ClueAnchor: Clue-Anchored Knowledge Reasoning Exploration and Optimization for Retrieval-Augmented Generation

Hao Chen[1,2*], **Yukun Yan**[1*], **Sen Mei**[1], **Wanxiang Che**[2†], **Zhenghao Liu**[3†], **Qi Shi**[1], **Xinze Li**[3], **Yuchun Fan**[3], **Pengcheng Huang**[3], **Qiushi Xiong**[3], **Zhiyuan Liu**[1], **Maosong Sun**[1]
[1]Department of Computer Science and Technology, Institute for AI, Tsinghua University
[2]Research Center for Social Computing and Interactive Robotics, Harbin Institute of Technology
[3]School of Computer Science and Engineering, Northeastern University

Abstract

Retrieval-Augmented Generation (RAG) augments Large Language Models (LLMs) with external knowledge to improve factuality. However, existing RAG systems frequently underutilize the retrieved documents, failing to extract and integrate the key clues needed to support faithful and interpretable reasoning, especially in cases where relevant evidence is implicit, scattered, or obscured by noise. To address this issue, we propose ClueAnchor, a novel framework for enhancing RAG via clue-anchored reasoning exploration and optimization. ClueAnchor extracts key clues from retrieved content and generates multiple reasoning paths based on different knowledge configurations, optimizing the model by selecting the most appropriate reasoning path for the given context through reward-based preference optimization. Experiments show that ClueAnchor significantly outperforms prior RAG baselines in the completeness and robustness of reasoning. Further analysis confirms its strong resilience to noisy or partially relevant retrieved content, as well as its capability to identify supporting evidence even in the absence of explicit clue supervision during inference. All codes are available at https://github.com/thunlp/ClueAnchor.

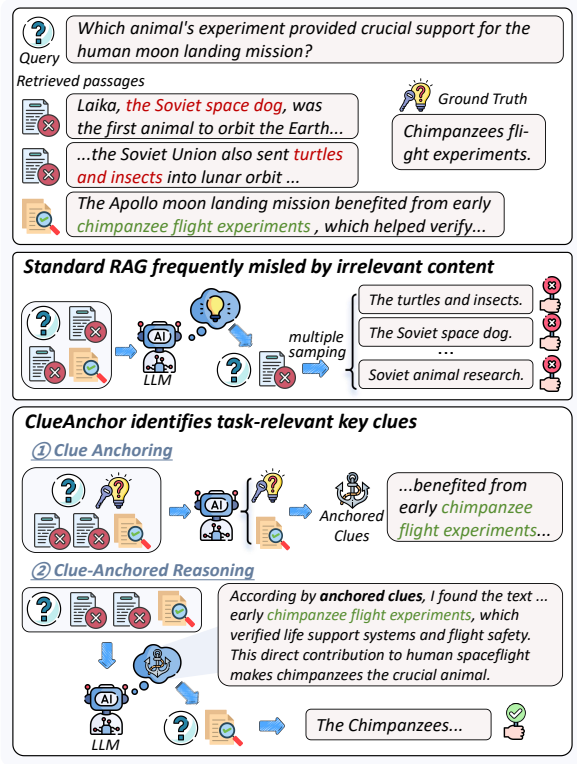


Figure 1: Illustration of ClueAnchor. Standard RAG frequently misled by irrelevant content, while ClueAnchor identifies key clues from retrieved documents and uses them to guide faithful reasoning and answer generation.

1 Introduction

Large Language Models (LLMs) have achieved remarkable progress across a wide range of NLP tasks (Achiam et al., 2023; Grattafiori et al., 2024; Yang et al., 2024). However, their reliance on static training data often results in knowledge gaps and hallucinations. Retrieval-Augmented Generation (RAG) mitigates this limitation by incorporating external evidence to enhance factual accuracy (Lewis et al., 2020a; Fan et al., 2024). Yet, since LLMs are not explicitly trained to utilize retrieved content, they often struggle to effectively incorporate and reason over external information (Lin et al., 2023).

To address this, recent work has focused on optimizing generation models to better leverage retrieved content. Lin et al. (2023) improves faithfulness via multi-task instruction tuning, while RAG-DDR (Li et al., 2024) further improves performance by sampling multiple candidate responses and optimizing toward those with higher reward signals. However, their success require the model to construct correct reasoning paths. As shown in Figure 1, relevant evidence can be implicit, disturbed by semantically similar noise, or scattered

[*] Equal Contribution.
[†] Corresponding Authors.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 19258–19278

across multiple passages. Existing RAG methods are frequently misled by irrelevant content, failing to establish robust links between retrieved evidence and the answer. This reveals a limitation of RAG systems: even when relevant information is retrieved, models may perform reasoning over incomplete or misaligned content (Barnett et al., 2024; Fayyaz et al., 2025; Chen et al., 2025b).

Motivated by this finding, we hypothesize that the ground-truth answer can serve as a backward signal to confirm correct information, thus uncovering key clues buried in the retrieved documents and facilitating the reconstruction of a correct reasoning path. Building on this intuition, we propose ClueAnchor, a novel framework that enhances RAG through clue-anchored knowledge reasoning exploration and optimization. The proposed framework consists of two components. The *Knowledge Reasoning Exploration* (KRE) module first predicts a key clue from retrieved documents conditioned on the ground truth and generates multiple reasoning paths under three configurations: internal reasoning without external context, external reasoning grounded in the retrieved evidence, and clue-anchored reasoning guided by the predicted clue. The *Knowledge Reasoning Optimization* (KRO) module evaluates these candidate paths using task-specific reward signals, and finetunes the model via preference optimization to favor the most effective one. By combining clue-aware generation with reward-guided path selection, ClueAnchor enables the model to identify key clues from noisy retrieved documents and use them as anchor for reasoning, thereby improving evidence grounding and producing more coherent and faithful outputs.

Our experiments results demonstrate that ClueAnchor outperforms all baseline models, achieving an improvement of more than 3.6% compared to the previous state-of-the-art method (Li et al., 2024). This improvement highlights ClueAnchor's ability to substantially improve answer generation quality, affirming the effectiveness of its overall design under challenging retrieval conditions. Further analysis shows that ClueAnchor maintains stable performance as retrieval noise increases, indicating strong robustness to imperfect evidence and an enhanced ability to focus on key clues despite distracting content. Moreover, ClueeAnchor generalizes beyond supervised clues, effectively identifying relevant information during inference without explicit clue guidance.

2 Related work

RAG enhances the factuality and robustness of LLMs by retrieving and incorporating external information during inference (Lewis et al., 2020b; Guu et al., 2020). However, real-world queries often involve implicit or dispersed facts scattered across multiple documents (Yang et al., 2018; Asai et al., 2019). To improve evidence coverage, prior research has focused on enhancing retrieval. Multi-hop (Li et al., 2021, 2025b) and graph-based methods (Hu et al., 2024; Wang et al., 2025b; Edge et al., 2024) identify and aggregate information from multiple sources. Memory-augmented systems (Qian et al., 2025; Wang et al., 2025a) boost efficiency via information reuse, while dynamic retrieval (Asai et al., 2023; Su et al.; Ye et al., 2024) adapts to evolving generation needs. These efforts highlight the importance of effective information access as a foundation for reliable RAG (Hwang et al., 2024).

In addition to retrieval quality, the effectiveness of RAG also depends on how well the generation model utilizes retrieved content (Shi et al., 2023). Prompting strategies like Chain-of-Note (Yu et al., 2023) guide attention to relevant context, while retrieval-aware fine-tuning (Lin et al., 2023; Soudani et al., 2024) explicitly trains models to integrate external knowledge. Other methods, such as differentiable data rewards (Li et al., 2024), address conflicts between parametric and retrieved knowledge to enhance factual consistency and reduce hallucinations. These strategies (Liu et al.) collectively align model outputs with external evidence, improving both performance and trustworthiness.

Another active research direction aims to improve transparency and reasoning quality by modeling intermediate steps. Methods such as Chain-of-Thought (Trivedi et al., 2022a; Wei et al., 2022), decomposition (Zhou et al., 2022), and self-refinement (Madaan et al., 2023) guide models to explicitly articulate their reasoning. DeepSeek-R1 (Guo et al., 2025) introduces "thought trajectories" to trace answer formation. In RAG, approaches like RADCoT (Lee et al., 2024) distill structured reasoning for greater efficiency and interpretability. These demonstrate that making reasoning explicit improves factuality and verifiability (Lightman et al., 2023; Yu and Ananiadou, 2024; Mosbach et al., 2024; Chen et al., 2025a).

While prior work has improved faithfulness and rationale alignment in RAG(Huang et al., 2025b,a), most approaches assumes relevant evidence is al-

ready available to the model and focuses on aligning generation accordingly (Menick et al., 2022; Lyu et al., 2023). In contrast, we address a more fundamental challenge: the model may still overlook critical evidence due to its implicit nature or low salience. We propose the ClueAnchor framework, which explicitly highlights clue signals from noisy retrievals to guide more grounded and interpretable reasoning.

3 Method

This section introduces the ClueAnchor framework, which is illustrated in Figure 2. We first describe the preliminaries of ClueAnchor (Section 3.1), and then detail its clue-anchored knowledge reasoning exploration and knowledge reasoning optimization process (Section 3.2).

