ReaRAG: Knowledge-guided Reasoning Enhances Factuality of Large Reasoning Models with Iterative Retrieval Augmented Generation

Jinxin Liu[1*], Zhicheng Lee[1*], Shulin Cao[1], Jiajie Zhang[1], Weichuan Liu[2], Xiaoyin Che[2], Lei Hou[1†], Juanzi Li[1,2†]
[1] Department of Computer Science and Technology, BNRist;
[2] THU - Siemens Ltd., China Joint Research Center for Industrial Intelligence and IoT;
Tsinghua University, Beijing, China
{liujinxi20@mails.tsinghua.edu.cn, lizhiche23@tsinghua.org.cn, houlei@tsinghua.edu.cn, lijuanzi@tsinghua.edu.cn}

***Abstract*—Large Reasoning Models (LRMs) exhibit strong reasoning abilities, but their reliance on parametric knowledge limits factual accuracy. Recent works adopt reinforcement learning (RL) training to integrate reasoning with retrieval, but such methods are often complex and resource-heavy. We propose ReaRAG, a factuality-enhanced reasoning model trained via strategic distillation, rather than costly RL training. Our solution includes a novel data construction framework with an upper bound on the reasoning chain length. Specifically, we first leverage a LRM to generate deliberate thinking, then select an action from a predefined action space (`Search` and `Finish`). For `Search` action, a query is executed against the RAG engine, where the result is returned as observation to guide reasoning steps later. This process iterates until a `Finish` action is chosen. Experimental results show that ReaRAG achieves the best overall performance across six question answering (QA) benchmarks. Further analysis highlights its strong reflective ability to recognize errors and revise its reasoning trajectory. Our study enhances LRMs' factuality while effectively integrating robust reasoning for Retrieval-Augmented Generation (RAG).**

***Index Terms*—Multi-hop QA, Open-domain QA, Large language model.**

I. Introduction

Large language models (LLMs) [1] demonstrate strong capabilities across complex tasks [2]. However, their reliance on parametric knowledge limits their performance on QA tasks that demand factual reasoning beyond the capacity of the model's internal parametric knowledge.

To enhance LLMs' factuality, Retrieval-Augmented Generation (RAG) [3], [4] offers a promising solution by integrating external knowledge, but struggles with forming precise retrieval queries [5]. Iterative retrieval approaches [6], [7] aim to formulate more precise queries by decomposing the input into chains of sub-questions, yet they may lead to error propagation, which degrades overall answer quality [8].

To address this limitation, Search-o1 [9] leverages recent Large Reasoning Models (LRMs) to iteratively retrieve documents and generate sub-answers through a Reason-in-Documents module, exploiting the LRM's inherent reasoning and instruction-following capabilities. However, effective reasoning with retrieval remains challenging due to (1) unreliable retrieval token generation, (2) information extraction failures, and (3) overthinking in multi-hop QA [10].

In response to this challenge, recent works such as R1-Searcher [11] and ReSearch [12] adopt reinforcement learning (RL) to incentivize LLMs to perform reasoning before retrieval. However, such policy optimization approaches involve complex training paradigms and require extensive computational resources.

In this paper, we propose **ReaRAG**, a factuality-enhanced **rea**soning model for **RAG** that achieves comparable performance without relying on complex RL-based training. Instead of resource-heavy policy optimization, we strategically distill the reasoning capabilities of LRMs, including search decisions and reflective corrections, to construct knowledge-guided reasoning chains for supervised fine-tuning (SFT) under the Thought-Action-Observation paradigm. During inference, ReaRAG iteratively performs the `search` action and dynamically decides when to trigger the `finish` action, enabling controlled iterative retrieval. Guided by external knowledge, ReaRAG reflects on its reasoning trajectory, detects errors, and realigns its reasoning toward the correct path, leading to strong overall performance while avoiding RL-based policy optimization.

To validate our approach, we conduct experiments on six benchmarks covering both multi-hop and single-hop QA, where ReaRAG achieves the highest overall performance. In summary, our contributions are as follows:

- **Enhancing LRMs' factuality through knowledge-guided reasoning chain**. We propose ReaRAG-9B, fine-tuned on a dedicated dataset to enable knowledge-guided reasoning with reliable external knowledge access. ReaRAG reflects on prior steps, leverages external knowledge to identify errors and refines its reasoning, showcasing robust multi-step reasoning capabilities.
- **A simple alternative to RL-based RAG**. Compared with Search-o1, ReaRAG generally reduces redundant retrieval in multi-hop QA, while replacing RL-based optimization with a

* Equal contribution.
† Corresponding author: houlei@tsinghua.edu.cn, lijuanzi@tsinghua.edu.cn

simpler supervised pipeline.

- **Enhanced benchmark performance**. ReaRAG performs on par with or surpasses previous methods on MuSiQue [13], HotpotQA [14], and IIRC [15], while outperforming them on harder tasks such as FRAMES [16] and FanOutQA [17].

II. Related Work

a) Reasoning-enhanced LLMs: Numerous works have investigated how to elicit the reasoning abilities of LLMs. Early approaches, such as Chain-of-Thought (COT) [18], ReAct [19] and Tree of Thought (ToT) [20] prompt LLMs to generate human-like step by step reasoning chains, though they still struggle on complex reasoning tasks. Recent advances of LRMs have scaled up CoT through RL [21], enabling models to generate long CoT before providing a final answer. Notably, LRMs including OpenAI's o1 [22], Qwen's QwQ-32B[1] [23] and DeepSeek-R1 [24] demonstrate impressive performance in complex reasoning tasks. Despite these advancements, most LRMs lack explicit access to external knowledge, limiting factual accuracy.

b) Retrieval-Augmented Generation for Reasoning: RAG has emerged as a promising paradigm for improving LLMs' factuality. Early methods rely on a single retrieval [3], [25], which often falls short for multi-hop QA tasks due to imprecise query and noisy retrieval [26]. To address these limitations, iterative methods [7], [27] propose to progressively retrieve relevant documents for final answer. However, these approaches lack a strong reflection mechanism to recover from earlier reasoning errors. To mitigate this issue, Search-o1 [9] introduces a document reasoning module and prompts a LRM to generate special tokens that trigger knowledge retrieval. However, this approach relies on the instruction-following and inherited reasoning ability of the base model, leading to formatting errors, extraction failures, and overthinking on simple questions. R1-Searcher [11] and ReSearch [12] adopt an RL-based method that incentivizes LLMs to reason before retrieving. In contrast to the complex and often unstable RL-based training, our approach achieves comparable performance through strategic distillation, offering simpler training pipeline and practical usability.

