DeepRAG: Thinking to Retrieve Step by Step for Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown remarkable reasoning capabilities, while their practical applications are limited by severe factual hallucinations due to limitations in the timeliness, accuracy, and comprehensiveness of their parametric knowledge. Meanwhile, enhancing retrieval-augmented generation (RAG) with reasoning remains challenging due to ineffective task decomposition and redundant retrieval, which can introduce noise and degrade response quality. In this paper, we propose DeepRAG, a framework that models retrieval-augmented reasoning as a Markov Decision Process (MDP), enabling reasonable and adaptive retrieval. By iteratively decomposing queries, DeepRAG dynamically determines whether to retrieve external knowledge or rely on parametric reasoning at each step. Experiments show that DeepRAG improves retrieval efficiency and boosts answer accuracy by 26.4%, demonstrating its effectiveness in enhancing retrieval-augmented reasoning.

1 Introduction

Large Language Models (LLMs) have shown considerable promise in reasoning (Plaat et al., 2024). Nevertheless, their limitations in capacity and capabilities result in significant issues with factual hallucinations, stemming from challenges related to the timeliness, accuracy, and comprehensiveness of their parametric knowledge (Zhang et al., 2023; Huang et al., 2023). To mitigate these problems, Retrieval-Augmented Generation (RAG) has been introduced as a promising approach. By incorporating relevant information from knowledge bases or search engines, RAG enhances the factual accuracy of model responses (Zhao et al., 2024).

However, enhancing RAG with reasoning still poses several challenges (Gao et al., 2025). One significant issue is that complex queries often necessitate multi-step decomposition to establish a coherent reasoning process (Radhakrishnan et al., 2023; Guan et al., 2024). Iterative retrieval has been proposed as a solution to continuously update retrieval results, addressing the dynamic information needs that arise during the generation process (Yue et al., 2024; Wang et al., 2025). Despite this, LLMs frequently struggle to generate precise and atomic subqueries, which are essential for more effective retrieval and question decomposition (Wu et al., 2024). From the perspective of RAG, iterative retrieval should ideally generate the next atomic query based on the current question and the available information in an adaptive manner. As illustrated in Figure 1, the process flows logically from one step to the next. Specifically, the goal of finding each movie's runtime in *steps 2-4* is derived from *step 1*'s identification of the three titles of the Lord of the Rings series.

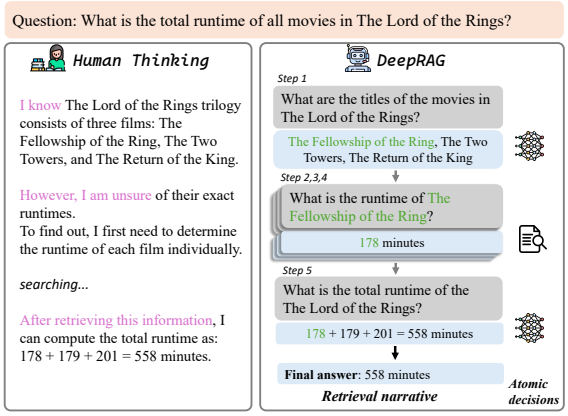

Figure 1: Correspondence between human thinking processes and DeepRAG. Specifically, *retrieval narrative* ensures a structured workflow by generating subqueries that seek additional information based on previous content, and *atomic decisions* dynamically determines whether to retrieve external knowledge or rely solely on the parametric knowledge for each subquery.

Additionally, retrieval is not always essential (Jeong et al., 2024). Some queries depend on external knowledge (*steps 2-4*), while others can be addressed through the reasoning capabilities of the LLM alone (*step 5* requires summarizing pre-

vious information). Moreover, LLMs have shown the ability to function as knowledge bases in their own right (Petroni et al., 2019) (such as in *step 1*, where the three movie titles are widely known). Unnecessary retrieval can be redundant and may introduce noise, and degrade the quality of generated responses (Chen et al., 2023; Tan et al., 2024; Yu et al., 2022).

To tackle these issues, we introduce **DeepRAG**, a new framework inspired by how humans search the Internet based on demand. This framework aims to enhance reasoning capabilities in retrieval-augmented generation by modeling the process as a Markov Decision Process. DeepRAG incorporates two main components: *retrieval narrative* and *atomic decisions*, which together create a strategic and adaptive retrieval system. As depicted in Figure 1, the *retrieval narrative* ensures a structured workflow by generating subqueries that seek additional information based on previous content. For each subquery, *atomic decisions* dynamically determines whether to retrieve external knowledge or rely solely on the LLM's parametric knowledge.

As illustrated in Figure 2, our framework consists of three components: 1) **Binary Tree Search**, which constructs a binary tree for each subquery related to the given question, exploring paths based on either parametric knowledge or external knowledge. 2) **Imitation Learning**, which extracts the reasoning process that leads to the correct final answer with minimal retrieval cost based on Binary Tree Search, enabling the model to learn the pattern of "subquery generation – *atomic decision* – intermediate answer". 3) **Chain of Calibration**, which calibrates the LLM's internal knowledge by refining each atomic decision, enabling it to make accurate *atomic decisions* about the necessity of retrieval. By explicitly enhancing the LLM's ability to recognize its own knowledge limits, we can train any model in an end-to-end manner, allowing it to dynamically decide when and what to retrieve.

We validate the effectiveness of DeepRAG across in-distribution, out-of-distribution, time-sensitive, and heterogeneous knowledge base datasets. Experimental results show that DeepRAG significantly outperforms existing methods, achieving a 21.99% increase in accuracy while also enhancing retrieval efficiency. Further analysis indicates that DeepRAG demonstrates a stronger correlation between its retrieval decisions and parametric knowledge, suggesting more effective calibration of knowledge boundaries.

2 Related Work

Adaptive Retrieval-Augmented Generation Existing adaptive RAG approaches can be broadly categorized into three types: classifier-based methods (Cheng et al., 2024; Jeong et al., 2024) requiring additional linear head training for retrieval decisions, confidence-based methods (Jiang et al., 2023; Su et al., 2024; Dhole, 2025) relying heavily on threshold-dependent uncertainty metrics, and LLM-based methods (Asai et al., 2023; Zhang et al., 2024) generating retrieval decisions but often fail to accurately recognize their knowledge boundaries, making it unreliable to delegate retrieval timing decisions to the model. Our method leverages the inherent generative capabilities of LLMs to explore knowledge boundaries in RAG settings. This design maintains the model's native generation abilities while eliminating the need for additional parameters or unreliable uncertainty metrics.