3.1 Preliminary of ClueAnchor

RAG aims to answer a query q by leveraging both the parametric knowledge of a language model and a set of retrieved documents $D = \{d_1, \ldots, d_n\}$. The generation process can be formalized as maximizing the conditional likelihood of an answer a given the query and retrieved passages:

$$y = \arg\max_{a} \; P_\theta(a \mid q, D), \tag{1}$$

where P_θ represents the generation probability distribution parameterized by θ.

Inspired by Chain-of-Thought (CoT) prompting (Kojima et al., 2022), we adopt a reasoning-then-answering paradigm in our RAG framework, where the model jointly generates an intermediate reasoning chain r and the final answer a conditioned on the query q and retrieved documents D:

$$y = \arg\max_{r,a} \; P_\theta(r, a \mid q, D). \tag{2}$$

This CoT-based RAG reasoning process enables the model to explicitly incorporate external knowledge into intermediate reasoning, resulting in more interpretable and robust generation (Li et al., 2025a).

While multi-task instruction tuning has been used to enhance RAG models (Lin et al., 2023), it often causes overfitting to retrieved content, reducing generalization and increasing sensitivity to noise (Jin et al., 2024; Xie et al., 2024). To address this, drawing inspiration from the optimization strategy of Differentiable Data Rewards (DDR) (Li et al., 2024), ClueAnchor is fine-tuned by sampling multiple response candidates and aligning with the one achieving the highest reward score.

For each query, ClueAnchor generates a set of candidate responses $\{y_1, \ldots, y_n\}$, where each response y_i consists of a CoT-style reasoning chain and a final answer (Section 3.2.1). To guide training, we assign task-specific reward scores to each response based on the correctness of its predicted answer. The highest-rewarded response is selected as the positive sample, and the lowest-rewarded one as the negative sample, forming a preference pair for optimization (Section 3.2.2).

3.2 Clue-Anchored Knowledge Reasoning Exploration and Optimization

ClueAnchor consists of two core modules: Knowledge Reasoning Exploration (KRE) first generates multiple reasoning paths under different knowledge conditions, and then Knowledge Reasoning Optimization (KRO) ranks them by answer quality, guiding the model to prefer higher-rewarded reasoning paths.

3.2.1 Knowledge Reasoning Exploration

The Knowledge Reasoning Exploration (KRE) module models a multi-path reasoning process for each query by exploring three complementary reasoning paths. Each path reflects a distinct knowledge grounding strategy and provides diverse supervisory signals, collectively enhancing the model's ability to reason over complex and noisy evidence.

Internal Knowledge Reasoning. The model generates a response based solely on its parametric memory, simulating a no-context generation scenario:

$$y^{\text{Internal}} = \arg\max_{r,a} \; P_\theta(r, a \mid q). \tag{3}$$

This path reflects the model's parametric knowledge and helps mitigate inconsistencies that may arise from misleading retrieved content.

External Knowledge Reasoning. The model generates a response by conditioning on the retrieved passages $D = \{d_1, \ldots, d_k\}$:

$$y^{\text{External}} = \arg\max_{r,a} \; P_\theta(r, a \mid q, D). \tag{4}$$

This path reflects the standard RAG setting, where retrieved evidence often directly contains the answer, enabling efficient answer generation without requiring complex reasoning.

Clue-Anchored Knowledge Reasoning. Although retrieved documents often include sufficient evidence, models struggle to identify and use key clues when they are implicit or dispersed, leading

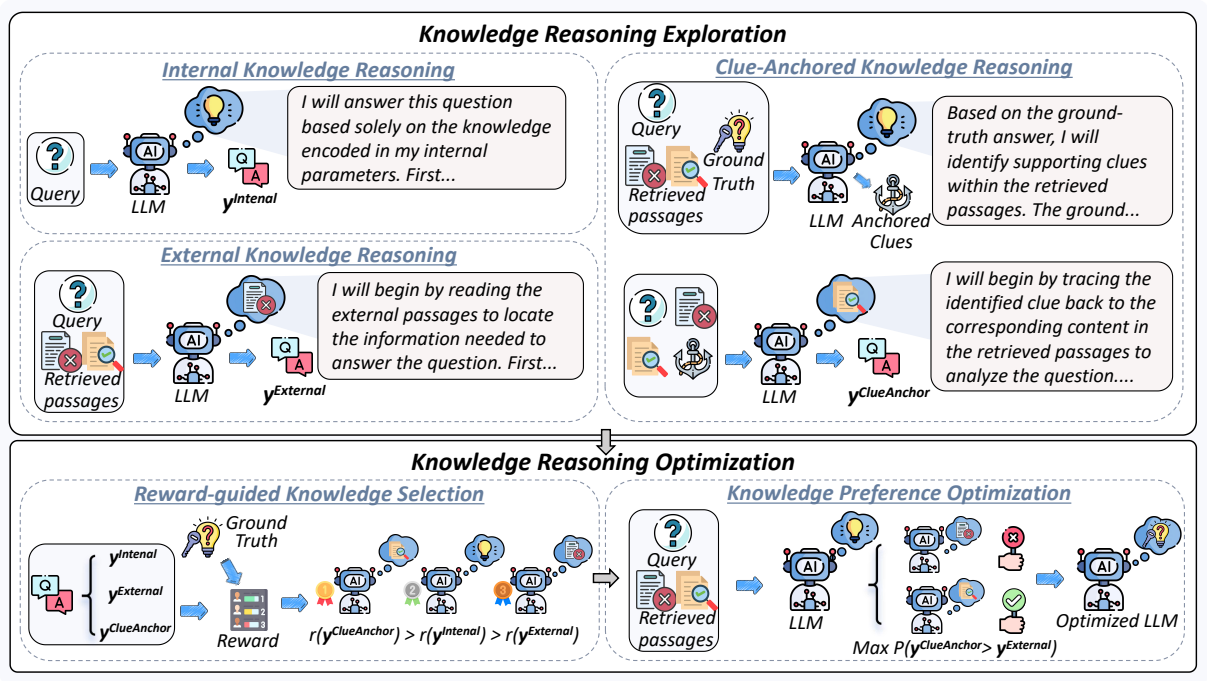


Figure 2: Overview of the ClueAnchor Framework. The Knowledge Reasoning Exploration (KRE) module generates diverse reasoning paths conditioned on different evidence scenarios. The Knowledge Reasoning Optimization (KRO) module selects and refines the most effective responses using reward-based preference signals.

to incorrect response sampling. To mitigate this, we introduce Clue-Anchored Knowledge Reasoning, which explicitly extracts key clues and guides the model's reasoning process more effectively.

Specifically, the model first predicts a clue $\hat{c}$ from the retrieved documents that supports the ground truth answer a^*:

$$\hat{c} = \arg\max_{c} \; P_\theta(c \mid q, D, a^*). \tag{5}$$

To avoid trivial copying, we explicitly prevent the model from directly restating the answer during clue generation. Each candidate clue is then validated by the backbone model to ensure that it leads to correct answer prediction. Only validated clues are retained for training.

Conditioned on the predicted clue $\hat{c}$, the model is then guided to generate a response:

$$y^{\text{ClueAnchor}} = \arg\max_{r,a} \; P_\theta(r, a \mid q, D, \hat{c}). \tag{6}$$

This clue-anchored generation process enhances the model's ability to extract and ground its reasoning in relevant evidence, especially in noisy retrieval scenarios. It also improves the quality of sampled responses, providing more effective training signals and promoting faithful reasoning. By anchoring on informative clues, the model learns to focus on meaningful information—much like finding a needle in a haystack.

3.2.2 Knowledge Reasoning Optimization

Each reasoning path reflects a distinct reasoning strategy. The internal reasoning path enables the model to quickly generate responses from parametric memory when relevant knowledge is present, but tends to hallucinate when encountering unfamiliar or unseen questions (Sun et al., 2024). The external knowledge reasoning path performs well when retrieved evidence clearly supports the answer, but is susceptible to distraction from irrelevant content in noisy retrieval scenarios (Yoran et al., 2023). The clue-anchored reasoning path strengthens the model's ability to leverage external knowledge by guiding reasoning with key clues, but may lead the model to overly focus on locating answers explicitly from retrieved content.

While the Knowledge Reasoning Exploration module generates diverse reasoning paths, not all are equally appropriate or reliable for a given query. To effectively utilize this diversity, we introduce the Knowledge Reasoning Optimization module, which refines the model's decision-making by learning to prefer higher-quality reasoning paths through reward-based supervision.

Reward-guided Knowledge Selection. To identify the most effective reasoning path, we compute a task-specific reward score $r(a_i, a^*)$ for each

response y_i by comparing its predicted answer a_i against the ground-truth answer a^*. The reasoning path with the highest reward is selected as the positive sample y^+, and the one with the lowest reward as the negative sample y^-:

$$y^+ = \arg\max_{y_i} \; r(a_i, a^*),$$
$$y^- = \arg\min_{y_i} \; r(a_i, a^*), \tag{7}$$

where $r(a_i, a^*)$ is a reward function that measures the quality of the generated answer relative to the ground truth.