III. Methodology

In this section, we first formalize the task, then present our novel approach as illustrated in Figure 1. The implementation details are available online[2].

A. Task formulation

We focus on the multi-hop QA task, where iterative reasoning improves answer accuracy. Given a question x, our goal is to construct a knowledge-guided reasoning chain $\mathcal{C}$ to enhance the factual correctness of the generated answer $\hat{y}$. Specifically, the reasoning chain is formulated as a variable-length sequence of N steps, where each step t consists of a reasoning thought τ_t, an action α_t, and an observation o_t:

$$\mathcal{C} = \{(\tau_t, \alpha_t, o_t)\}_{t=1}^{N}, \quad 1 \leq t \leq T_{max} \tag{1}$$

The total number of reasoning steps N is not predefined and is determined by the first step at which the model emits a `finish` action. Otherwise, the process stops at T_{max}. We define the action space as $\mathcal{A} = \{\texttt{search()}, \texttt{finish()}\}$, where `search` takes a query as input and `finish` outputs the answer. At each step t, the model formulates a query based on τ_t and invokes `search` over the RAG engine $\mathcal{R}$. This process grounds the final answer $\hat{y}$ in retrieved knowledge through an iterative and structured reasoning process, thereby enhancing the factual reliability of LRMs.

B. Knowledge-guided reasoning chain generation

While existing LRMs demonstrate strong reasoning capabilities, they often fail to ground their reasoning process in factual knowledge. To make external knowledge accessible, we design a structured reasoning step where each step consists of a reasoning thought τ_t, an action α_t, and an observation o_t.

- Reasoning thought τ_t: Represents the model's thought process where it reflects on prior actions and observations before deciding the next action and its input parameter.
- Action α_t: A JSON dictionary containing an action sampled from the action space $\mathcal{A}$ along with the corresponding input parameter.
- Observation o_t: Feedback from action α_t, guiding subsequent reasoning.

a) Data Construction: As detailed in Algorithm 1, for each seed question x_i, we prompt the LRM $\mathcal{M}_{LRM}$ with $\mathcal{P}_d$ (see our code[3]) to generate reasoning thoughts and actions. Search query is then executed against the RAG engine $\mathcal{R}$ to obtain an observation o_t. The process repeats until `finish` is emitted or T_{max} is reached.

b) Data Filtering: To ensure high-quality reasoning chains, we apply data filtering by comparing the final answer $\hat{y}_i$ derived from the reasoning chain $\mathcal{C}_i$ against the ground truth answer y_i using the F1 metric. Reasoning chains with an F1 score of 0 are discarded to maintain data integrity.

c) Self-Reflection Injection: The ability to recognize and correct errors in previous reasoning steps is crucial for improving reasoning robustness, which we refer to as self-reflection in this paper. To induce this behavior, we explicitly instruct the LRM to review and revise prior steps when necessary and provide reflection exemplars during data construction. After data filtering, approximately 79.2% of the resulting trajectories exhibit self-reflective behavior.

C. Factuality-enhanced LRMs: ReaRAG

a) Fine-tuning: To incorporate knowledge-guided reasoning ability into the model, we perform SFT on the constructed dataset discussed in the previous section. Each sample is a sequence of conversation chain $\mathcal{S} =$

[1]For simplicity, all mentions of QwQ-32B in this paper refer to QwQ-32B-Preview.