Reasoning in Retrieval-Augmented Generation Recent advances in RAG have increasingly emphasized the integration of reasoning capabilities. Search-o1 (Li et al., 2025) incorporates retrieval into inference to build an agentic system, while its application is limited to reasoning models (Chen et al., 2025). Self-RAG (Asai et al., 2023) and Auto-RAG (Yu et al., 2024) enhance reasoning through automatic data synthesis within retrieval-augmented frameworks, while AirRAG (Feng et al., 2025) combines Monte Carlo Tree Search with self-consistency techniques. These methods, however, often depend heavily on extensive retrieval or sampling overhead. More recent developments have explored reinforcement learning to enhance retrieval quality (Jin et al., 2025; Song et al., 2025; Gao et al., 2024), while these methods generally overlook retrieval efficiency in their reward function. In contrast, DeepRAG offers a flexible, end-to-end solution that enables arbitrary models to retrieve information step by step as needed, based on their evolving reasoning process.

Knowledge Boundary LLMs struggle to accurately distinguish between what they know and what they don't know (Yin et al., 2023; Kapoor et al., 2024a; Yin et al., 2024). Additional fine-tuning (Kapoor et al., 2024b) or precise probing (Cheng et al., 2024) is typically required to calibrate the model's cognition. Our approach explores knowledge boundaries in RAG settings.

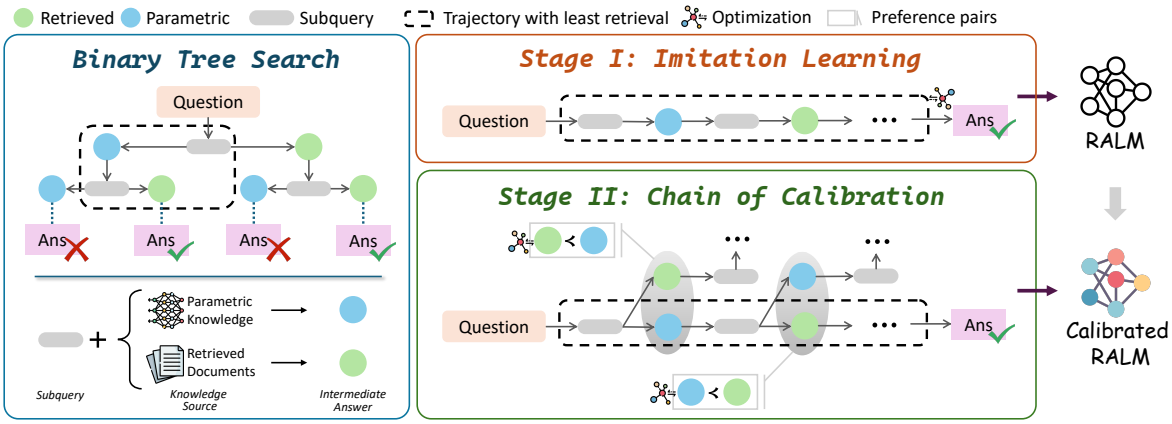

Figure 2: An overview of DeepRAG, our framework comprises three steps: (1) Binary Tree Search, (2) Imitation Learning, and (3) Chain of Calibration. Given a set of supervised datasets, we first use binary tree search to synthesize data for imitation learning, allowing the model to learn effective retrieval patterns. Next, we employ binary tree search to generate preference data, further calibrating the LLM's awareness of its knowledge boundaries.

3 Thinking to Retrieve Step by Step

In this section, we present our proposed method, DeepRAG. At its core, DeepRAG models the process of question decomposition, atomic decisions, and final answer generation as a Markov Decision Process. Given a set of supervised datasets, we first use binary tree search to synthesize data for imitation learning, allowing the model to learn effective retrieval patterns. Next, we employ binary tree search to generate preference data, further calibrating the LLM's awareness of its knowledge boundaries. In the following subsections, we provide a detailed description of each component of DeepRAG.

3.1 Overview of the MDP Modeling

We formalize the step-by-step reasoning process of retrieval-augmented generation as a Markov Decision Process (Sutton and Barto, 2018), represented by the tuple $(\mathcal{S}, \mathcal{A}, P, R)$, where $\mathcal{S}$ denotes the set of states, $\mathcal{A}$ represents the set of actions, P defines the transition dynamics, and R specifies the reward function.

States. At each step t, the state $s_t \in \mathcal{S}$ represents the partial solution to the original question. We denote $s_t = \big[x,\ (q_1, r_1),\ \ldots,\ (q_t, r_t)\big]$, where x is the input question, q_i refers to the i-th subquery, and r_i refers to the i-th intermediate answer (and any retrieved documents based on q_i).

Actions. At state s_t, the model selects an action $a_{t+1} = (\sigma_{t+1}, \delta_{t+1}) \in \mathcal{A}$, which consists of two sub-decisions:

1. *Termination decision*: Given the partial solution s_t, the model makes a binary decision $\sigma_{t+1} \in \{\texttt{continue}, \texttt{terminate}\}$ to determine whether to proceed with generating the next subquery q_{t+1} or finalize the answer o.

2. *Atomic decision*: For each subquery q_{t+1}, the model decides whether to retrieve external knowledge or rely solely on its parametric knowledge. Formally, this decision is represented as $\delta_{t+1} \in \{\texttt{retrieve}, \texttt{parametric}\}$.

Transitions. After executing the action $a_{t+1} = (\sigma_{t+1}, \delta_{t+1})$ in state s_t, the environment updates the state to s_{t+1} based on transition dynamics P.

Specifically, if $\sigma_{t+1} = \texttt{terminate}$, the process concludes by generating the final answer o, resulting in the terminal state $s_{t+1} = \big[x,\ (q_1, r_1),\ \ldots,\ (q_t, r_t), o\big]$. Otherwise, it generates the next subquery q_{t+1}.

If $\delta_{t+1} = \texttt{retrieve}$, the model retrieves documents d_{t+1} and generates an intermediate answer ia_{t+1} for subquery q_{t+1}. Otherwise, it relies on parametric knowledge to generate the intermediate answer. The response r_{t+1} is set as $[d_{t+1}, ia_{t+1}]$ (if retrieved) or ia_{t+1} (if not). The updated state is $s_{t+1} = \big[x,\ (q_1, r_1),\ \ldots,\ (q_{t+1}, r_{t+1})\big]$.

Rewards. The reward function evaluates the state based on answer correctness and retrieval cost, applied only after generating the final answer o. Formally, $R\big(s_{t+1} = s_t + [o]\big) = -C(o) \times T(s_t)$, where $C(o)$ indicates correctness (1 if correct, ∞ otherwise), and $T(s_t)$ represents the total retrieval cost in state s_t. Therefore, this reward prioritizes answer correctness while encouraging the model to reduce retrieval cost as much as possible.

3.2 Binary Tree Search

Building on this formulation, LLM iteratively decomposes a given question into subqueries, each derived from previously acquired information. The detailed generation instruction is outlined in Appendix A.2, with the answer format below.