Knowledge Preference Optimization. To guide the model toward better reasoning strategies, we apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to fine-tune the model by maximizing the preference for the positive response y^+ over the negative one y^-. The contrastive learning objective is defined as:

$$\mathcal{L}(\theta; \theta^{\text{ref}}) = -\mathbb{E}\Bigg[\log \sigma\Big(\beta \log \frac{P_\theta(y^+ \mid q, D)}{P_{\theta^{\text{ref}}}(y^+ \mid q, D)} - \beta \log \frac{P_\theta(y^- \mid q, D)}{P_{\theta^{\text{ref}}}(y^- \mid q, D)}\Big)\Bigg], \tag{8}$$

where $\sigma(\cdot)$ denotes the sigmoid function, β is a scaling factor, θ is the policy model being trained, and θ^{ref} is a fixed reference model used for relative likelihood comparison.

4 Experimental Methodology

This section details our experimental setup, covering datasets, metrics, baselines, and implementation details. More experimental details are provided in the Appendix A.1 and A.3.

Dataset. To evaluate our approach, we construct training and evaluation sets from diverse QA benchmarks. All datasets use passages retrieved from Wikipedia (Izacard et al., 2022) via the bge-large-en-v1.5 retriever (Xiao et al., 2024). The training set spans various reasoning paradigms, including open-domain QA (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017)), multi-hop QA (HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020)), and reading comprehension (SQuAD (Rajpurkar et al., 2016)). Evaluation covers ten datasets, including five in-domain test sets and five out-of-domain benchmarks—SearchQA (Dunn et al., 2017), PopQA (Mallen et al., 2022), BeerQA (Qi et al., 2021), WebQuestions (Berant et al., 2013), and Musique (Trivedi et al., 2022b), which reflect diverse knowledge and reasoning styles, from commonsense to complex multi-hop.

Evaluation Metrics. We adopt accuracy as the primary evaluation metric across all QA tasks, following previous work (Lewis et al., 2020b; Yu et al., 2024; Li et al., 2024).

Baselines. We evaluate a range of RAG methods under a unified setup, spanning from parametric LLMs to retrieval-augmented approaches incorporating reasoning, instruction tuning, and reward optimization. Vanilla LLM relies solely on internal knowledge. Vanilla RAG (Ram et al., 2023) and REPLUG (Shi et al., 2023) enhances query generation by incorporating retrieved passages through in-context learning. RA-DIT (Lin et al., 2023) applies multi-task instruction tuning to better utilize retrieved passages. RADCoT (Lee et al., 2024) augments reasoning ability by distilling chain-of-thought rationales from a teacher model. RAG-DDR (Li et al., 2024) leverages differentiable data rewards by sampling multiple candidate responses and optimizing toward those with higher reward signals. For fair comparison, all methods share the same fixed retriever to isolate improvements from generation modeling.

Implementation Details. We adopt Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024) as backbone models, fine-tuned with LoRA (Hu et al., 2022) for efficient adaptation. All models are trained for one epoch with a learning rate of 5e-5, using ten retrieved passages as external input during both training and inference. For fair comparison, retrieval-related settings follow Li et al. (2024), and all baselines are kept consistent. Following prior studies on structured reasoning (Guo et al., 2025), we constrain model outputs to a template format: reasoning steps are enclosed within `<think>...</think>` tags and final answers within `<answer>...</answer>` tags.

5 Results and Analysis

In this section, we first evaluate the overall performance of ClueAnchor and conduct ablation studies to assess the impact of each component. We then examine its ability to utilize knowledge under different evidence conditions and its robustness to noisy retrieval. Finally, we analyze the model's ability to attend to key clues, with case studies provided in Appendix A.10.

Methods	In-Domain QA					Out-of-Domain QA					Avg.
	NQ	TriQA	2Wiki	HotQA	SquAD	SeaQA	PopQA	BeerQA	WebQ	MusQ	
Llama-3.1-Instruct$_{8B}$											
Vanilla LLM	35.47	70.90	35.40	32.97	18.33	69.77	25.90	26.40	38.05	8.54	36.17
Vanilla RAG (2023)	43.40	75.63	49.23	49.80	38.70	62.00	48.93	46.63	37.70	15.08	46.71
REPLUG (2023)	38.67	71.50	46.67	45.17	28.00	62.50	41.17	39.67	40.00	19.00	43.24
RA-DIT (2023)	50.90	79.57	56.70	50.10	40.40	78.37	57.43	48.07	46.20	13.33	52.11
RADCoT (2024)	43.00	76.23	44.90	47.33	36.73	67.63	50.00	45.77	39.50	13.08	46.42
RAG-DDR (2024)	53.83	**84.37**	57.43	55.00	42.60	75.97	60.23	52.43	45.95	20.79	54.56
ClueAnchor	**54.67**	83.33	**63.70**	**61.03**	**45.83**	**82.80**	**62.60**	**56.20**	**48.90**	**24.67**	**58.37**
Qwen2.5-Instruct$_{7B}$											
Vanilla LLM	25.07	59.87	39.17	28.10	17.23	60.60	15.40	26.63	35.50	7.17	31.27
Vanilla RAG (2023)	42.30	74.70	47.50	47.13	37.97	64.53	45.87	45.00	38.20	11.46	45.47
REPLUG (2023)	36.00	69.18	41.33	37.33	31.33	58.50	43.00	39.17	32.83	13.50	40.22
RA-DIT (2023)	45.77	78.53	49.93	48.00	38.23	74.87	52.30	44.90	45.55	11.96	49.00
RADCoT (2024)	42.47	74.97	50.93	48.63	35.30	69.40	47.93	43.83	39.05	14.00	46.65
RAG-DDR (2024)	46.30	79.77	50.93	51.67	43.47	74.40	52.63	49.93	42.95	16.79	50.88
ClueAnchor	**50.60**	**81.03**	**59.97**	**56.27**	**45.00**	**76.70**	**56.63**	**52.73**	**45.90**	**19.04**	**54.39**

Table 1: Overall Performance of Different RAG Models. The highest scores are emphasized in **bold**, while the second highest scores are marked with an underline.

5.1 Overall Performance

We present the performance of RAG methods on both in-domain and out-of-domain QA tasks for the Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct models in Table 1. Overall, ClueAnchor significantly outperforms all baselines, achieving average improvements of over 3.6% across all datasets.

Compared with Vanilla RAG and REPLUG, which simply incorporate retrieved content, ClueAnchor achieves significantly better results, revealing the limitations of LLMs in selectively reasoning over noisy inputs. It also surpasses fine-tuned methods such as RA-DIT and RADCoT, which enhance external knowledge usage via instruction tuning and CoT distillation. However, RA-DIT tends to overfit to SFT labels, limiting its generalization capability, while RADCoT suffers from error propagation due to noisy CoT supervision. In contrast, ClueAnchor uses Knowledge Reasoning optimization (KRO) to dynamically align with the most effective reasoning path, enabling more adaptive and robust reasoning.

While RAG-DDR introduces reward-based alignment to select better outputs between parametric and retrieval-based responses, it supervises only the final answer and lacks further exploration of deeper clues. ClueAnchor goes further by identifying potential clues and anchoring the reasoning process around them, providing finer-grained guidance throughout generation and improving robustness and faithfulness under varying retrieval conditions, especially in complex queries where multi-step reasoning is required.

5.2 Ablation Study

As shown in Table 2, we conduct ablation studies to assess the contribution of Knowledge Reasoning Exploration (KRE) module, including Internal Knowledge Reasoning (IKR), External Knowledge Reasoning (EKR), and Clue-Anchored Knowledge Reasoning (CKR), as well as the role of anchored clues in Clue-Anchored Knowledge Reasoning.

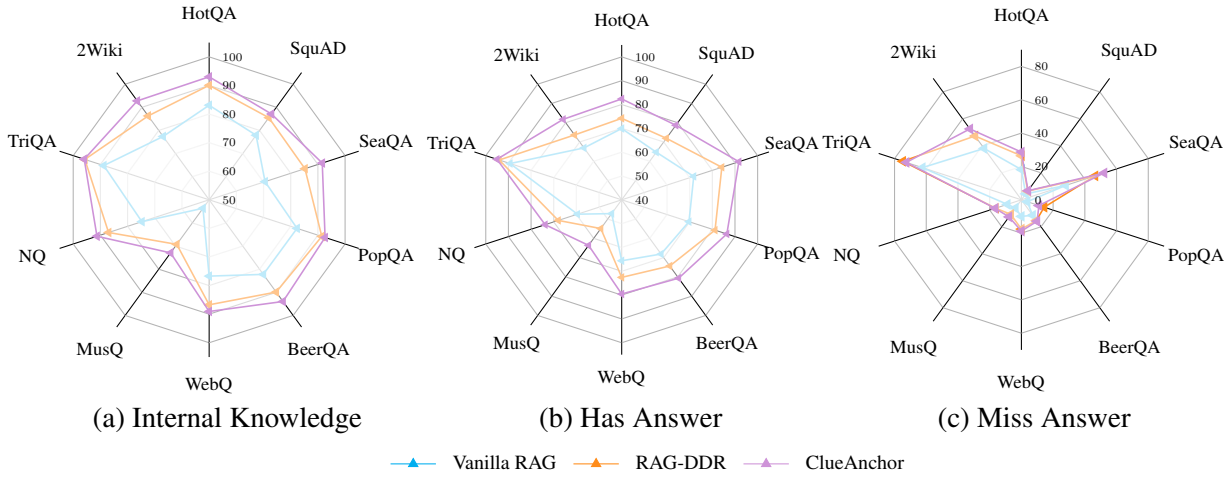
We begin by evaluating the two knowledge reasoning strategies. Removing either the Internal Knowledge Reasoning or External Knowledge Reasoning leads to a noticeable performance drop, especially in the latter case, reflecting the critical role of retrieved evidence in RAG. These results highlight the Internal Knowledge Reasoning's role in balancing parametric and retrieved knowledge, and the External Knowledge Reasoning's importance in modeling the natural reasoning path and providing intermediate signals.