[2]https://anonymous.4open.science/r/ReaRAG

[3]https://anonymous.4open.science/r/ReaRAG/src/prompts.py

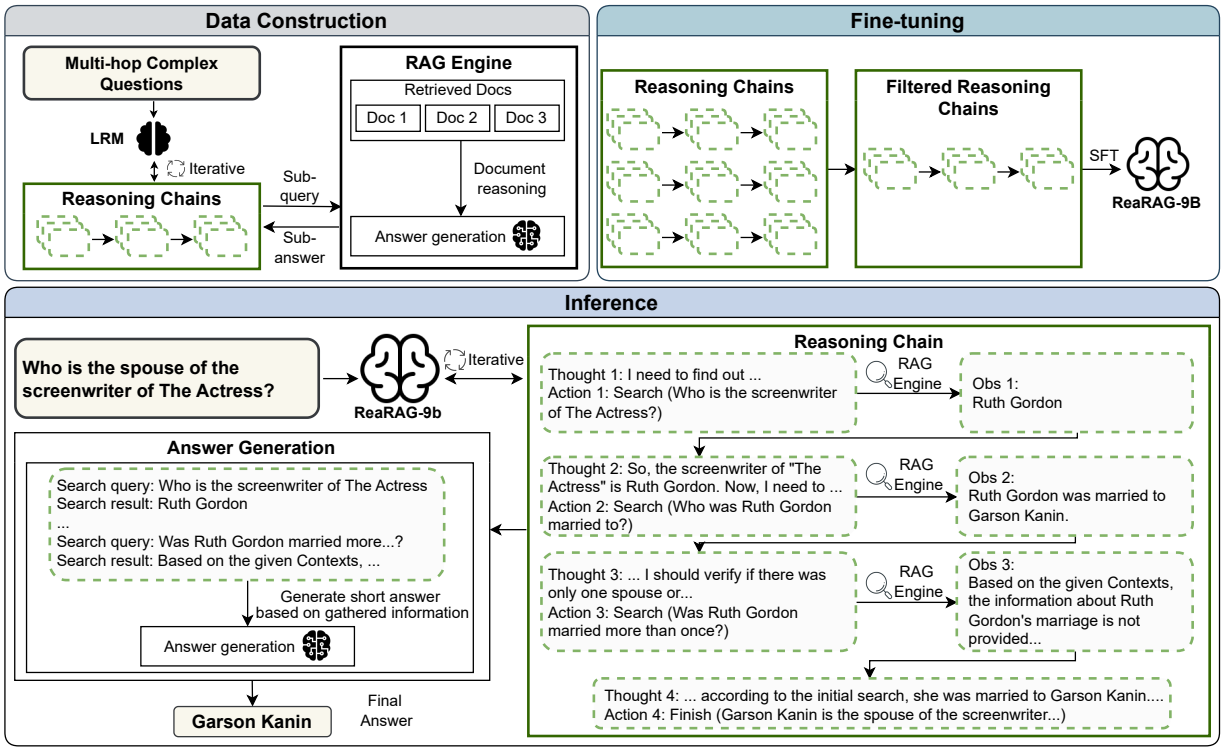

Fig. 1. Overview of our approach to develop a factuality-enhanced reasoning model **ReaRAG**. To equip ReaRAG with knowledge-guided reasoning ability, we propose an automated data construction approach (Algorithm 1), then fine-tune ReaRAG on the constructed dataset to conduct reasoning iteratively following the Thought-Action-Observation Paradigm. Pseudocode for the inference is shown in Algorithm 2.

$\{\mathcal{P}, x_i, \{(\tau_t, \alpha_t, o_t)\}_{t=1}^{N}\}$, where $\mathcal{P}$ denotes the instruction prompt described in our code. We fine-tune our factuality-enhanced LRM, ReaRAG ($\mathcal{M}_{ReaRAG}$) using the following loss function:

$$L = -\sum_{j} \mathbf{1} \times \log \mathcal{M}_{ReaRAG}(s_j \mid s_{<j}) \tag{2}$$

where s_j represents the textual tokens of the input sequence $\mathcal{S}$, $\mathbf{1}(\cdot)$ is a loss mask indicator, which is set to True on thought τ_t and action α_t tokens, ensuring that the loss is only computed over the tokens contributed to the thought reasoning as well as action, rather than the entire sequence.

b) Inference: After fine-tuning, ReaRAG is equipped with advanced reasoning capabilities to solve multi-hop QA. Given an instruction prompt $\mathcal{P}$ and a question x, the model first generates a reasoning thought τ_0 and an initial action α_0, typically a `search` action. The search query q_s is extracted and executed over the RAG engine $\mathcal{R}$, which returns an observation o_0. The triplet (τ_t, α_t, o_t) is appended to the reasoning chain $\mathcal{C}$, while the pair (q_s, o_t) is appended to the summary chain Sum. This process iterates until ReaRAG decides on a `finish` action at step N. Upon this action, we concatenate all gathered information in Sum and prompt an answer model $\mathcal{M}_{Ans}$ to synthesize the final answer $\hat{y}$ using the prompt $\mathcal{P}_{ans}$ described in our code. The pseudocode for the inference stage is provided in Algorithm 2.

IV. Experiments

A. Experimental setup

a) Dataset and metrics: We evaluate our approach on multi-hop QA benchmarks requiring reasoning across multiple documents, including MuSiQue (MQ) [13], HotpotQA (HP) [14] and IIRC [15], as well as recently proposed benchmarks, including FRAMES (FR) [16] and FanOutQA (FQ) [17]. To evaluate single-hop reasoning capabilities, we additionally include NQ [28]. Since these datasets require open-ended answers, traditional metrics such as exact match (EM) may fail to capture semantically equivalent responses. Hence, we include the LLM-as-a-Judge metric (ACC_L) [29] based on GPT-4o for semantic evaluation using a prompt $\mathcal{P}_{judge}$. For FanOutQA, we follow the metric used in the original benchmark and report the Rouge-L (R-L) score instead of EM. We randomly sample 500 validation examples each from MuSiQue, HotpotQA, and NQ, 200 from IIRC, and 396 from FRAMES. For FanOutQA, we evaluate on the full development set, which contains 310 examples.

b) Baselines: We compare our approach against multiple baselines, categorized based on their access to external knowledge. These include *in-context* retrieval, where the corpus is directly appended to the language model's context; *vanilla RAG*, which performs a single retrieval based on the original input question; and state-of-the-art *advanced RAG* methods proposed recently.

TABLE I

MAIN EXPERIMENTAL RESULTS ON SIX BENCHMARKS. **BOLD** AND UNDERLINE DENOTE THE BEST AND SECOND BEST RESULTS. WE REPORT EM SCORES AND ACC_L (LLM-AS-A-JUDGE WITH GPT-4O). FOR FANOUTQA, WE FOLLOW THE BENCHMARK AND REPORT THE ROUGE-L (R-L). OUR MODEL, **REARAG-9B**, ACHIEVES THE BEST OVERALL RESULTS, VALIDATING THE EFFECTIVENESS OF OUR STRATEGIC DISTILLATION APPROACH EVEN AGAINST RECENT RL-BASED RAG BASELINES.