Answer format

Question: <Question>
Follow up: <Subquery1>
Let's search the question in Wikipedia.
Context: <Paragraph Text>
Intermediate answer: <Intremediate Answer1>
Follow up: <Subquery2>
Intermediate answer: <Intermediate Answer2>
.....
So the final answer is: <Answer>

Then, we implement a binary tree search to construct reasoning paths that integrate different retrieval strategies for each subquery. As illustrated in Figure 2, given a question, the model generates the i-th subquery and explores two answering strategies: directly leveraging parametric knowledge (blue node) or retrieving external documents (green node). Therefore, we can construct a binary tree for each subquery related to the given question, exploring paths based on either parametric knowledge or external knowledge.

3.3 Imitation Learning

We present an algorithm that leverages binary trees to identify the optimal reasoning process that leads to the correct final answer while minimizing retrieval costs, corresponding to the highest reward as defined in Section 3.1. Based on the synthesized optimal reasoning data, we fine-tune the model to improve its termination and atomic decisions while enhancing its query decomposition capabilities and generating faithful intermediate answers.

Synthesizing Data As shown in Alg. 1, we employ a priority queue to maintain reasoning trajectories based on their retrieval costs. This allows us to efficiently explore potential reasoning paths by iteratively constructing and evaluating them until either finding a correct answer or exhausting all viable options within specified constraints. For instances where no correct answer can be obtained after exhausting all options, we discard them.

Through the synthesis process above, the training dataset obtained contains an adaptive reasoning process, which can be used to facilitate arbitrary LLMs in enhancing the RAG capabilities.

Algorithm 1 Data Construction for Stage I

Require: Question x, answer y, language model $\mathcal{M}$, Retriever $\mathcal{R}$, max history length T
Ensure: Optimal reasoning process s^* or $null$

Initialize priority queue $\mathcal{PQ} \leftarrow \{([x], 0)\}$ ▷ (trajectory, retrieval count)
while $\mathcal{PQ}$ is not empty **do**
  $(h, r) \leftarrow \mathcal{PQ}.dequeue()$ ▷ Get trajectory with lowest retrieval count
  $q \leftarrow \mathcal{M}(h)$ ▷ Subquery Generation
  if $ShouldAnswer(q)$ or $length(h) > T$ **then**
    $o \leftarrow \mathcal{M}(h, q)$ ▷ Final answer
    if $IsEqual(o, y)$ **then return** h
  else
    $a \leftarrow \mathcal{M}(h, q)$ ▷ Direct answer
    $\mathcal{PQ}.enqueue(([h, (q, a)], r))$
    $d \leftarrow \mathcal{R}(q)$ ▷ Retrieve document
    $a \leftarrow \mathcal{M}(h, q, d)$ ▷ Retrieved answer
    $\mathcal{PQ}.enqueue(([h, (q, (d, a))], r + 1))$
return $null$

Training Objective We implement a masked loss function for the retrieved documents to prevent the model from learning irrelevant or noisy text that could negatively impact its performance. The detailed objective is shown in Appendix B.1.

3.4 Chain of Calibration

Building on the markov process in Section 3.1, we identify four key optimization aspects for DeepRAG: termination and atomic decisions, query decomposition, and intermediate answer generation. Unlike the others, *atomic decisions* require the model to recognize its own knowledge boundaries to make precise judgments.

We propose a method that dynamically optimizes atomic decisions for each subquery, rather than training LLMs on complete reasoning paths. Our approach consists of two key components: (1) synthesizing preference data to determine when retrieval is necessary, and (2) fine-tuning the LLM with this data using Chain of Calibration training to enhance its ability to make informed atomic decisions based on its internal knowledge boundaries.

Synthesizing Preference Data First, we identify an optimal path with minimal retrieval based on Alg. 1 using the model trained in Stage I. This provides the optimal atomic decision for each subquery, determining whether retrieval is necessary. From this path, we construct preference pairs for each subquery to indicate the preferred retrieval choice. For example, in Figure 2, the optimal path may suggest answering the first subquery using parametric knowledge while requiring document retrieval for the second. Accordingly, we generate preference pairs favoring parametric knowledge for

the first subquery and retrieval for the second. This process enables LLMs to learn when to retrieve external information, thereby improving its ability to maximize the use of parametric knowledge and reducing unnecessary retrievals.

Chain of Calibration Objective We fine-tune the LLM using a Chain of Calibration objective on our synthesized preference data. Given the i-th subquery and a state $s_i = [x, (q_1, r_1), \cdots, (q_{i-1}, r_{i-1})]$, we have two distince intermediate answer $r_i^1 = a_i^1$ and $r_i^2 = (d_i, a_i^2)$. Based on the synthesis process above, we can tag which r_i is preferred and optimize it. The detailed equation is shown in Appendix B.2.

4 Experiment

4.1 Datasets

We use six open-domain QA datasets for our experiments. We treat training datasets as *in-distribution*, and unseen ones as *out-of-distribution*. The in-distribution datasets include HotpotQA (Yang et al., 2018), and 2WikMultihopQA (Ho et al., 2020), and the out-of-distribution datasets consist of CAG (Pan et al., 2024), PopQA (Mallen et al., 2022), WebQuestions (Berant et al., 2013), and MuSiQue (Trivedi et al., 2022). Specifically, we employ the time-sensitive subset of CAG to evaluate temporal reasoning capabilities. Furthermore, WebQuestions is built upon Freebase to assess model robustness when information may be absent from the knowledge base.

4.2 Baselines

We use the following baselines to evaluate the performance: **CoT** (Wei et al., 2022) and **CoT***, which employ 8-shot examples extracted from the training dataset. The asterisk (*) indicates the model output was trained using the same data employed for training the DeepRAG. **CoT-Retrieve** and **CoT-Retrieve*** augment the eight examples in the context with retrieved relevant documents based on the query. **IterDRAG** (Yue et al., 2024) refers to decomposing the question and answer step by step based on in-context learning. **AutoRAG** (Yu et al., 2024) uses trained models to iteratively decompose questions and retrieve relevant documents for answering. **Search-o1** (Li et al., 2025) leverages special tokens to prompt reasoning models to autonomously invoke retrieval as needed. **UAR** (Cheng et al., 2024) employs a trained classifier to determine when to retrieve. **FLARE** (Jiang et al., 2023) and **DRAGIN** (Su et al., 2024) are confidence-based method that decide the timing of retrieval based on token importance and uncertainty. **TAARE** (Zhang et al., 2024) allows the LLM itself to determine when retrieval is needed.

4.3 Implementation Details

We train our target model on two QA datasets: HotpotQA and 2WikiMultihopQA. For imitation learning, we randomly sample 4,000 examples from each dataset. To enhance the model's question decomposition and context-based generation capabilities, we employ Qwen-2.5-72B to generate the gray (query decomposition) and green nodes (retrieved answers) in Figure 2, and use the target model to generate the blue nodes (parametric answers) for data synthesis. For chain of calibration, we sample an additional 1,000 examples from each dataset. The performance is evaluated using Exact Match (EM) and F1 score.