Next, we remove the Clue-Anchored Knowledge Reasoning and observe consistent performance degradation compared with ClueAnchor framework, particularly on questions requiring multi-hop or implicit reasoning, such as those in HotpotQA. This suggests that Clue-Anchored Knowledge Reasoning complements the External Knowledge Reasoning by guiding the model toward overlooked or fine-grained evidence.

Finally, we replace the anchored clue with the ground-truth answer, using it directly with the query and retrieved passages as input. Despite access to the correct answer, performance declines noticeably. Without an intermediate clue to serve

Methods	In-Domain QA					Out-of-Domain QA					Avg.
	NQ	TriQA	2Wiki	HotQA	SquAD	SeaQA	PopQA	BeerQA	WebQ	MusQ	
Llama-3.1-Instruct$_{8B}$											
ClueAnchor	**54.67**	83.33	**63.70**	**61.03**	**45.83**	**82.80**	**62.60**	**56.20**	48.90	**24.67**	**58.37**
w/o IKR	47.00	79.50	56.93	53.53	38.10	74.13	50.30	48.40	42.30	17.33	50.75
w/o EKR	45.60	76.93	54.13	53.80	41.23	69.63	51.67	49.73	39.80	16.08	49.86
w/o CKR	53.50	**83.60**	63.50	60.73	45.77	81.57	61.30	54.90	47.60	24.00	57.65
w/o Anchored Clues	51.93	83.20	63.10	58.07	43.83	78.20	60.07	53.70	**49.05**	21.42	56.26
Qwen2.5-Instruct$_{7B}$											
ClueAnchor	**50.60**	**81.03**	**59.97**	**56.27**	**45.00**	**76.70**	56.63	**52.73**	45.90	**19.04**	**54.39**
w/o IKR	47.03	79.23	56.97	53.53	41.27	75.77	55.30	50.37	**46.80**	16.83	52.31
w/o EKR	47.80	79.83	58.37	53.73	42.77	71.73	52.70	50.33	42.95	16.71	51.69
w/o CKR	49.63	80.80	58.20	55.53	44.33	76.27	**56.70**	52.23	45.30	18.21	53.72
w/o Anchored Clues	46.40	77.73	56.30	52.20	41.50	70.13	51.60	49.33	42.95	15.92	50.41

Table 2: Ablation Study. We evaluate the contribution of Internal Knowledge Reasoning (**IKR**), External Knowledge Reasoning (**EKR**), and Clue-Anchored Knowledge Reasoning (**CKR**), as well as the impact of using anchored clues during Clue-Anchored Knowledge Reasoning.



(a) Internal Knowledge (b) Has Answer (c) Miss Answer

Figure 3: Effectiveness of Knowledge Reasoning Optimization in ClueAnchor. Results are shown on Llama-3.1-8B-Instruct.

as an anchor, the model struggles to localize and reason over relevant content, underscoring the importance of explicit clue extraction during Clue-Anchored Knowledge Reasoning.

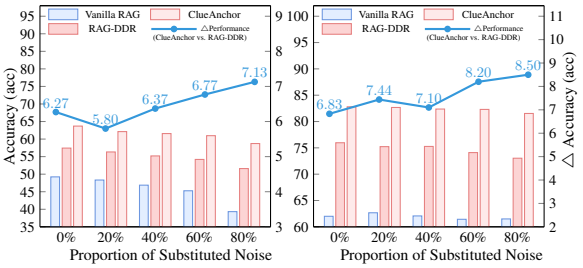
5.3 Effectiveness of Knowledge Reasoning Optimization in ClueAnchor

To assess the effectiveness of Knowledge Reasoning Optimization in ClueAnchor, we evaluate it under three distinct conditions: questions that can be answered using internal knowledge alone, questions where the retrieved passages contain the ground-truth answer, and questions where no retrieved passage includes the correct answer. We use these scenarios to assess whether the model can appropriately select between internal and external knowledge, accurately leverage retrieved evidence when available, and remain robust to irrelevant content when retrieval fails.

As shown in Figure 3, ClueAnchor consistently outperforms all baselines across most datasets under all three conditions. This demonstrates that, after knowledge reasoning optimization, ClueAnchor can adaptively rely on internal knowledge when retrieval is unhelpful, ground its reasoning in retrieved content when relevant, and leverage clue-anchored reasoning to focus on critical information within noisy passages. These results highlight ClueAnchor's ability to dynamically adjust reasoning strategies and remain robust across varying knowledge conditions. Additional experimental results are provided in the appendix A.8.

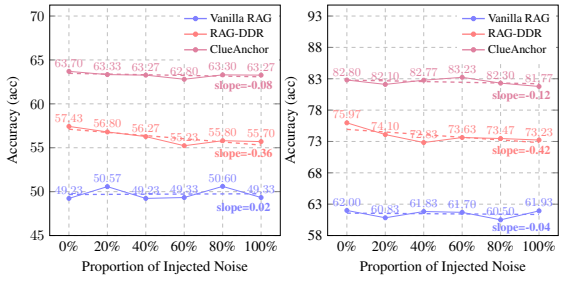
5.4 Effectiveness of ClueAnchor under Noisy Retrieval Conditions

In this experiment, we evaluate the robustness and external knowledge exploration ability of ClueAnchor under two types of noisy retrieval scenarios on the 2Wiki and SeaQA datasets using Llama-3.1-8B-Instruct. More results are provided in Ap-



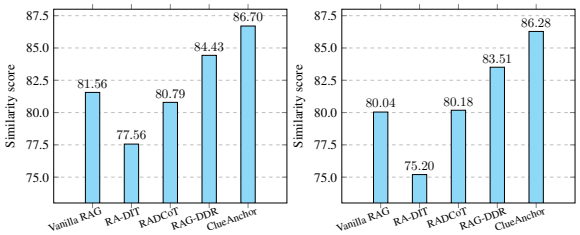
(a) 2Wiki (b) SeaQA

Figure 4: Performance of Different RAG Methods under Noisy Substitution Scenario.



(a) 2Wiki (b) SeaQA

Figure 5: Performance of Different RAG Methods under Noisy Injection Scenario.



(a) 2Wiki (b) SeaQA

Figure 6: Performance of Different RAG Methods Based on Clue-Hit Rates.

pendix A.9.

Noise Substitution. As shown in Figure 4, we degrade retrieval quality by gradually replacing the original relevant passages with irrelevant ones. While all methods suffer performance drops as noise increases, ClueAnchor consistently outperforms RAG-DDR, with the performance gap expanding steadily as noise levels rise. This widening margin indicates that ClueAnchor is better at resisting noise and maintaining useful signal extraction under increasingly noisy conditions.

Noise Injection. We incrementally add irrelevant passages while preserving the original retrieved content, and present the results in Figure 5. We quantify robustness by measuring the slope of accuracy decline, where a smaller slope indicates stronger resistance to noise. In contrast to RAG-DDR, ClueAnchor maintains an almost flat slope, demonstrating its superior ability to resist noisy inputs and anchor on relevant evidence. Interestingly, Vanilla RAG remains stable under noise, possibly because it does not effectively leverage retrieved evidence in the first place.

5.5 Evaluating the Contribution of Clue-Anchored Reasoning in ClueAnchor

While previous experiments have demonstrated that ClueAnchor performs well across various tasks and retrieval settings, they do not directly verify its ability to trace key clues during reasoning. To verify this, we conduct a clue-hit analysis on the out-domain test sets of 2Wiki and SeaQA using Llama-3.1-8B-Instruct. Complete results are provided in Appendix A.7. We further support this analysis with case studies in Appendix A.10.

For each sample, we first generate a ground-truth clue based on the ground-truth answer and retrieved documents. We then segment the model's generated reasoning into individual sentences and compute the semantic similarity between each sentence and the ground-truth clue using the bge-large-en-v1.5 retriever. The highest similarity score is recorded as the final clue-hit score, reflecting how well the model's reasoning aligns with informative evidence.

As shown in Figure 6, ClueAnchor achieves the highest clue-hit scores on both datasets, significantly outperforming all baselines. In contrast, RA-DIT achieves lower similarity scores because it is trained to directly predict answers, without learning how to extract and reason over useful clues from retrieved content. These findings confirm that ClueAnchor effectively aligns its reasoning with key clue evidence, validating the core intuition behind our method.

6 Conclusion

This paper presents ClueAnchor, a novel framework that enhances retrieval-agmented generation by anchoring reasoning on key evidence clues extracted from retrieved documents. It combines knowledge reasoning exploration with knowledge reasoning optimization to improve the model's ability to identify and leverage critical information. Experimental results demonstrate that ClueAnchor maintains robust performance under increasingly noisy retrieval conditions. Further analysis shows that it effectively learns to trace and utilize rele-

vant clues during inference, even without access to ground-truth supervision.