Model	Multi-hop										Single-hop		Average
	MuSiQue		HotpotQA		IIRC		FRAMES		FanOutQA		NQ		
	ACC_L	EM	ACC_L	EM	ACC_L	EM	ACC_L	EM	ACC_L	R-L	ACC_L	EM	ACC_L
In-context													
GLM-4-9B (128k)	27.4	18.6	59.9	44.6	-	-	-	-	-	-	47.4	31.2	-
GLM-4-32B (128k)	33.3	20.4	65.6	48.6	-	-	-	-	-	-	58.0	30.8	-
Vanilla RAG													
GLM-4-9B (128k)	29.7	20.2	65.3	50.8	27.3	22.0	11.0	7.3	4.2	15.6	46.1	34.0	30.6
GLM-4-32B (128k)	31.9	19.4	61.3	46.4	26.0	14.5	13.9	7.6	3.9	12.3	53.8	33.6	31.8
QwQ-32B	39.3	25.6	66.7	46.2	32.8	25.0	23.4	13.9	9.7	14.2	56.1	32.6	38.0
Advanced RAG													
Self-RAG-7B	18.7	12.8	48.4	32.8	21.8	12.0	8.8	5.6	3.9	14.2	47.1	36.6	24.8
SearChain-7B	26.3	17.6	44.8	33.6	25.8	18.5	16.0	10.6	4.8	17.7	22.5	9.4	23.4
Search-o1-32B	41.9	31.2	55.6	43.2	32.8	29.5	27.8	19.2	**12.3**	14.9	46.0	33.0	36.0
R1-Searcher-7B-Base	61.1	49.8	**77.1**	**64.0**	**42.5**	**35.5**	26.9	18.7	5.8	15.6	52.8	40.0	44.4
ReSearch-Qwen-7B-Instruct	63.3	**51.0**	74.0	60.0	41.8	33.5	29.4	21.0	6.8	15.8	56.0	**42.0**	45.2
Ours													
ReaRAG-7B	**64.7**	45.6	73.9	55.4	38.5	25.5	**34.9**	**23.2**	10.0	**25.4**	**59.7**	38.2	46.9
ReaRAG-9B	64.4	45.6	73.7	53.6	41.0	27.5	34.2	21.2	11.6	24.3	58.7	37.4	**47.3**

In the *in-context* settings, we use GLM-4-9B and GLM-4-32B [30], both with a 128k context window. The *vanilla RAG* baselines employ the same models, along with QwQ-32B [23]. For *advanced RAG* baselines, we include Self-RAG [31], which fine-tunes Llama-2-7B to retrieve documents on demand and filter noisy evidence; SearChain [27], which generates a Chain-of-Query structure enabling multi-turn retrieval with verification; and Search-o1 [9], which proposes a framework for LRMs to perform iterative knowledge retrieval. We further consider R1-Searcher [11] and ReSearch [12], which encourage reasoning before retrieval via RL.

B. Implementations details

a) Data construction and fine-tuning: The seed dataset described in Algorithm 1 is derived from the training sets of MuSiQue, HotpotQA, and NQ, with QwQ-32B as the LRM. To maintain the model's general capabilities, we fine-tune GLM-4-9B [30] using the constructed dataset (roughly 20k samples), and the general SFT dataset from GLM-4 [30]. For fair comparison, we also train a 7B variant, ReaRAG-7B, based on Qwen2.5-7B [32].

b) Evaluation: For MuSiQue and HotpotQA, we use the original corpora from the respective authors. For NQ, we follow the corpus setup of "Lost in the middle" [33]. To increase the difficulty, especially when comparing with long-context LLMs, we scale up the number of distractors, yielding corpora of 48k-58k token lengths. Thus, this design demands high-quality queries to enhance retrieval quality.

For IIRC, FRAMES, and FanOutQA, where official corpora are incomplete or unavailable, we retrieve the top 5 most relevant Wikipedia snippets via the Google Serper API. These benchmarks are excluded from the *in-context* settings evaluations due to the lack of official corpora. All baselines are evaluated using their official implementations with our RAG engine and corpus setup.

c) RAG engine: Our RAG engine consists of two main components: retrieval and generation. For retrieval, we use Zhipu's embedding-3 model[4] with a GLM3-based reranker to enhance retrieval. This setup is applied only to benchmarks with self-constructed local corpora, while for IIRC, FRAMES, and FanOutQA, documents are retrieved via Google API. We use GLM-4-32B with a 128k context window to generate responses based on the retrieved documents.

C. Main results

Table I reports main results on six benchmarks. Overall, ReaRAG-9B achieves the highest average ACC_L of 47.3. Despite the rise of RL-based RAG models (R1-Searcher, ReSearch), ReaRAG-9B trained solely with SFT matches or surpasses them on MuSiQue, HotpotQA, IIRC, and NQ, and exceeds them on FRAMES and FanOutQA (by 5.8%-7.3% ACC_L), suggesting SFT alone can yield equally strong performance for open-ended QA. Our 7B variant, ReaRAG-7B, achieves equally strong performance, indicating model-agnostic effectiveness.

However, the gap between EM and ACC_L (e.g., on NQ: 58.7 vs. 52.8 ACC_L but 37.4 vs. 40.0 EM) highlights that EM often fails to capture semantically correct answers, motivating the use of ACC_L. We use the same judge model and prompt across all compared methods for fair comparison.