Following Su et al. (2024), we adopt BM25 for retrieval and Wikipedia[1] as knowledge base. For time-sensitive questions in CAG, we utilize the dataset-provided up-to-date passages as knowledge base. We selected Llama-3-8B-Instruct (Dubey et al., 2024), Qwen-2.5-7B and Qwen-2.5-32B (Yang et al., 2024) as our target model. To implement Search-o1, we employ the distillation series of DeepSeek-R1 (Guo et al., 2025), as the method depends on reasoning models.

4.4 Overall Results

The results in Table 1 demonstrate DeepRAG's superior performance and robustness across different scenarios.

DeepRAG demonstrates superior performance across most datasets via thinking to retrieve step by step. Our method consistently outperforms existing approaches across various backbones and model sizes. Compared to reasoning-based and adaptive RAG baselines, DeepRAG outperforms across all datasets, demonstrating the effectiveness of the structured *retrieval narrative* and on-demand *atomic decisions*. Specifically, the limited performance of IterDRAG highlights the necessity of learning both query decomposition and faithful answering. Confidence-based methods like FLARE struggle to determine the optimal retrieval timing due to their reliance on unstable, predefined metrics. Moreover, we observe that confidence-based methods suffer from instability, as their per-

[1] https://dl.fbaipublicfiles.com/dpr/wikipedia_split/psgs_w100.tsv.gz

Types	Methods	*in-distribution*				*out-of-distribution*								Avg
		Hotpot QA		2WikiMultihopQA		CAG		PopQA		Web Question		MuSiQue		
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	
Llama-3-8B														
Reasoning	CoT	27.20	37.75	28.20	34.85	7.17	10.41	21.20	25.33	25.20	40.56	13.70	22.97	24.54
	CoT-Retrieve	34.90	46.85	35.80	43.41	**55.45**	**64.08**	32.80	45.87	22.90	39.22	19.10	28.18	39.05
	CoT*	21.80	31.69	25.60	30.89	5.30	7.58	23.10	25.31	26.80	40.20	4.80	13.85	21.41
	CoT-Retrieve*	22.50	32.15	23.70	29.21	44.86	55.69	38.70	45.64	17.60	29.20	5.70	11.60	29.71
	IterDRAG	23.20	30.95	19.60	24.80	38.32	46.18	22.70	34.53	15.90	26.79	12.40	17.75	26.09
	Auto-RAG	25.80	36.09	23.00	30.09	49.22	59.61	27.80	42.02	17.40	32.94	19.10	28.33	32.62
	Search-o1	14.80	24.08	22.20	27.10	3.43	6.61	10.30	13.54	15.30	29.60	5.40	11.98	15.36
Adaptive	FLARE	23.80	32.88	30.30	37.45	34.89	43.45	28.80	40.61	28.80	40.61	14.50	23.57	31.64
	DRAGIN	27.60	38.05	29.10	35.68	4.05	7.18	22.60	28.53	21.20	38.72	11.80	19.97	23.71
	UAR	29.70	40.66	34.80	42.40	52.96	61.53	33.00	45.95	22.70	39.10	19.10	28.38	37.52
	TAARE	30.60	41.43	35.20	42.85	52.96	61.59	33.20	46.01	23.40	39.56	18.60	27.55	37.75
Ours	DeepRAG-Imi	35.10	46.59	47.20	52.33	50.47	59.55	**43.60**	**48.50**	30.00	41.76	22.30	**30.46**	42.32
	DeepRAG	**40.70**	**51.54**	**48.10**	**53.25**	52.96	61.92	42.50	47.80	**32.70**	**45.24**	**22.50**	30.40	**44.13**
Qwen-2.5-7B														
Resaoning	CoT	18.90	27.81	23.40	28.97	3.12	5.71	15.20	19.20	18.30	34.86	5.60	13.12	17.85
	CoT-Retreive	24.90	34.78	18.60	23.44	41.43	51.47	27.30	41.20	15.10	29.84	6.70	15.10	27.49
	CoT*	17.60	26.15	25.10	29.62	3.12	5.62	7.90	11.06	15.60	32.45	4.70	13.40	16.03
	CoT-Retrieve*	23.40	32.29	22.40	27.51	43.30	54.51	26.60	35.46	13.80	25.60	6.20	12.85	26.99
	IterDRAG	13.70	26.84	9.30	20.47	21.81	39.59	18.00	31.44	12.50	26.95	9.20	17.25	20.59
	Search-o1	11.60	16.95	22.00	25.02	3.43	4.78	4.40	7.61	7.70	19.97	2.10	7.48	11.09
Adaptive	FLARE	23.40	32.06	21.80	26.51	34.89	42.62	19.00	28.24	16.10	31.89	8.40	15.15	25.00
	DRAGIN	16.70	24.60	12.40	16.76	3.43	5.45	12.00	15.80	17.40	32.43	4.20	7.98	14.10
	UAR	24.50	34.22	23.90	28.20	34.89	43.92	27.00	40.47	16.60	32.28	7.10	15.62	27.39
	TAARE	25.30	35.03	21.30	25.67	40.81	50.78	27.00	40.92	18.20	33.14	6.90	15.46	28.38
Ours	DeepRAG-Imi	30.40	39.44	32.00	38.32	47.98	56.99	37.50	40.72	23.90	38.62	16.50	24.67	35.59
	DeepRAG	**32.10**	**41.14**	**40.40**	**44.87**	**51.09**	**59.76**	**40.60**	**43.19**	**24.20**	**38.83**	**19.50**	**32.35**	**39.00**
Qwen-2.5-32B														
Reasoning	CoT	20.6	30.62	24.4	30.94	3.12	5.42	10.9	14.45	9.7	26	9.5	18.26	16.99
	CoT-Retrieve	28.6	39.43	27.9	36.73	39.56	49.97	33.8	45.91	17.2	34.15	12.9	21.98	32.34
	Iter-DRAG	22.9	38.26	19.6	35.70	33.02	45.61	20.3	33.2	13.3	27.57	17.6	**27.8**	27.91
	Search-o1	34	45.64	29.1	35.12	19	24.35	23.1	30.69	17.9	35.11	16.4	25.6	28.00
Ours	DeepRAG	**36.2**	**46.90**	**46.3**	**50.50**	**52.02**	**61.42**	**46.3**	**49.2**	**28.1**	**43.27**	**19.60**	27.47	**42.27**

Table 1: The overall experimental results of DeepRAG and other baselines on five benchmarks. The best/second best scores in each dataset are **bolded**/underlined. DeepRAG-Imi (Stage I) and DeepRAG (Stage II) both demonstrate superior performance compared to existing methods across all test scenarios.

formance is highly sensitive to threshold selection. Meanwhile, iterative retrieval methods like Auto-RAG often fall into continuous retrieval loops when no highly relevant information is found.