Limitations

Despite ClueAnchor's effectiveness in guiding reasoning through key clues, its success still depends on the model's ability to comprehend and internalize complex semantic relationships between the question, retrieved content, and the ground-truth answer. When the reasoning chain involves subtle or implicit connections, even providing the ground-truth answer may not ensure accurate clue extraction. This reveals a fundamental challenge: large language models may still lack the fine-grained discriminative capacity to localize the correct evidential span, particularly when the supporting content is obliquely phrased, dispersed across multiple documents, or overshadowed by semantically similar but irrelevant information. Consequently, the model may not fully anchor its reasoning on the appropriate clues, weakening the connection between retrieved evidence and the final answer.

Ethics Statement

This work does not involve any ethical concerns. All datasets used in our study are publicly available and sourced from open-access repositories. No personal, sensitive, or private data is involved. All models and data are used strictly in accordance with their intended research purposes and license agreements.

Acknowledgements

We gratefully acknowledge the support of the National Natural Science Foundation of China (NSFC) via grant 62236004, 62206042, 62206078 and 62476073. We also acknowledge the support of the AI9Stars community.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.

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

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 194–199.

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.

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

Yuxuan Chen, Dewen Guo, Sen Mei, Xinze Li, Hao Chen, Yishan Li, Yixuan Wang, Chaoyue Tang, Ruobing Wang, Dingjun Wu, and 1 others. 2025b. Ultrarag: A modular and automated toolkit for adaptive retrieval-augmented generation. *arXiv preprint arXiv:2504.08761*.

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

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Mohsen Fayyaz, Ali Modarressi, Hinrich Schuetze, and Nanyun Peng. 2025. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. *arXiv preprint arXiv:2503.05037*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. Grag: Graph retrieval-augmented generation. *arXiv preprint arXiv:2405.16506*.

Pengcheng Huang, Shuhao Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Zulong Chen, and Tong Xiao. 2025a. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models. *arXiv preprint arXiv:2508.13021*.

Pengcheng Huang, Zhenghao Liu, Yukun Yan, Haiyan Zhao, Xiaoyuan Yi, Hao Chen, Zhiyuan Liu, Maosong Sun, Tong Xiao, Ge Yu, and Chenyan Xiong. 2025b. Parammute: Suppressing knowledge-critical ffns for faithful retrieval-augmented generation. *Preprint*, arXiv:2502.15543.

Jeongyeon Hwang, Junyoung Park, Hyejin Park, Sangdon Park, and Jungseul Ok. 2024. Retrieval-augmented generation with estimation of source reliability. *arXiv preprint arXiv:2410.22954*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

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*.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. 2024. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In *Proceedings of COLING*, pages 16867–16878.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Sung-Min Lee, Eunhwan Park, Donghyeon Jeon, Inho Kang, and Seung-Hoon Na. 2024. Radcot: Retrieval-augmented distillation to specialization models for generating chain-of-thoughts in query expansion. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13514–13523.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Feiyang Li, Peng Fang, Zhan Shi, Arijit Khan, Fang Wang, Dan Feng, Weihao Wang, Xin Zhang, and

Yongjian Cui. 2025a. Cot-rag: Integrating chain of thought and retrieval-augmented generation to enhance reasoning in large language models. *arXiv preprint arXiv:2504.13534*.

Shaobo Li, Xiaoguang Li, Lifeng Shang, Xin Jiang, Qun Liu, Chengjie Sun, Zhenzhou Ji, and Bingquan Liu. 2021. Hopretriever: Retrieve hops over wikipedia to answer complex questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13279–13287.

Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, Zhiyuan Liu, and 1 others. 2024. Rag-ddr: Optimizing retrieval-augmented generation using differentiable data rewards. *arXiv preprint arXiv:2410.13509*.

Zhonghao Li, Kunpeng Zhang, Jinghuai Ou, Shuliang Liu, and Xuming Hu. 2025b. Treehop: Generate and filter next query embeddings efficiently for multi-hop question answering. *arXiv preprint arXiv:2504.20114*.

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, and 1 others. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Zhenghao Liu, Pengcheng Huang, Zhipeng Xu, Xinze Li, Shuliang Liu, Chunyi Peng, Haidong Xin, Yukun Yan, Shuo Wang, Xu Han, and 1 others. Knowledge intensive agents. *Available at SSRN 5459034*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Transactions on Information Systems*, 43(2):1–32.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

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*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and 1 others. 2022. Teaching language models to support answers with verified quotes, 2022. *URL https://arxiv.org/abs/2203.11147*.

Marius Mosbach, Vagrant Gautam, Tomás Vergara-Browne, Dietrich Klakow, and Mor Geva. 2024. From insights to actions: The impact of interpretability and analysis research on nlp. *arXiv preprint arXiv:2406.12618*.

Peng Qi, Haejun Lee, Oghenetegiri "TG" Sido, and Christopher D. Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Empirical Methods for Natural Language Processing (EMNLP)*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pages 2366–2377.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024. Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 12–22.

W Su, Y Tang, Q Ai, Z Wu, and Y Liu. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. arxiv 2024. *arXiv preprint arXiv:2403.10081*.

Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Yang Song, Jun Xu, Xiao Zhang, Weijie Yu, and Han Li. 2024. Redeep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. *arXiv preprint arXiv:2410.11414*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

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

Ruobing Wang, Qingfei Zhao, Yukun Yan, Daren Zha, Yuxuan Chen, Shi Yu, Zhenghao Liu, Yixuan Wang, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025a. Deepnote: Note-centric deep retrieval-augmented generation. *Preprint*, arXiv:2410.08821.

Yubo Wang, Haoyang Li, Fei Teng, and Lei Chen. 2025b. Graph-based retrieval augmented generation for dynamic few-shot text classification. *arXiv preprint arXiv:2501.02844*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *Proceedings of ICML*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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*.

Fuda Ye, Shuangyin Li, Yongqi Zhang, and Lei Chen. 2024. Rˆ 2ag: Incorporating retrieval information into retrieval augmented generation. *arXiv preprint arXiv:2406.13249*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pages 102–120. Springer.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Zeping Yu and Sophia Ananiadou. 2024. Interpreting arithmetic mechanism in large language models through comparative neuron analysis. *arXiv preprint arXiv:2409.14144*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Appendix

A.1 Prompt Templates Used in ClueAnchor

In this subsection, we introduce the prompt templates used by ClueAnchor's Knowledge Reasoning Exploration module to elicit distinct reasoning behaviors.

Internal Knowledge Reasoning. The model answers the question based solely on its internal knowledge.

Internal Knowledge Reasoning

Please think about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.
You could perform thinking with decomposing, understanding, recalling, reflecting, brainstorming, verifying, refining, and revising.
Question: {question}
Answer:

External Knowledge Reasoning. The model answers the question by reasoning over the retrieved passages.

External Knowledge Reasoning

Please think about the reasoning process in the mind and then provides the user with the answer based on the given background.
The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.
You could perform thinking with decomposing, understanding, recalling, reflecting, brainstorming, verifying, refining, and revising.
You first need to determine whether the background contains information related to the problem. If not, please answer the question based on general knowledge.
Background: {background}
Question: {question}
Answer:

Clue Extraction. The model extracts explicit sentence(s) from the passage that directly support the given answer.

Clue Extraction

You are given a background passage, a question, and its correct answer. Your task is to extract the key clue sentence(s) from the passage that directly support the answer.
Instructions:
1.Only extract content that appears explicitly in the passage.
2.Do not include any reasoning, explanation, or inferred information.
3.Output must be faithful to the original wording in the passage, with no paraphrasing or modification.
Background: {background}
Question: {question}
Answer: {answer}
Extracted supporting content:

Clue-Anchored Knowledge Reasoning. The model answers the question by identifying and utilizing key clues from the retrieved passages.

Clue-Anchored Knowledge Reasoning

Please think about the reasoning process in the mind and then provides the user with the answer based on the given background.
The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.
You could perform thinking with decomposing, understanding, recalling, reflecting, brainstorming, verifying, refining, and revising.
You first need to determine whether the background contains information related to the problem. If not, please answer the question based on general knowledge.
To assist your reasoning, some potentially key clue information from the Background may have been highlighted or emphasized in the input, Please use these as guidance when they are available, but still ensure you consider the entire Background as needed. Or it may indicate that the Background does not contain the answer, in which case you should rely on general knowledge and reasoning.
Background: {background}
Key clue information: {clue}
Question: {question}
Answer:

Task	LLaMA-3.1-8B Data Size (# / %)	Qwen2.5-7B Data Size (# / %)
NQ	9385 (27.9%)	11198 (31.6%)
TriQA	6062 (18.0%)	6055 (17.1%)
2Wiki	5804 (17.3%)	4193 (11.8%)
HotQA	6587 (19.6%)	7223 (20.4%)
SquAD	5794 (17.2%)	6728 (19.0%)
Total	33632	35397

Table 3: Data Statistics of Training Data. Percentages represent the proportion of each dataset within the total samples for each model.

In-Domain Tasks		Out-of-Domain Tasks	
Task	Data Size	Task	Data Size
NQ	3000	SeaQA	3000
TriQA	3000	PopQA	3000
2Wiki	3000	BeerQA	3000
HotQA	3000	WebQA	2032
SquAD	3000	MuSiQue	2417

Table 4: Data Statistics of Evaluation Data.

