:
Muhammad Ali was 74 years old when he died.
Alan Turing was 41 years old when he died.
The original question might be:
Who lived longer, Muhammad Ali or Alan Turing?
Example 2:
Intermediate answer:
Craigslist was founded by Craig Newmark.
Craig Newmark was born on December 6, 1952.
The original question might be:
When was the founder of craigslist born?

Intermediate answers: {reasoning state}
The original question might be:

LLM-as-Verifier

Given a question, your task is to determine the consistency score of its decomposed sub-questions and corresponding intermediate answers with the original question.
Directly output **JUST A NUMBER** between 0 and 10 to represent the consistency score.
DO NOT output anything else.

Original question: {original question}
Sub-questions: {sub-questions}
Intermediate answers: {reasoning state}
Output:

ARISE Simple Case

STATE 1: depth: 1; action: decompose
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions: Who was the winner of FA Cup?
intermediate-answer:
reasoning state:
STATE 2: depth: 1; action: retrieve-then-reason
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions: Who was the winner of FA Cup?
– retrieve: {supporting facts}
intermediate-answer: The winner of the 1894- 95 FA Cup is Aston Villa.
reasoning state: The winner of the 1894- 95 FA Cup is Aston Villa.
STATE 3: depth: 2; action: decompose
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions: What was the sports team of Peter Till?
intermediate-answer:
reasoning state: The winner of the 1894- 95 FA Cup is Aston Villa.
STATE 4: depth: 2; action: retrieve-then-reason
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions: What was the sports team of Peter Till?
– retrieve: {supporting facts}
intermediate-answer: Peter Till's sports team is Birming-ham City.
reasoning state: The winner of the 1894- 95 FA Cup is Aston Villa. Peter Till's sports team is Birming-ham City.
STATE 5: depth: 3; action: decompose
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions: When was the last time Birming-ham City beat Aston Villa in SC?
intermediate-answer:
reasoning state: The winner of the 1894- 95 FA Cup is Aston Villa. Peter Till's sports team is Birming-ham City.
STATE 6: depth: 3; action: retrieve-then-reason
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions: When was the last time Birming-ham City beat Aston Villa in SC?
– retrieve: {supporting facts}
intermediate-answer: The last time Birming-ham City beat Aston Villa was in March 2005.
reasoning state: The winner of the 1894- 95 FA Cup is Aston Villa. Peter Till's sports team is Birming-ham City. The last time Birming-ham City beat Aston Villa was in March 2005.
STATE 7: depth: 4; action: answer
original question: When was the last time Peter Till's team beat winner of 1894-95 FA Cup in SC?
sub-questions:
intermediate-answer:
reasoning state: The winner of the 1894- 95 FA Cup is Aston Villa. Peter Till's sports team is Birming-ham City. The last time Birming-ham City beat Aston Villa was in March 2005.
final answer: March 2005

NOTE: Decomposition is based on original question, reasoning state, and retrieved documents.