DeepRAG exhibits remarkable generalization capabilities and robustness in time-sensitive and out-of-distribution settings. In the time-sensitive dataset CAG, DeepRAG performs well compared to other adaptive and reasoning retrieval methods. It is worth noting that CoT-Retrieve outperforms it on CAG. We attribute this to the core challenge of the time-sensitive setting is to trigger retrieval most of the time. Furthermore, DeepRAG achieves substantial F1 score improvements of 2.63 and 4.57 on PopQA and WebQuestions respectively, even in scenarios where relevant information may be sparse or missing from the knowledge base.

By learning from self-synthesized data, DeepRAG effectively explores knowledge boundaries while minimizing hallucination risks. TAARE often underperforms direct retrieval methods, highlighting the mismatch between its internal knowledge and verbose. Moreover, aggressive fine-tuning approaches like CoT* and CoT-Retrieve* degrade model performance by forcing the model to learn knowledge beyond its knowledge boundaries. DeepRAG carefully preserves model capabilities during fine-tuning by leveraging self-synthesized data, effectively preventing additional hallucination while maintaining performance.

5 Analysis

5.1 Retrieval Efficiency

To evaluate the efficiency of our method, we compare the average number of retrievals on the WebQuestions dataset and report the average computation time per query. The computation time is measured on an H20*8 machine. As shown in Table 2, We have the following observations: 1) DeepRAG can achieve higher accuracy with relatively lower retrieval costs, attributed to its dynamic usage of internal knowledge. 2) Confidence-based

Method	EM	Avg. Retrievals			Time
		All	Correct	Incorrect	
FLARE	28.80	0.00	0.00	0.00	2.58
DRAGIN	21.20	0.00	0.00	0.00	1.36
UAR	22.70	0.96	0.95	0.97	0.43
TAARE	23.40	0.66	0.65	0.66	0.11
IterDRAG	15.90	2.25	2.16	2.27	1.09
Auto-RAG	17.40	4.52	3.03	2.35	0.71
DeepRAG-Imi	30.00	0.43	0.13	0.56	0.67
DeepRAG	32.70	0.28	0.12	0.36	0.50

Table 2: Retrieval frequency analysis on WebQuestions across different methods. "Correct" indicates the average number of retrievals for instances where the model produced correct answers, while "Incorrect" represents the average retrievals for cases with incorrect answers. Time refers to the average seconds spent per item.

approaches demonstrate limited robustness across datasets. For instance, neither FLARE nor DRAGIN triggers retrieval under the default confidence threshold in the WebQuestions dataset. 3) Iterative retrieval-based methods typically require numerous retrieval operations. Therefore, efficient adaptive retrieval methods like DeepRAG become crucial for optimizing resource utilization while maintaining performance.

5.2 Relevance to Parametric Knowledge

In this section, we investigate the relationship between retrieval needs and internal knowledge to demonstrate how effectively *atomic decisions* explores the knowledge boundary.

Ideally, models should initiate retrieval for queries beyond their parametric knowledge while utilizing their existing knowledge for familiar queries. We use CoT results as an indicator of whether the model can answer questions using its parametric knowledge. Then, we analyze whether other adaptive retrieval methods align with this pattern of parametric knowledge utilization.

We report four metrics. F1 score and Accuracy serve as basic performance measures, while balanced accuracy and Matthews Correlation Coefficient(MCC) (contributors, 2025) are employed to account for the class imbalance between retrieval-required and retrieval-not-required cases.

As shown in Table 3, we find that: 1) DeepRAG demonstrates superior relevance performance across F1, balanced accuracy, and MCC metrics. This suggests that DeepRAG successfully identifies retrieval necessity by exploring knowledge boundary; 2) While FLARE, DRAGIN, and TAARE exhibit high accuracy scores, their relatively low

Method	F1	Acc	Balanced Acc	MCC
FLARE	0.000	0.718	0.500	0.000
DRAGIN	0.007	0.709	0.495	-0.045
UAR	0.481	**0.756**	0.648	0.341
TAARE	0.127	0.712	0.518	0.078
Iter-DRAG	0.000	0.718	0.500	0.000
Auto-RAG	0.000	0.718	0.500	0.000
DeepRAG-Imi	0.580	0.732	0.709	0.393
DeepRAG	**0.621**	0.749	**0.743**	**0.451**

Table 3: Analysis of internal knowledge utilization across different adaptive retrieval methods on 2WikiMultihopQA.

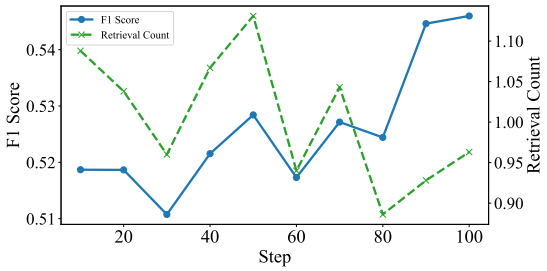

Figure 3: Experiment result and retrieval efficiency on 2WikiMultihopQA under RL setting.

balanced accuracy and MCC scores suggest they mainly succeed in retrieval-required cases but struggle to properly avoid unnecessary retrievals.

5.3 Effectiveness under RL Setting

Recently, GRPO (Shao et al., 2024) has shown promising results in enhancing model capabilities. Building upon this, we further explore the potential of DeepRAG by incorporating reinforcement learning (RL). Specifically, we initialize from DeepRAG-Imi and optimize Stage II using the GRPO objective. The detailed implementation can be found in Appendix C.5.

Figure 3 presents the training dynamics of our RL-enhanced model. The results reveal an encouraging trend: as training progresses, the F1 score on 2WikiMultihopQA gradually improves while the average number of retrievals decreases. This demonstrates that our reward design effectively guides the model to achieve better performance with more efficient retrieval behavior.

5.4 Question Decomposition Effectiveness

We systematically analyze the effectiveness of question decomposition in *retrieval narrative*. As shown in Figure 7, we present the distribution of subquery counts and retrieval attempts for different questions. Most questions require 3-5 decomposition steps, while retrieval attempts are primarily

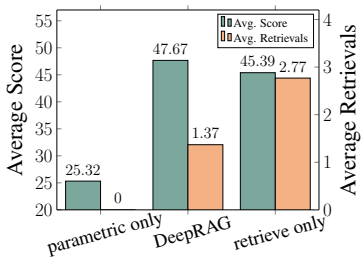

Figure 4: Comparative analysis of retrieval strategies: parametric only or retrieve only.