Notably, the original SearChain results rely on GPT-3.5, a proprietary large-scale model. For a fair comparison, we

[4]https://bigmodel.cn/dev/api/vector/embedding

Algorithm 1 Data Construction

Input: Seed Dataset $\mathcal{D}_{\text{seed}}$, Large Reasoning Model $\mathcal{M}_{LRM}$, Instruction prompt $\mathcal{P}_d$, Max iterations T_{max}, RAG engine $\mathcal{R}$
Output: Dataset $\mathcal{D}_{\text{reason}}$ with reasoning chains

Initialize $\mathcal{D}_{\text{reason}} \leftarrow \emptyset$
for each $(x_i, doc_i) \sim \mathcal{D}_{\text{seed}}$ **do** ▷ Sample question and gold documents
  $t \leftarrow 0$, $\mathcal{C}_i \leftarrow []$ ▷ Iteration counter t and Reasoning chain $\mathcal{C}_i$
  while $t < T_{\text{max}}$ **do**
    $y'_t \leftarrow \mathcal{M}_{LRM}([\mathcal{P}_d \oplus x_i \oplus \mathcal{C}_i])$ ▷ Generate response
    $(\tau_t, \alpha_t) \leftarrow \texttt{parse}(y'_t)$ ▷ Extract thought τ_t and action α_t
    Get action type $\alpha_{t_{\text{type}}} \leftarrow \alpha_t$['type']
    if $\texttt{finish} \in \alpha_{t_{\text{type}}}$ **then**
      Append (τ_t, α_t) to $\mathcal{C}_i$
      break
    else if $\texttt{search} \in \alpha_{t_{\text{type}}}$ **then**
      $q_s \leftarrow \alpha_t$['query'], $o_t \leftarrow \mathcal{R}(q_s, doc_i)$ ▷ Get o_t from RAG engine
      Append (τ_t, α_t, o_t) to $\mathcal{C}_i$
    end if
    $t \leftarrow t + 1$
  end while
  Append $\mathcal{C}_i$ to $\mathcal{D}_{\text{reason}}$
end for
return $\mathcal{D}_{\text{reason}}$

Algorithm 2 Inference

Input: Input question x, documents doc, ReaRAG $\mathcal{M}_{ReaRAG}$, Answer LLM $\mathcal{M}_{Ans}$, Instruction prompt $\mathcal{P}$, Answer Prompt $\mathcal{P}_{ans}$, Max iterations T_{max}, RAG engine $\mathcal{R}$
Output: Final answer, $\hat{y}$

$t \leftarrow 0$, $\mathcal{C} \leftarrow []$, $Sum \leftarrow []$ ▷ Initialization
while $t < T_{\text{max}}$ **do**
  $y'_t \leftarrow \mathcal{M}_{ReaRAG}([\mathcal{P} \oplus x \oplus \mathcal{C}])$ ▷ Generate response for iteration t
  $(\tau_t, \alpha_t) \leftarrow \texttt{parse}(y'_t)$ ▷ Extract thought τ_t and action α_t
  Get action type $\alpha_{t_{\text{type}}} \leftarrow \alpha_t$['type']
  if $\texttt{finish} \in \alpha_{t_{\text{type}}}$ **then**
    $\hat{y} \leftarrow \mathcal{M}_{Ans}([\mathcal{P}_{ans} \oplus x \oplus Sum])$
    return final answer $\hat{y}$
  else if $\texttt{search} \in \alpha_{t_{\text{type}}}$ **then**
    $q_s \leftarrow \alpha_t$['query'], $o_t \leftarrow \mathcal{R}(q_s, doc)$ ▷ Get o_t from RAG engine
    Append (τ_t, α_t, o_t) to $\mathcal{C}$
    Append (q_s, o_t) to Sum
  end if
  $t \leftarrow t + 1$
end while

replace it with Qwen2.5-7B-Instruct [32], under which performance drops across all benchmarks. Similarly, Self-RAG, despite aiming to improve retrieval quality, shows limited gains due to lack of multi-turn retrieval. Search-o1, although leveraging QwQ-32B, a strong reasoning LRM to perform multi-turn retrieval, only performs on par with *vanilla RAG* settings. Further analysis is provided in Section IV-E.

D. Ablation

a) Closed-book performance: We conduct a closed-book experiment to evaluate the parametric knowledge of the language models. The results, presented in Table II show that QwQ-32B outperforms GLM-4 on multi-hop benchmarks requiring strong reasoning, except for HotpotQA and FanOutQA. Nevertheless, their parametric knowledge remains insufficient compared to the results in Table I.

b) Advantage of strong reasoning: To evaluate the impact of strong reasoning capabilities of LRM, we fine-tune an LLM lacking such abilities under the same Thought-Action-Observation paradigm. This variant, denoted as w/o reasoning in Table III, uses the same backbone as ReaRAG-9B. However, instead of using the reasoning-strong QwQ-32B for data construction, we employ GLM-4-9B that lacks robust reasoning skills. To compensate, we provide additional supervision during the process including the gold answer and gold decomposition as hints, using the prompt $\mathcal{P}_{ablat}$ described in our anonymous online repository.

Table III shows that ReaRAG-9B (w/ reasoning) consistently outperforms its counterpart, yielding 6-12% ACC_L gain on multi-hop benchmarks and 3.5% gain on single-hop NQ. The smaller gain on NQ suggests that strong reasoning offers limited benefit for single-hop tasks, consistent with the assumption that such questions require less complex reasoning.

E. Analysis

1) Performance against strong baseline: We conduct an in-depth analysis and compare our approach against the baselines, including Search-o1, R1-Searcher and ReSearch. Below, we highlight key factors affecting their performance.

TABLE IV
INVALID GENERATION RATES OF SPECIAL TOKENS IN QWQ-32B.

	Multi-hop					Single-hop
	MQ	HP	IIRC	FR	FQ	NQ
Invalid (%)	19.0	28.0	18.0	18.9	30.0	25.0

a) Formatting Error: QwQ-32B often fails to follow the instruction prompt and leads to formatting errors (e.g. missing retrieval token $|begin\_search\_query|$). This forces Search-o1 into a closed-book setting, severely degrading performance. Format error rates are shown in Table IV. On the other hand, ReaRAG can almost void the format error after the fine-tuning.

b) Information Extraction Failure: Search-o1 introduces the Reason-in-Documents module, leveraging QwQ-32B for

TABLE II

CLOSED-BOOK PERFORMANCE OF LANGUAGE MODELS ON MULTI-HOP AND SINGLE-HOP BENCHMARKS. THESE MODELS PERFORM BETTER ON SINGLE-HOP BENCHMARKS BUT SCORE SIGNIFICANTLY LOWER ON MULTI-HOP BENCHMARKS, HIGHLIGHTING THE LIMITATIONS OF RELYING SOLELY ON PARAMETRIC KNOWLEDGE FOR THESE BENCHMARKS.