A.2 License

We present the licenses of the datasets used in this paper: Natural Questions (CC BY-SA 3.0 license), PopQA and NewsQA (MIT License), 2WikiMultihopQA, SearchQA and TriviaQA(Apache License 2.0), HotpotQA, SQuAD, Web Question and MusiQue (CCBY-SA 4.0 license).

All these licenses and agreements permit the use of their data for academic purposes.

A.3 Additional Experimental Details

In this subsection, we provide details of the data processing procedures used for training and evaluation in the ClueAnchor framework.

Training Data. We begin by randomly sampling 20,000 instances from each individual task dataset. These candidates are then filtered through a preference-based sampling procedure using the backbone model, retaining only those suitable for preference supervision. As a result, the final training set sizes vary across tasks, as shown in Table 3. Finally, all task-specific instances are mixed to ensure task diversity during training.

Evaluation Data. For evaluation, we randomly sample 3,000 instances from each benchmark dataset to ensure consistency and computational efficiency. For smaller datasets (e.g., WebQA and MuSiQue), we use the entire set. This fixed-size strategy ensures fair comparisons across models and tasks while keeping evaluation costs manageable. Detailed statistics are shown in Table 4.

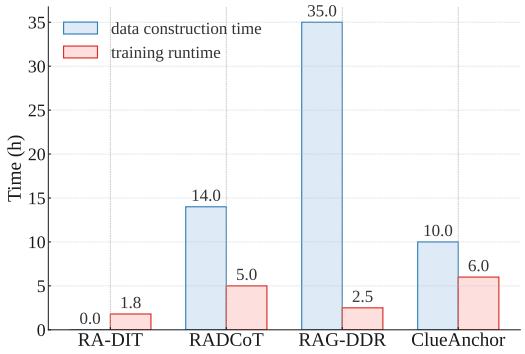


Figure 7: Training-side Computation Cost across Different RAG Methods.

A.4 Computational Cost Analysis

To further assess the computational cost of ClueAnchor, we compare its training-side overhead and inference-time efficiency against representative RAG baselines.

Training-side Overhead. We evaluate the training-side overhead by measuring the *data construction time* and *training runtime* of different methods, as shown in Figure 7. Compared with RAG-DDR, which also samples multiple responses per instance and applies reward-based optimization, ClueAnchor achieves high computational efficiency overall. Specifically, its data construction time is less than one-third of RAG-DDR, while the training runtime is only slightly higher, which is mainly due to the incorporation of CoT supervision during fine-tuning. In contrast to large-scale CoT distillation approaches such as RAD-CoT, ClueAnchor achieves substantial performance gains with only about 85% of the computational cost, highlighting its cost-effectiveness and practical advantage.

Inference-time Efficiency. During the inference phase, both ClueAnchor and the compared RAG baselines were evaluated under identical experimental configurations to ensure fairness. Since ClueAnchor does not introduce any additional modules into the inference process, its inference pipeline is strictly equivalent to that of standard RAG. Consequently, ClueAnchor incurs no extra latency or memory consumption, and its improvements are achieved without sacrificing deployment-time efficiency.

Methods	T-REx	WoW	GSM8K	Summary	Continuation
Llama-3.1-Instruct$_{8B}$					
Vanilla RAG	30.73	10.57	81.25	39.60	13.14
RAG-DDR	36.20	11.79	81.50	40.17	16.73
ClueAnchor	**41.27**	**12.11**	**85.05**	**41.98**	**18.07**
Qwen2.5-Instruct$_{7B}$					
Vanilla RAG	31.37	10.84	90.95	37.26	14.43
RAG-DDR	33.23	11.40	90.55	40.73	15.18
ClueAnchor	**37.73**	**11.86**	**91.25**	**41.75**	**15.68**

Table 5: Experimental Evaluation on varied benchmark settings.

Methods	NQ	TriQA	2Wiki	HotQA	SquAD
Qwen2.5-Instruct$_{3B}$					
Search-R1	36.83	66.47	32.93	38.03	33.03
ClueAnchor	**48.20**	**78.00**	**52.33**	**51.10**	**42.80**
Qwen2.5-Instruct$_{7B}$					
Search-R1	41.40	67.87	41.37	37.53	36.10
ClueAnchor	**50.60**	**81.03**	**59.97**	**56.27**	**45.00**
Methods	SeaQA	PopQA	BeerQA	WebQ	MusQ
Qwen2.5-Instruct$_{3B}$					
Search-R1	44.47	41.90	39.07	34.05	8.92
ClueAnchor	**69.03**	**53.90**	**49.90**	**45.00**	**14.29**
Qwen2.5-Instruct$_{7B}$					
Search-R1	49.00	44.73	42.93	35.45	15.67
ClueAnchor	**76.70**	**56.63**	**52.73**	**45.90**	**19.04**

Table 6: Performance Comparison between ClueAnchor and Search-R1 on RAG Benchmarks.

A.5 Experiments on varied benchmark settings

In this subsection, we extend the evaluation beyond traditional QA settings. Specifically, we evaluate ClueAnchor on T-REx (Elsahar et al., 2018), a factual slot filling benchmark; WoW (Dinan et al., 2018), a knowledge-grounded dialogue dataset; and GSM8K (Cobbe et al., 2021), a benchmark for mathematical reasoning. In addition, we conduct experiments on the CRUD-RAG benchmark(Lyu et al., 2025), a real-world Chinese dataset focusing on news-related tasks. CRUD-RAG contains heterogeneous and loosely structured content, requiring models to perform deeper understanding and generation over long-form news articles, rather than direct factual answering. We evaluate two representative tasks: Summary (news summarization) and Continuation (news continuation), both of which substantially differ from QA in terms of structure and objective.

As shown in Table 5, ClueAnchor delivers consistently strong performance across these non-QA tasks, demonstrating its ability to generalize effectively to heterogeneous and unstructured scenarios. These results highlight the generalizability of clue-anchored reasoning, suggesting that its benefits extend well beyond question answering.

A.6 Comparison with Deep Reasoning Approaches

Recent progress in deep reasoning has largely focused on reinforcement learning–based strategies that enhance the internal reasoning capability of large language models. Representative works, such as DeepResearch and its implementation Search-R1 (Jin et al., 2025), employ reinforcement learning to optimize a policy over multi-step reasoning trajectories under outcome-level or process-level rewards. While these approaches have shown promising improvements in complex reasoning tasks, their optimization objective is irrelevant to the problem addressed by ClueAnchor. Specifically, ClueAnchor focuses on leveraging ground-truth answers during training to help the model identify implicit but essential clue signals from retrieved documents, thereby mitigating the failure cases where correct answers cannot be reached even after multiple samplings.

To provide a clearer comparison, we evaluate ClueAnchor against Search-R1 under two backbone models: Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct. As reported in Table 6, ClueAnchor consistently outperforms Search-R1 across both model scales. This result highlights the complementary nature of the two approaches: reinforcement learning–based methods primarily improve internal reasoning dynamics, while ClueAnchor strengthens the integration of external evidence through explicit clue anchoring.

A.7 Additional Experiments on Clue-Anchored Reasoning

In Section 5.5, we show that ClueAnchor's performance gains on the 2Wiki and SeaQA datasets using LLaMA-3.1-8B-Instruct largely stem from its ability to identify and follow key evidence clues. To further support this observation, we perform the same clue-hit analysis across all datasets using both LLaMA-3.1-8B-Instruct and Qwen2.5-7B-Instruct. Results consistently confirm that ClueAnchor aligns generated reasoning more closely with key clues, highlighting its effectiveness in improving reasoning faithfulness and robustness. Full results are reported in Table 7.

Methods	In-Domain QA					Out-of-Domain QA					Avg.
	NQ	TriQA	2Wiki	HotQA	SquAD	SeaQA	PopQA	BeerQA	WebQ	MusQ	
Llama-3.1-Instruct$_{8B}$											
Vanilla RAG	81.56	83.38	81.56	79.99	82.21	80.04	79.33	82.13	81.26	79.74	81.12
RA-DIT	77.56	78.52	77.56	75.88	78.16	75.20	74.31	77.46	77.74	75.54	76.79
RADCoT	80.97	83.36	80.97	80.05	81.67	80.18	78.96	81.22	80.58	79.97	80.79
RAG-DDR	84.43	86.32	84.43	83.09	85.29	83.51	83.47	85.06	83.90	83.53	84.30
ClueAnchor	**86.70**	**88.73**	**86.70**	**85.29**	**88.20**	**86.28**	**86.41**	**87.31**	**85.81**	**86.25**	**86.77**
Qwen2.5-Instruct$_{7B}$											
Vanilla RAG	82.73	84.77	80.38	81.15	83.84	81.12	79.80	83.24	82.01	80.02	81.91
RA-DIT	76.21	77.29	76.93	74.28	77.05	75.90	73.82	77.43	77.20	75.37	76.15
RADCoT	82.34	83.65	79.88	80.80	83.13	80.81	78.81	82.48	81.66	79.51	81.31
RAG-DDR	85.60	87.13	84.21	84.74	86.86	84.03	83.03	86.63	84.79	83.88	85.09
ClueAnchor	**87.90**	**89.28**	**86.78**	**87.30**	**89.30**	**86.64**	**86.31**	**88.97**	**86.75**	**86.63**	**87.59**

Table 7: Experimental Evaluation of Clue-Hit Rates Across RAG Models.