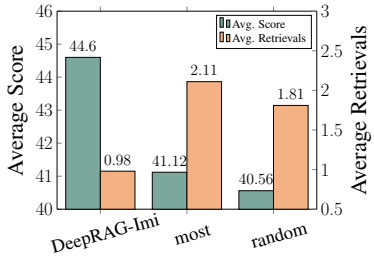

Figure 5: Average score and retrievals on the ablation study for Imitation Learning.

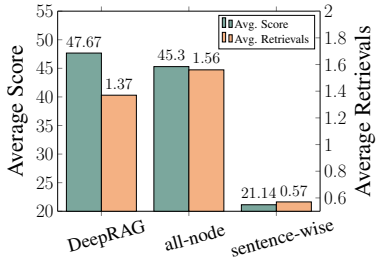

Figure 6: Average score and retrievals on the ablation study for Chain of Calibration.

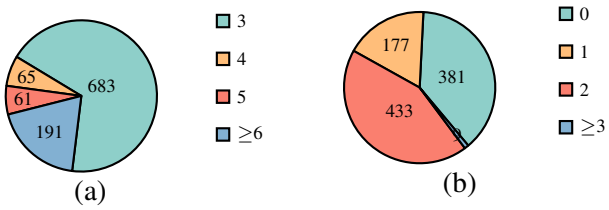

Figure 7: (a) Subquery Statistics. (b) Retrieval Statistics.

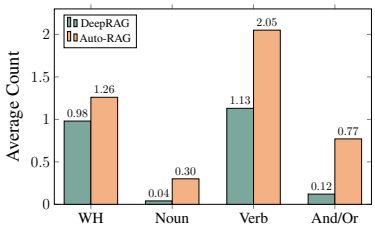

Figure 8: Average counts of WH-words, nouns, verbs, and conjunctions (and/or) per subquery.

concentrated within 0-2 rounds. This demonstrates that DeepRAG effectively decomposes questions while minimizing redundant retrieval.

Moreover, we analyze the average counts of WH-words, nouns, verbs, and conjunctions in subqueries, as shown in Figure 8. The results indicate that DeepRAG decomposes atomic queries with fewer pronouns and conjunctions.

5.5 Different Inference Strategy

To gain a deep insight into the effectiveness of *atomic decision*, we evaluate DeepRAG's performance under two extreme scenarios: relying solely on internal knowledge (retrieve only) and using retrieval in each subquery (parametric only). As shown in Figure 4, parametric only yields poor performance, while retrieve only achieves relatively higher accuracy but incurs substantial retrieval costs. DeepRAG achieves superior performance by adaptively selecting between internal and external knowledge sources.

Moreover, DeepRAG outperforms the retrieve only approach because retrieval can hinder model performance due to long context or irrelevant knowledge in certain scenarios.

5.6 Ablation Study

In this section, we conducted experiments to validate the effectiveness of DeepRAG's data construction and training process.

For Imitation Learning, we compare our default strategy of selecting paths with minimal retrieval cost against two alternative approaches: maximum retrieval cost (*most*) and random path selection (*random*). As shown in Figure 5, DeepRAG-Imi achieves lower retrieval costs and higher average performance compared to both the *most* and *random* methods.

For Chain of Calibration, we compare our default approach of constructing preferences based on nodes from optimal paths against two alternatives: constructing pairs for all nodes and constructing sentence-level partial order pairs based on retrieval efficiency. As shown in Figure 6, DeepRAG achieves lower retrieval costs while maintaining higher average performance. In contrast, the sentence-level partial order pairs learned incorrect preferences, resulting in over-reliance on internal knowledge and consequently leading to both low retrieval costs and poor performance.

6 Conclusion

In this paper, we present DeepRAG to model retrieval-augmented reasoning as a Markov Decision Process, enabling strategic and adaptive retrieval by decomposing queries into subqueries and retrieval on demand. Specifically, we develop a binary tree search method to synthesize data for imitation learning and further chain of calibration to train the model in an end-to-end manner. Experiments across various QA tasks show that DeepRAG improves retrieval efficiency while improving answer accuracy by 26.4%,demonstrating its effectiveness in optimizing retrieval-augmented reasoning.

Limitations

There are several limitations of our current DeepRAG framework, which we plan to address in the future. Firstly, we construct datasets based on the final answer accuracy using exact match score. In the future, we will expand our work to more domains like multi-turn dialogue with richer metrics like perplexity. Secondly, despite our method showing strong generalization across multi-hop factual QA, time-sensitive QA, and heterogeneous knowledge base QA, it lacks integration with external resources such as knowledge graphs and tools. We will expand our work to domains requiring diverse external information integration, including retrieved data, knowledge graph data, and tool output.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

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

Hung-Ting Chen, Fangyuan Xu, Shane Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering. *arXiv preprint arXiv:2310.12150*.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Qinyuan Cheng, Xiaonan Li, Shimin Li, Qin Zhu, Zhangyue Yin, Yunfan Shao, Linyang Li, Tianxiang Sun, Hang Yan, and Xipeng Qiu. 2024. Unified active retrieval for retrieval augmented generation. *arXiv preprint arXiv:2406.12534*.

Wikipedia contributors. 2025. Phi coefficient — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Phi_coefficient. Accessed: 2025-01-22.

Kaustubh D. Dhole. 2025. To retrieve or not to retrieve? uncertainty detection for dynamic retrieval augmented generation. *Preprint*, arXiv:2501.09292.

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

Jingsheng Gao, Linxu Li, Weiyuan Li, Yuzhuo Fu, and Bin Dai. 2024. Smartrag: Jointly learn rag-related tasks from the environment feedback. *arXiv preprint arXiv:2410.18141*.

Yunfan Gao, Yun Xiong, Yijie Zhong, Yuxi Bi, Ming Xue, and Haofen Wang. 2025. Synergizing rag and reasoning: A systematic review. *arXiv preprint arXiv:2504.15909*.

Xinyan Guan, Yanjiang Liu, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2024. Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18126–18134.

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

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller,

Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024a. Large language models must be taught to know what they don't know. *arXiv preprint arXiv:2406.08391*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. 2024b. Calibration-tuning: Teaching large language models to know what they don't know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 1–14, St Julians, Malta. Association for Computational Linguistics.

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

OpenAI. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. [Online; accessed 22-January-2025].

Ruotong Pan, Boxi Cao, Hongyu Lin, Xianpei Han, Jia Zheng, Sirui Wang, Xunliang Cai, and Le Sun. 2024. Not all contexts are equal: Teaching llms credibility-aware generation. *arXiv preprint arXiv:2404.06809*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, et al. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *arXiv preprint arXiv:2307.11768*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, 2nd edition. The MIT Press, Cambridge, MA.