Model	Multi-hop										Single-hop		Average
	MuSiQue		HotpotQA		IIRC		FRAMES		FanOutQA		NQ		
	ACC_L	EM	ACC_L	EM	ACC_L	EM	ACC_L	EM	ACC_L	R-L	ACC_L	EM	ACC_L
GLM-4-9B(128k)	3.5	0.0	29.5	23.0	17.0	**15.0**	5.2	3.5	7.1	18.0	27.50	16.00	15.0
GLM-4-32B(128k)	6.5	1.0	**40.0**	**28.0**	17.0	11.5	10.5	4.6	**11.9**	**22.5**	**44.5**	**25.0**	**21.7**
QwQ-32B	**11.0**	**2.0**	35.0	10.0	**21.3**	14.5	**14.4**	**9.1**	11.0	13.2	37.5	12.0	**21.7**

TABLE III

PERFORMANCE COMPARISON OF MODELS WITH AND WITHOUT STRONG REASONING CAPABILITIES. *w/ reasoning* CONSISTENTLY OUTPERFORMS *w/o reasoning* ACROSS ALL BENCHMARKS, DEMONSTRATING THE EFFECTIVENESS OF OUR FINE-TUNING PROCESS.

Model	Multi-hop										Single-hop		Average
	MuSiQue		HotpotQA		IIRC		FRAMES		FanOutQA		NQ		
	ACC_L	EM	ACC_L	EM	ACC_L	EM	ACC_L	EM	ACC_L	R-L	ACC_L	EM	ACC_L
ReaRAG-9B													
- *w/o reasoning*	57.5	38.0	69.5	50.0	31.8	22.0	26.6	19.2	5.5	21.6	52.0	29.0	40.5
- *w/ reasoning*	**67.0**	**46.0**	**81.5**	**58.0**	**41.0**	**27.5**	**34.2**	**21.2**	**11.6**	**24.3**	**55.5**	**32.0**	**48.5**

in-depth reasoning over retrieved documents to generate responses as search results. However, this module often suffers from the hallucination of the LRM and misjudges that "No helpful information"(Table V). Our analysis identifies the primary cause: the module attempts to answer the original multi-hop question using insufficient information retrieved from the sub-question. This flaw weakens Search-o1, as previous misjudgements cause reasoning paths to diverge and lead to searches for non-existent information.

c) Overthinking in multi-hop QA: Recent studies show that LRMs often overthink in multi-hop QA, generating unnecessarily long reasoning chains [10]. We examine this by comparing retrieval steps per task. Figure 2 shows that ReaRAG uses fewer steps than Search-o1 on most benchmarks except FanOutQA and NQ, indicating more efficient reasoning and retrieval with better performance. Tables V show a similar pattern: Search-o1 sometimes gets stuck after repeatedly receiving "no information found." Although ReaRAG uses more steps than RL-based baselines (ReSearch and R1-Searcher), it achieves better overall performance.

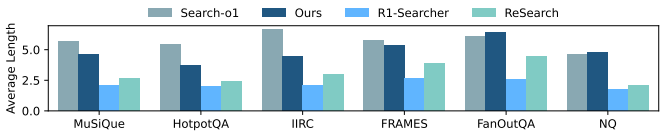

Fig. 2. Comparison of chain length across all benchmarks. We measure the number of retrievals needed to complete the task. Search-o1 requires more steps in general, highlighting the tendency of the underlying LRM to overthink in multi-hop QA tasks.

d) Poor Generalization on Harder Benchmarks: R1-Searcher and ReSearch exhibit weaker performance on FRAMES and FanOutQA, which require more complex retrieval. Both RL-based models struggle to reason effectively under noisy retrieved contexts, suggesting limited generalization to out-of-distribution QA.

2) Strength of ReaRAG: As shown in Table V, ReaRAG retrieves relevant information more accurately than the Reason-in-Documents module in Search-o1, which jointly reasons over documents and prior steps and thus is more prone to false judgments that propagate through subsequent reasoning. In contrast, ReaRAG focuses retrieval on the current sub-query, enabling cleaner evidence collection at each hop and reducing errors from noisy or incomplete results. Moreover, Table VI shows that ReaRAG can identify early mistakes and recover through corrective retrieval and reasoning. This self-reflective behavior, rarely seen in standard LLMs, emerges from fine-tuning on our self-reflective reasoning data, which enables the model to reassess intermediate steps, adjust retrieval strategies, and produce more accurate answers.

V. CONCLUSION

In this study, we present **ReaRAG**, a factuality-enhanced reasoning model capable of performing knowledge-guided reasoning. ReaRAG iteratively plans and reflects on reasoning steps, using external knowledge to validate each step in the reasoning chain. Through comprehensive evaluation across six benchmarks, ReaRAG achieves the best overall performance, outperforming RL-based methods on challenging benchmarks such as FRAMES and FanOutQA. Unlike RL-based approaches requiring complex policy optimization and high computational cost, our method offers a practically simpler alternative. Further analysis shows its robustness in handling complex multi-hop QA while generally reducing overthinking relative to Search-o1. These results underscore

TABLE V

COMPARISON OF SEARCH-O1 AND REARAG. SEARCH-O1'S REASON-IN-DOCUMENTS MODULE MAY RETURN "**NO HELPFUL INFORMATION FOUND**" WHEN THE RETRIEVED EVIDENCE IS INSUFFICIENT FOR THE ORIGINAL MULTI-HOP QUESTION, WHICH CAN MISLEAD LATER REASONING.