A.8 Additional Experiments on Knowledge Reasoning Optimization

In Section 5.3, we visualize the performance of different RAG methods under varying knowledge availability conditions using LLaMA-3.1-8B-Instruct (Figure 3). To provide a more comprehensive analysis, we report the complete results for both LLaMA-3.1-8B-Instruct and Qwen2.5-7B-Instruct in Table 8, covering all evaluation scenarios. Additionally, we present the corresponding visualization for Qwen2.5-7B-Instruct in Figure 8, enabling direct cross-model comparison. The results on Qwen2.5-7B-Instruct closely mirror those of LLaMA-3.1-8B-Instruct, reinforcing the robustness and generalizability of our findings.

A.9 Additional Experiments on ClueAnchor under Noisy Retrieval Conditions

In Section 5.4, we evaluate ClueAnchor's robustness on 2Wiki and SeaQA using the LLaMA-3.1-8B-Instruct model under both noise substitution and noise injection settings. To provide a more comprehensive view, we extend this analysis to all ten datasets and include results from both LLaMA-3.1-8B-Instruct and Qwen2.5-7B-Instruct. Table 9 reports the performance under noise substitution, where relevant passages are progressively replaced with irrelevant ones, while Table 10 presents results under noise injection, where irrelevant content is added without removing the original evidence.

Across both settings, ClueAnchor consistently shows more stable performance compared to RAG-DDR. As the noise level increases, its accuracy degrades more gracefully, and the performance gap between the two methods widens, especially in high-noise scenarios. These results confirm that ClueAnchor better preserves reasoning quality by anchoring on useful clues, even when retrieval is noisy or partially corrupted.

A.10 Case Studies

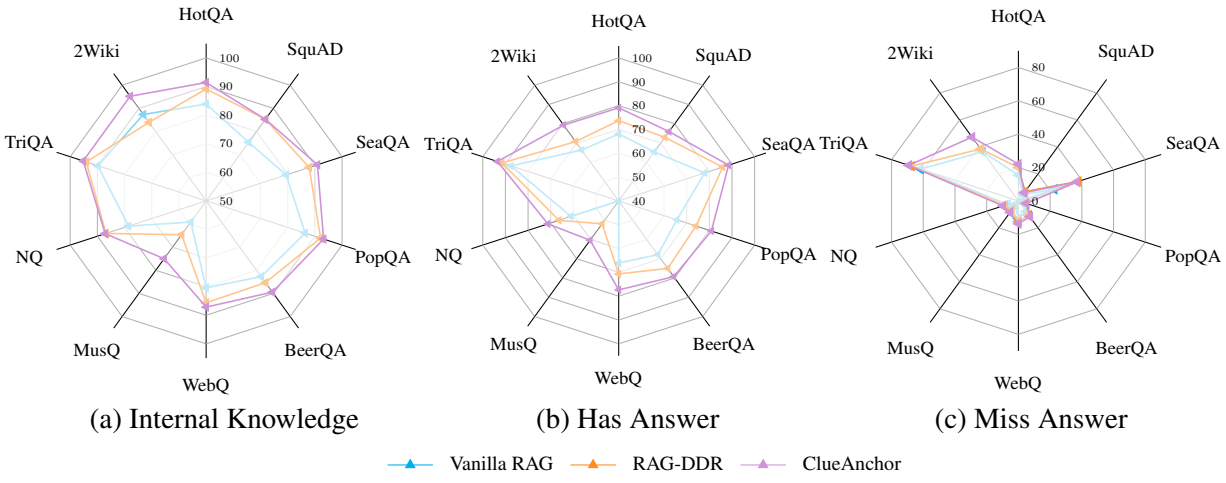
In this subsection, we present two representative cases, one from a multi-hop reasoning task and another from a fact-intensive QA task, to further demonstrate the effectiveness of the ClueAnchor framework.

Multi-Hop Case. As shown in Table 11, multi-hop tasks require models to retrieve and integrate discrete evidence from multiple documents. In this case, only ClueAnchor successfully distinguishes between several individuals named Walter Devereux and accurately resolves their familial relationships. By linking entities across passages, it reconstructs the correct lineage and completes the multi-hop reasoning process. Moreover, ClueAnchor better adheres to the expected answer format compared to other methods.

Fact-Intensive Case. A similar challenge occurs in the fact-intensive case shown in Table 12, where the model must extract precise information from a large body of distracting content. Although the correct answer is retrievable, the presence of semantically similar but incorrect details increases difficulty. ClueAnchor initially considers multiple candidates but then re-examines the context to identify the correct answer through a reflective reasoning step. This enables it to resolve ambiguity and deliver a more accurate and well-grounded response, demonstrating robustness under noisy retrieval conditions.

Methods	In-Domain QA					Out-of-Domain QA					Avg.
	NQ	TriQA	2Wiki	HotQA	SquAD	SeaQA	PopQA	BeerQA	WebQ	MusQ	
Llama-3.1-Instruct$_{8B}$											
Internal knowledge											
Vanilla RAG	74.74	88.76	77.39	83.08	77.92	70.56	82.33	82.24	76.69	53.67	76.74
RAG-DDR	86.99	95.72	86.28	90.05	85.51	85.23	91.43	89.89	86.72	69.27	86.71
ClueAnchor	**91.26**	**95.95**	**92.77**	**93.03**	**87.10**	**91.60**	**92.64**	**93.96**	**89.06**	**72.94**	**90.03**
Has answer											
Vanilla RAG	59.53	89.26	66.93	69.92	64.71	71.80	69.69	68.13	65.58	47.16	67.27
RAG-DDR	67.99	94.07	73.65	74.20	71.91	84.20	81.19	74.34	72.56	54.92	74.90
ClueAnchor	**73.79**	**95.06**	**81.76**	**82.35**	**78.69**	**91.73**	**86.40**	**80.66**	**79.54**	**63.64**	**81.36**
Miss answer											
Vanilla RAG	8.86	62.44	38.07	18.11	4.38	27.92	3.69	11.49	9.94	6.09	19.48
RAG-DDR	16.24	**74.78**	47.30	26.38	6.45	46.44	**14.35**	15.32	17.55	10.80	27.56
ClueAnchor	**16.54**	72.62	**52.54**	**28.63**	**6.75**	**52.14**	11.08	**15.93**	**18.76**	**12.65**	**28.76**
Qwen2.5-Instruct$_{7B}$											
Internal knowledge											
Vanilla RAG	78.55	89.81	87.36	83.88	75.33	79.42	86.42	82.74	80.39	59.15	80.31
RAG-DDR	86.86	94.10	83.98	89.18	85.28	87.89	92.08	85.34	85.64	64.63	85.50
ClueAnchor	**87.13**	**95.21**	**95.24**	**91.41**	**85.28**	**90.93**	**93.21**	**89.45**	**87.15**	**75.00**	**89.00**
Has answer											
Vanilla RAG	61.04	87.54	66.40	68.09	65.44	77.94	65.69	67.91	66.06	39.96	66.61
RAG-DDR	66.38	91.50	70.77	73.70	73.00	85.90	74.12	75.04	70.65	51.89	73.30
ClueAnchor	**71.22**	**93.54**	**79.32**	**79.08**	**75.97**	**88.51**	**80.90**	**79.31**	**77.34**	**60.42**	**78.56**
Miss answer											
Vanilla RAG	6.79	61.63	36.46	15.44	4.53	22.65	2.11	7.83	10.65	3.39	17.15
RAG-DDR	8.56	67.36	38.51	19.95	**6.90**	**37.75**	4.32	10.53	12.37	6.93	21.32
ClueAnchor	**9.75**	**69.12**	**47.30**	**22.04**	6.01	36.89	**4.96**	**11.75**	**13.69**	**8.52**	**23.00**

Table 8: Performance of RAG Methods under Different Knowledge Scenarios. **Internal Knowledge** refers to instances answerable without retrieved documents. **Has Answer** denotes cases where the retrieved content contains the correct answer, while **Miss Answer** represents cases where no retrieved passage provides the correct answer.



(a) Internal Knowledge (b) Has Answer (c) Miss Answer

Figure 8: Effectiveness of Knowledge Reasoning Optimization in ClueAnchor. Results are shown on Qwen2.5-7B-Instruct.