Hexiang Tan, Fei Sun, Wanli Yang, Yuanzhuo Wang, Qi Cao, and Xueqi Cheng. 2024. Blinded by generated contexts: How language models merge generated and retrieved contexts for open-domain qa? *arXiv preprint arXiv:2401.11911*.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

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

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025. Chain-of-retrieval augmented generation. *arXiv preprint arXiv:2501.14342*.

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

Zhuofeng Wu, He Bai, Aonan Zhang, Jiatao Gu, VG Vydiswaran, Navdeep Jaitly, and Yizhe Zhang. 2024. Divide-or-conquer? which part should you distill your llm? *arXiv preprint arXiv:2402.15000*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and

Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language model: A different perspective on model evaluation. *arXiv preprint arXiv:2402.11493*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Tian Yu, Shaolei Zhang, and Yang Feng. 2024. Autorag: Autonomous retrieval-augmented generation for large language models.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2024. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zihan Zhang, Meng Fang, and Ling Chen. 2024. Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. *arXiv preprint arXiv:2402.16457*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

A Templates

A.1 Case Study

A.2 DeepRAG Construct Instruction

> Instruction: You are a helpful Retrieval-Augmented Generation (RAG) model. Your task is to answer questions by logically decomposing them into clear sub-questions and iteratively addressing each one.
>
> Use "Follow up:" to introduce each sub-question and "Intermediate answer:" to provide answers.
>
> For each sub-question, decide whether you can provide a direct answer or if additional information is required. If additional information is needed, state, "Let's search the question in Wikipedia." and then use the retrieved information to respond comprehensively. If a direct answer is possible, provide it immediately without searching.

B Method Details

B.1 Imitation Learning Objective

We implement a masked loss function for the retrieved documents to prevent the model from learning irrelevant or noisy text that could negatively impact its performance. In this way, we hope the model to enhance the ability to decompose subqueries and retrieve them based on demand. For each instance, the loss function is formulated as follows:

$$\mathcal{L} = -\sum_{1 \le i \le n} \log \left[\Pr(q_i | s_{i-1}) + \Pr(a_i | s_{i-1}, q_i, d_i) \right]$$

where, d_i refers to *null* if there is no reieval for ith reasoning step, n refers to the total iteration.

B.2 Chain of Calibration Objective

We fine-tune the LLM using a Chain of Calibration objective on our synthesized preference data.

Given the i-th subquery and the state $s_i = [x, q_1, r_1, \cdots, q_{i-1}, r_{i-1}]$, we have two distince intermediate answer $r_i^1 = a_i^1$ and $r_i^2 = (d_i, a_i^2)$. Based on the process above, we have known which r_i is preferred. As a result, the training objective can be formulated as follows:

$$\mathcal{L} = -\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid s_i, q_i)}{\pi_{\text{ref}}(y_w \mid s_i, q_i)} - \beta \log \frac{\pi_\theta(y_l \mid s_i, q_i)}{\pi_{\text{ref}}(y_l \mid s_i, q_i)} \right)$$

where σ is the logistic function, the hyperparameter β regulates the penalty imposed for the deviations from the base reference model π_{ref}. The terms y_w and y_l refer to the generated snippets

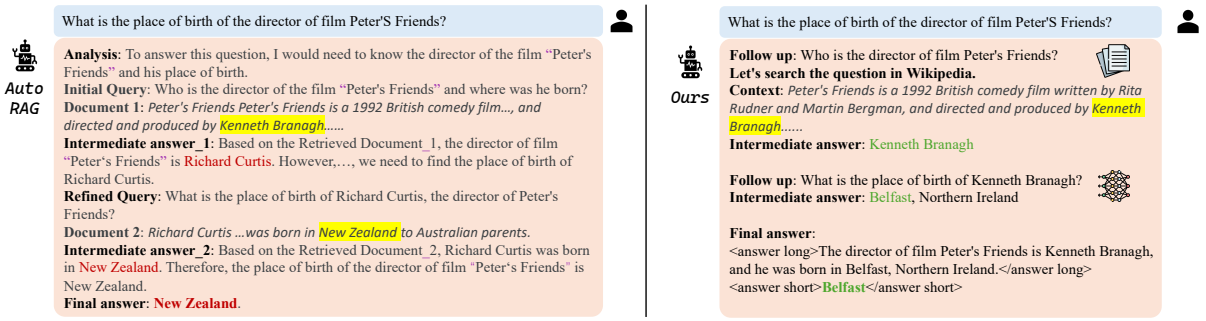

Figure 9: Case Study: Auto-RAG vs. DeepRAG. DeepRAG achieves success by atomic query decomposition, faithful intermediate answer, and adaptively using internal knowledge.

for direct answers and retrieved answers, respectively. Specifically, the snippet "Intermediate Answer:" corresponds to a direct answer, while the snippet "Let's search the question on Wikipedia" corresponds to retrieval-based answers.

C Detailed Analysis

As illustrated in Figure 9, we conduct a case study comparing DeepRAG with Auto-RAG (Yu et al., 2024), a closely related method that utilizes iterative retrieval for retrieval-augmented generation. For each subquery, Auto-RAG retrieves relevant documents and generates a corresponding subanswer. This approach is not only time-consuming but also fails when no relevant documents are retrieved. Although Auto-RAG attempts to address this issue using its own relevant documents, it falls into endless loops in most cases. In contrast, DeepRAG iteratively generates subqueries and determines whether to use internal knowledge at each iteration. The binary tree search data synthesis method for optimization ensures reliable subquery generation, intermediate answers, and final answers. Even when no related information exists in retrieved documents, the model is directed to provide a final answer based on internal knowledge.

C.1 Retrieval Efficiency

To demonstrate the efficiency of our method, we compare the average number of retrievals on 2WikiMultihopQA and WebQuestions. As shown in Table 2, We have following observations:

1) Compared to other adaptive retrieval methods, DeepRAG can achieve higher accuracy with relatively lower retrieval costs. This can be attributed to our dynamic usage of internal knowledge. Additionally, DeepRAG exhibits a positive trend in exploring relevant evidence when faced with insufficient retrieval results, as evidenced by the lower average retrieval numbers in both 2WMQA (0.92 compared to 1.25) and WQ (0.12 compared to 0.36). 2) Confidence-based approaches demonstrate limited robustness across datasets. For instance, while using identical thresholds, both FLARE and DRAGIN methods show inconsistent behaviors: they trigger approximately one retrieval per query in 2WMQA, but fail to reach the retrieval threshold entirely in WQ. This inconsistency highlights the challenge of maintaining reliable performance across different datasets using confidence-based methods. 3) Iterative retrieval-based approaches typically require numerous retrieval operations, resulting in substantial computational costs. Therefore, efficient adaptive retrieval methods like DeepRAG become crucial for optimizing resource utilization while maintaining performance.

C.2 Relevance to Parametric Knowledge

In this section, we investigate the relationship between retrieval needs and parametric knowledge to demonstrate how effectively our method explores the knowledge boundary.