 # Case Study 1: Where was the author of Hannibal and Scipio educated at?
Ground truth: Exeter College
Search-o1
Alright, I've got this question: *... Simplified here.* So, I'll do a search for "Hannibal and Scipio book." <|begin_search_query|>Hannibal and Scipio book<|end_search_query|> <|begin_search_result|>**No helpful information found.**<|end_search_result|> that didn't turn up anything. Maybe the book has a different title. Let me think differently. *... Simplified here. Search-o1 reasoning diverges, unable to find related information.* *Only search queries and results are listed below for simplicity.* <|begin_search_query|>Punic Wars book<|end_search_query|> <|begin_search_result|>No helpful information found.<|end_search_result|> *... Simplified many steps. Search-o1 tries many unrelated search queries.* I've reached the maximum number of searches. Based on the information I have, I need to make an educated guess. *... Simplified here. Maximum searches reaches* Considering that Oxford and Cambridge are traditional centers for classical studies in the UK, I'll go with Oxford. Therefore, my answer is: Oxford
ReaRAG-9B
Thought 1: So I need to find out where the author of "Hannibal" and "Scipio" was educated. First, I need to know who the author is. **Action 1:** {'function': 'search', 'parameters': {'query': 'Who is the author of Hannibal and Scipio?'}} **Observation 1:** The author of "Hannibal and Scipio" is Thomas Nabbes.
Thought 2: Okay, now that I know the author is Thomas Nabbes, I need to find out where he was educated. **Action 2:** {'function': 'search', 'parameters': {'query': 'Where was Thomas Nabbes educated?'}} **Observation 2:** Thomas Nabbes was educated at Exeter College, Oxford.
Thought 3: Great, so Thomas Nabbes studied at Exeter College in Oxford. I should make sure that this the correct information. **Action 3:** {'function': 'search', 'parameters': {'query': 'Confirm the educational background of Thomas Nabbes'}} **Observation 3:** Thomas Nabbes was educated at Exeter College, Oxford in 1621. He left the university without taking a degree.
Thought 4: So, he studied at Exeter College in Oxford, but didn't finish his degree. That's interesting. I think this is the information I need. **Action 4:** {'function': 'finish', 'parameters': {'answer': 'Thomas Nabbes was educated at Exeter College, Oxford.'}}
Final Answer: Exeter College, Oxford

TABLE VI

CASE STUDY ILLUSTRATING REARAG'S SELF-REFLECTION. DESPITE INITIALLY CONFUSING THE TARGET SPECIES (E.G., *Ambystoma maculatum* VS. *Ambystoma texanum*), REARAG RECOGNIZES THE MISTAKE, PERFORMS CORRECTIVE SEARCHES, AND RECOVERS THE CORRECT ANSWER.

 # Case Study 2: What is the scientific name of an amphibian that is listed as endangered by the Canadian government (as of 2024), and its only population in Canada occurs on an island which is the southernmost inhabited part of Canada?
Ground truth: Ambystoma texanum
ReaRAG-9B
Thought 1: So I need to find the scientific name of an endangered amphibian in Canada, and its only population in the southernmost inhabited part of Canada. Okay, first I need to know what the southernmost inhabited island in Canada is. **Action 1:** {'function': 'search', 'parameters': {'query': 'What is the southernmost inhabited island in Canada?'}} **Observation 1:** The southernmost inhabited island in Canada is Pelee Island, Ontario.
... Omit some steps
Thought 9: Wait a minute, in my earlier steps, I thought it was Ambystoma maculatum, but now the search is showing Ambystoma texanum. There might be some confusion here. *Despite earlier mistakes, ReaRAG resolves the confusion by taking correct search action* **Action 9:** {'function': 'search', 'parameters': {'query': 'Scientific name of small-mouth salamander'}}
Final Answer: Ambystoma texanum

the effectiveness of strategic reasoning distillation as a practical alternative to RL-based RAG methods.

VI. ACKNOWLEDGMENTS

We would like to thank all the anonymous reviewers for their careful work and thoughtful suggestions. This work is supported by a grant from the Institute for National Natural Science Foundation of China (62476150), Guo Qiang, Tsinghua University (2019GQB0003), and Tsinghua University (Department of Computer Science and Technology)-Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIIOT).

REFERENCES

[1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. arXiv preprint arXiv:2302.13971.

[2] F. Xu, Q. Hao, Z. Zong, J. Wang, Y. Zhang, J. Wang, X. Lan, J. Gong, T. Ouyang, F. Meng, C. Shao, Y. Yan, Q. Yang, Y. Song, S. Ren, X. Hu, Y. Li, J. Feng, C. Gao, and Y. Li, "Towards large reasoning models: A survey of reinforced reasoning with large language models," 2025. arXiv preprint arXiv:2501.09686.

[3] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

[4] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proceedings of the 37th In-*

ternational Conference on Machine Learning, ICML, vol. 119, pp. 3929–3938, PMLR, 2020.

[5] C. Chan, C. Xu, R. Yuan, H. Luo, W. Xue, Y. Guo, and J. Fu, "RQ-RAG: learning to refine queries for retrieval augmented generation," 2024. arXiv preprint arXiv:2404.00610.

[6] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and narrowing the compositionality gap in language models," in *Findings of the Association for Computational Linguistics: EMNLP, pp. 5687–5711*, Association for Computational Linguistics, 2023.

[7] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," in *Findings of the Association for Computational Linguistics: EMNLP, pp. 9248–9274*, Association for Computational Linguistics, 2023.

[8] S. Cao, J. Zhang, J. Shi, X. Lv, Z. Yao, Q. Tian, L. Hou, and J. Li, "Probabilistic tree-of-thought reasoning for answering knowledge-intensive complex questions," in *Findings of the Association for Computational Linguistics: EMNLP , pp. 12541–12560*, Association for Computational Linguistics, 2023.