Ideally, models should initiate retrieval for queries beyond their parametric knowledge while utilizing their existing knowledge for familiar queries. We use CoT results as an indicator of whether the model can answer questions using its parametric knowledge. Subsequently, we analyze whether other adaptive retrieval methods align with this pattern of parametric knowledge utilization. We evaluate the relevance using four metrics. F1 score and Accuracy serve as basic performance measures, while balanced accuracy and Matthews Correlation Coefficient(MCC) are employed to account for the class imbalance between retrieval-required and retrieval-not-required cases. The MCC ranges from -1 to 1, where a value of 1

indicates perfect correlation, 0 represents no correlation (random chance), and -1 signifies an inverse correlation.

As shown in Table 3, we find that 1) DeepRAG demonstrates superior relevance performance across F1, balanced accuracy, and MCC metrics. This suggests that DeepRAG successfully identifies retrieval necessity by exploring knowledge boundary. 2) While FLARE, DRAGIN, and TAARE exhibit high accuracy scores, their relatively low balanced accuracy and MCC scores suggest they mainly succeed in retrieval-required cases but struggle to properly avoid unnecessary retrievals.

C.3 Performance against Strong Baseline Models

We compare DeepRAG with recent strong reasoning models: QwQ-32B-preview (Team, 2024) and gpt-4o-turbo (OpenAI). As shown in Table 4, DeepRAG achieves superior average performance over QwQ and gpt-4o, particularly in time-sensitive QA tasks. While DeepRAG does not surpass gpt-4o in some cases, it achieves comparable performance levels. These results demonstrate that by adaptively leveraging retrieval, DeepRAG can achieve an equivalent level of factual accuracy to the parametric knowledge of strong reasoning models.

Models	ID	CAG	PopQA	WQ	Avg
QwQ-32B	31.43	3.43	10.60	15.10	18.40
gpt-4o-turbo	**60.6**	23.36	43.50	25.35	42.68
DeepRAG-qwen	43.00	51.09	40.60	24.20	40.38
DeepRAG-llama	52.40	**52.96**	**42.50**	**32.70**	**46.59**

Table 4: Performance against strong baseline models.

C.4 Ablation Study

In this section, we conducted experiments to validate the effectiveness of DeepRAG's data construction and training process.

Method	ID F1	CAG EM	PopQA EM	WebQuestion EM	Avg
DeepRAG-Imi	**49.46**	50.47	**43.60**	**30.00**	**44.60**
most	47.31	51.09	31.30	28.00	41.12
random	44.76	**51.40**	34.80	27.10	40.56

Table 5: Experiment results of the ablation study on the Imitation Learning Stage. ID refers to the average score of two in-distribution dataset HotpotQA and 2WikiMultihopQA.

Imitation Learning We compare our default strategy of selecting paths with minimal retrieval cost against two alternative approaches: maximum retrieval cost and random path selection. As shown in Table 5, DeepRAG-Imi enables the model to learn knowledge boundaries during the imitation learning stage. Notably, CAG performs relatively poorly at this stage due to its time-sensitive nature, which necessitates constant retrieval of up-to-date information. Moreover, as illustrated in Figure 5, DeepRAG-Imi achieves lower retrieval costs and higher average performance compared to both the maximum-retrieval-cost and random selection methods.

Method	ID F1	CAG EM	PopQA EM	WebQuestion EM	Avg
DeepRAG	**52.40**	**61.92**	**47.80**	**45.24**	**47.67**
all-node	50.92	50.47	41.50	32.70	45.30
sentence-wise	30.16	12.46	20.00	12.90	21.14

Table 6: Experiment results of the ablation study on the Chain of Calibration Stage.

Chain of Calibration We compare our default approach of constructing preferences based on nodes from optimal paths against two alternatives: constructing pairs for all nodes and constructing sentence-level partial order pairs based on retrieval efficiency. As shown in Table 6, DeepRAG demonstrates significant advantages over both variants. Specifically, as illustrated in Figure 6, DeepRAG achieves lower retrieval costs while maintaining higher average performance. In contrast, the sentence-level partial order pairs learned incorrect preferences, resulting in over-reliance on internal knowledge and consequently leading to both low retrieval costs and poor performance.

C.5 Implementation Details under RL Setting

We implement based on Search-R1 repository [2]. We adopt GRPO with a batch size of 32 and perform 8 rollouts per prompt. To avoid introducing noise, we additionally mask the retrieved text during training.

$$R_t = \begin{cases} 0, \text{answer} \times \text{ and format} \times \\ 0.1, \text{answer} \times \text{ and format} \checkmark \\ 1 - 0.1 \times \min(5, \text{retrieve\_time}_t), \text{answer} \checkmark \end{cases}$$

C.6 Retrieval Efficiency

To demonstrate the efficiency of our method, we compare the average number of retrievals on 2Wiki-

[2] https://github.com/PeterGriffinJin/Search-R1

Dataset	Method	EM	Avg. Retrievals		
			All	Correct	Incorrect
2WMQA	FLARE	30.30	0.99	1.00	0.99
	DRAGIN	29.10	1.03	1.03	1.03
	UAR	34.80	0.81	0.68	0.89
	TAARE	35.20	0.93	0.93	0.97
	IterDRAG	19.60	2.46	2.49	2.45
	Auto-RAG	23.00	6.26	4.13	1.81
	DeepRAG-Imi	47.20	1.13	0.95	1.28
	DeepRAG	48.10	1.09	0.92	1.25
WQ	FLARE	28.80	0.00	0.00	0.00
	DRAGIN	21.20	0.00	0.00	0.00
	UAR	22.70	0.96	0.95	0.97
	TAARE	23.40	0.66	0.65	0.66
	IterDRAG	15.90	2.25	2.16	2.27
	Auto-RAG	17.40	4.52	3.03	2.35
	DeepRAG-Imi	30.00	0.43	0.13	0.56
	DeepRAG	32.70	0.28	0.12	0.36

Table 7: Retrieval frequency analysis on 2WikiMultihopQA(2WMQA) and WebQuestions(WQ) across different adaptive retrieval methods. "Correct" indicates the average number of retrievals for instances where the model produced correct answers, while "Incorrect" represents the average retrievals for cases with incorrect answers.

MultihopQA and WebQuestions. As shown in Table 7, We have the following observations: 1) DeepRAG can achieve higher accuracy with relatively lower retrieval costs, attributed to its dynamic usage of internal knowledge. 2) Confidence-based approaches demonstrate limited robustness across datasets. For instance, neither FLARE nor DRAGIN trigger retrieval under the default confidence threshold in WQ. 3) Iterative retrieval-based methods typically require numerous retrieval operations. Therefore, efficient adaptive retrieval methods like DeepRAG become crucial for optimizing resource utilization while maintaining performance.