[9] X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou, "Search-o1: Agentic search-enhanced large reasoning models," 2025. arXiv preprint arXiv:2501.05366.

[10] X. Chen, J. Xu, T. Liang, Z. He, J. Pang, D. Yu, L. Song, Q. Liu, M. Zhou, Z. Zhang, R. Wang, Z. Tu, H. Mi, and D. Yu, "Do NOT think that much for 2+3=? on the overthinking of o1-like llms," 2024. arXiv preprint arXiv:2412.21187.

[11] H. Song, J. Jiang, Y. Min, J. Chen, Z. Chen, W. X. Zhao, L. Fang, and J. Wen, "R1-searcher: Incentivizing the search capability in llms via reinforcement learning," 2025. arXiv preprint arXiv:2503.05592.

[12] M. Chen, T. Li, H. Sun, Y. Zhou, C. Zhu, H. Wang, J. Z. Pan, W. Zhang, H. Chen, F. Yang, Z. Zhou, and W. Chen, "Research: Learning to reason with search for llms via reinforcement learning," 2025. arXiv preprint arXiv:2503.19470.

[13] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Musique: Multihop questions via single-hop question composition," *Trans. Assoc. Comput. Linguistics*, vol. 10, pp. 539–554, 2022.

[14] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 2369–2380*, Association for Computational Linguistics, 2018.

[15] J. Ferguson, M. Gardner, H. Hajishirzi, T. Khot, and P. Dasigi, "IIRC: A dataset of incomplete information reading comprehension questions," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP pp. 1137–1147*, Association for Computational Linguistics, 2020.

[16] S. Krishna, K. Krishna, A. Mohananey, S. Schwarcz, A. Stambler, S. Upadhyay, and M. Faruqui, "Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation," 2024. arXiv preprint arXiv:2409.12941.

[17] A. Zhu, A. Hwang, L. Dugan, and C. Callison-Burch, "Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, pp. 18–37*, 2024.

[18] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2022.

[19] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

[20] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems, NeurIPS*, 2023.

[21] A. Kumar, V. Zhuang, R. Agarwal, Y. Su, J. D. Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, *et al.*, "Training language models to self-correct via reinforcement learning," 2024. arXiv preprint arXiv:2409.12917.

[22] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, A. Iftimie, A. Karpenko, A. T. Passos, A. Neitz, A. Prokofiev, A. Wei, A. Tam, A. Bennett, A. Kumar, A. Saraiva, A. Vallone, A. Duberstein, A. Kondrich, A. Mishchenko, A. Applebaum, A. Jiang, A. Nair, B. Zoph, B. Ghorbani, B. Rossen, B. Sokolowsky, B. Barak, B. McGrew, B. Minaiev, B. Hao, B. Baker, B. Houghton, B. McKinzie, B. Eastman, C. Lugaresi, C. Bassin, C. Hudson, C. M. Li, C. de Bourcy, C. Voss, C. Shen, C. Zhang, C. Koch, C. Orsinger, C. Hesse, C. Fischer, C. Chan, D. Roberts, D. Kappler, D. Levy, D. Selsam, D. Dohan, D. Farhi, D. Mely, D. Robinson, D. Tsipras, D. Li, D. Oprica, E. Freeman, E. Zhang, E. Wong, E. Proehl, E. Cheung, E. Mitchell, E. Wallace, E. Ritter, E. Mays, F. Wang, F. P. Such, F. Raso, F. Leoni, F. Tsimpourlas, F. Song, F. von Lohmann, F. Sulit, G. Salmon, G. Parascandolo, G. Chabot, G. Zhao, G. Brockman, G. Leclerc, H. Salman, H. Bao, H. Sheng, H. Andrin, H. Bagherinezhad, H. Ren, H. Lightman, H. W. Chung, I. Kivlichan, I. O'Connell, I. Osband, I. C. Gilaberte, and I. Akkaya, "Openai o1 system card," 2024. arXiv preprint arXiv:2412.16720.

[23] Q. Team, "Qwq: Reflect deeply on the boundaries of the unknown," 2024.

[24] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, and S. S. Li, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. arXiv preprint arXiv:2501.12948.

[25] G. Izacard, P. S. H. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Atlas: Few-shot learning with retrieval augmented language models," *J. Mach. Learn. Res.*, vol. 24, pp. 251:1–251:43, 2023.

[26] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," in *International Conference on Machine Learning, pp. 31210–31227, ICML*, 2023.

[27] S. Xu, L. Pang, H. Shen, X. Cheng, and T. Chua, "Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks," in *Proceedings of the ACM on Web Conference 2024, WWW, pp. 1362–1373*, ACM, 2024.

[28] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: a benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 452–466, 2019.

[29] L. Zheng, W. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2023.

[30] A. Zeng, B. Xu, B. Wang, C. Zhang, D. Yin, D. Rojas, G. Feng, H. Zhao, H. Lai, H. Yu, H. Wang, J. Sun, J. Zhang, J. Cheng, J. Gui, J. Tang, J. Zhang, J. Li, L. Zhao, L. Wu, L. Zhong, M. Liu, M. Huang, P. Zhang, Q. Zheng, R. Lu, S. Duan, S. Zhang, S. Cao, S. Yang, W. L. Tam, W. Zhao, X. Liu, X. Xia, X. Zhang, X. Gu, X. Lv, X. Liu, X. Liu, X. Yang, X. Song, X. Zhang, Y. An, Y. Xu, Y. Niu, Y. Yang, Y. Li, Y. Bai, Y. Dong, Z. Qi, Z. Wang, Z. Yang, Z. Du, Z. Hou, and Z. Wang, "Chatglm: A family of large language models from GLM-130B to GLM-4 all tools," 2024. arXiv preprint arXiv:2406.12793.

[31] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," in *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

[32] Q. Team, "Qwen2.5: A party of foundation models," September 2024.

[33] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 157–173, 2024.