Methods	In-Domain QA					Out-of-Domain QA					Avg.
	NQ	TriQA	2Wiki	HotQA	SquAD	SeaQA	PopQA	BeerQA	WebQ	MusQ	
Llama-3.1-Instruct$_{8B}$											
0% Noise Substitution											
Vanilla RAG	43.40	75.63	49.23	49.80	38.70	62.00	48.93	46.63	37.70	15.08	46.71
RAG-DDR	53.83	84.37	57.43	55.00	42.60	75.97	60.23	52.43	45.95	20.79	54.56
ClueAnchor	54.67	83.33	63.70	61.03	45.83	82.80	62.60	56.20	48.90	24.67	58.37
ΔScore	3.84	-1.04	6.27	6.03	3.23	6.83	2.37	3.77	2.95	3.88	3.81
20% Noise Substitution											
Vanilla RAG	42.33	75.17	48.33	48.63	36.77	62.67	47.80	45.97	36.85	15.88	46.04
RAG-DDR	49.60	83.43	56.33	54.07	42.13	75.23	57.73	51.53	46.90	19.78	53.67
ClueAnchor	54.13	83.37	62.13	60.03	45.03	82.67	61.03	55.23	49.85	23.08	57.66
ΔScore	4.53	-0.06	5.80	5.96	2.90	7.44	3.30	3.70	2.95	3.30	3.98
40% Noise Substitution											
Vanilla RAG	42.43	75.77	46.87	48.83	36.73	62.07	46.30	43.73	36.80	14.83	45.44
RAG-DDR	48.63	83.43	55.20	53.53	40.93	75.27	56.57	50.00	47.05	19.11	52.97
ClueAnchor	53.90	82.47	61.57	59.53	44.70	82.37	59.30	53.37	49.85	22.58	56.96
ΔScore	5.27	-0.96	6.37	6.00	3.77	7.10	2.73	3.37	2.80	3.47	3.99
60% Noise Substitution											
Vanilla RAG	41.33	75.00	45.27	46.37	34.77	61.43	44.00	43.17	36.60	14.00	44.19
RAG-DDR	47.87	83.13	54.20	52.27	39.97	74.10	53.97	47.90	47.00	18.87	51.93
ClueAnchor	52.77	83.20	60.97	57.57	43.30	82.30	56.80	52.43	49.85	22.58	56.18
ΔScore	4.90	0.07	6.77	5.30	3.33	8.20	2.83	4.53	2.85	3.71	4.25
80% Noise Substitution											
Vanilla RAG	38.17	72.67	39.33	44.20	31.67	61.50	40.80	40.23	36.30	12.83	41.77
RAG-DDR	46.80	82.07	51.60	50.70	36.40	73.03	50.93	46.07	46.90	18.04	50.25
ClueAnchor	51.27	82.00	58.73	56.03	40.33	81.53	53.87	50.77	49.50	21.42	54.55
ΔScore	4.47	-0.07	7.13	5.33	3.93	8.50	2.94	4.70	2.60	3.38	4.29
Qwen2.5-Instruct$_{7B}$											
0% Noise Substitution											
Vanilla RAG	42.30	74.70	47.50	47.13	37.97	64.53	45.87	45.00	38.20	11.46	45.47
RAG-DDR	46.30	79.77	50.93	51.67	43.47	74.40	52.63	49.93	42.95	16.79	50.88
ClueAnchor	50.60	81.03	59.97	56.27	45.00	76.70	56.63	52.73	45.90	19.04	54.39
ΔScore	4.30	1.26	9.04	4.60	1.53	2.30	4.00	2.80	2.95	2.25	3.50
20% Noise Substitution											
Vanilla RAG	42.37	74.07	47.03	46.10	37.60	64.37	44.63	43.47	38.85	12.58	45.11
RAG-DDR	46.00	78.60	49.57	49.97	41.83	74.13	51.00	49.33	42.86	16.22	49.95
ClueAnchor	49.93	81.00	59.13	55.37	43.07	76.80	54.73	51.57	45.45	18.50	53.56
ΔScore	3.93	2.40	9.56	5.40	1.24	2.67	3.73	2.24	2.59	2.28	3.60
40% Noise Substitution											
Vanilla RAG	41.57	73.10	46.10	44.60	35.63	65.40	43.43	41.73	38.10	11.50	44.12
RAG-DDR	45.87	77.87	48.83	49.77	40.47	74.63	49.50	46.80	41.73	15.68	49.12
ClueAnchor	49.40	80.00	58.50	54.53	42.40	76.83	54.00	49.97	45.35	17.83	52.88
ΔScore	3.53	2.13	9.67	4.76	1.93	2.20	4.50	3.17	3.62	2.15	3.77
60% Noise Substitution											
Vanilla RAG	39.43	72.07	45.23	43.23	33.90	63.10	41.60	40.63	37.25	11.25	42.77
RAG-DDR	44.33	77.80	47.40	47.53	39.03	73.13	46.90	46.27	40.90	15.27	47.86
ClueAnchor	47.70	79.40	56.80	52.50	41.43	76.93	50.43	49.93	45.20	17.33	51.77
ΔScore	3.37	1.60	9.40	4.97	2.40	3.80	3.53	3.66	4.30	2.06	3.91
80% Noise Substitution											
Vanilla RAG	37.33	69.87	42.53	39.73	31.07	62.40	37.10	37.53	35.15	9.16	40.19
RAG-DDR	41.70	75.33	43.63	45.77	35.47	72.13	43.70	43.23	39.37	14.40	45.47
ClueAnchor	45.10	77.13	55.07	49.87	36.93	74.43	46.33	46.00	43.85	16.83	49.15
ΔScore	3.40	1.80	11.44	4.10	1.46	2.30	2.63	2.77	4.48	2.43	3.68

Table 9: Results of Noise Substitution Experiments. Retrieved documents are progressively corrupted by replacing content with irrelevant (noisy) passages at different ratios. To better highlight robustness differences, we report the performance gap (ΔScore) between ClueAnchor and RAG-DDR at each noise level.

Methods	In-Domain QA					Out-of-Domain QA					Avg.
	NQ	TriQA	2Wiki	HotQA	SquAD	SeaQA	PopQA	BeerQA	WebQ	MusQ	
Llama-3.1-Instruct$_{8B}$											
0% Noise Injection											
Vanilla RAG	43.40	75.63	49.23	49.80	38.70	62.00	48.93	46.63	37.70	15.08	46.71
RAG-DDR	53.83	84.37	57.43	55.00	42.60	75.97	60.23	52.43	45.95	20.79	54.56
ClueAnchor	54.67	83.33	63.70	61.03	45.83	82.80	62.60	56.20	48.90	24.67	58.37
20% Noise Injection											
Vanilla RAG	42.77	75.17	50.57	49.53	38.20	60.83	48.33	46.23	38.98	15.10	46.77
RAG-DDR	48.93	83.57	56.80	54.67	42.87	74.10	58.30	51.60	47.05	19.61	53.95
ClueAnchor	55.70	83.67	63.33	59.47	45.43	82.10	62.50	55.33	49.46	23.62	58.06
40% Noise Injection											
Vanilla RAG	42.20	76.40	49.23	50.03	37.73	61.83	49.00	46.70	37.40	14.81	46.73
RAG-DDR	48.83	83.37	56.27	55.27	42.17	72.83	58.40	51.40	47.15	18.91	53.75
ClueAnchor	54.20	84.00	63.27	60.33	45.93	82.77	61.33	55.43	49.85	23.42	58.15
60% Noise Injection											
Vanilla RAG	43.30	75.33	49.33	49.60	37.10	61.70	48.93	45.47	38.24	14.23	46.52
RAG-DDR	49.93	83.47	55.23	55.17	42.33	73.63	57.20	51.90	47.24	20.40	53.82
ClueAnchor	54.03	83.87	62.80	60.23	44.87	83.23	61.97	55.53	49.16	23.17	57.82
80% Noise Injection											
Vanilla RAG	43.57	75.67	50.60	48.60	37.80	60.50	48.93	46.57	38.83	15.14	47.12
RAG-DDR	49.10	83.60	55.80	54.77	42.43	73.47	58.27	51.53	47.83	19.82	53.92
ClueAnchor	54.13	83.77	63.30	59.97	45.43	82.30	61.77	55.43	49.95	22.55	57.87
100% Noise Injection											
Vanilla RAG	42.93	75.97	49.33	49.63	36.40	61.93	48.73	47.00	38.14	15.31	46.81
RAG-DDR	49.13	83.83	55.70	53.53	42.30	73.23	57.83	51.67	46.95	20.19	53.70
ClueAnchor	54.70	83.87	63.27	59.43	45.33	81.77	62.57	54.83	49.75	22.47	57.89
Qwen2.5-Instruct$_{7B}$											
0% Noise Injection											
Vanilla RAG	42.30	74.70	47.50	47.13	37.97	64.53	45.87	45.00	38.20	11.46	45.47
RAG-DDR	46.30	79.77	50.93	51.67	43.47	74.40	52.63	49.93	42.95	16.79	50.88
ClueAnchor	50.60	81.03	59.97	56.27	45.00	76.70	56.63	52.73	45.90	19.04	54.39
20% Noise Injection											
Vanilla RAG	42.80	74.40	48.00	46.63	36.83	64.67	45.37	44.87	38.58	12.04	45.42
RAG-DDR	46.87	78.87	49.43	51.77	42.30	74.40	50.57	49.07	42.52	15.89	50.17
ClueAnchor	50.60	81.23	59.43	55.70	44.20	77.20	55.87	51.60	46.31	18.49	54.06
40% Noise Injection											
Vanilla RAG	42.23	74.17	48.47	47.20	37.40	63.87	44.77	44.63	39.32	12.08	45.51
RAG-DDR	46.47	79.13	49.90	50.17	41.67	74.70	51.23	49.23	42.91	15.59	50.10
ClueAnchor	50.27	80.80	59.93	55.50	44.57	77.33	55.00	52.93	46.21	17.87	54.04
60% Noise Injection											
Vanilla RAG	42.37	74.23	47.57	46.80	37.43	63.97	45.37	44.47	38.63	11.71	45.36
RAG-DDR	46.03	79.40	49.17	50.20	41.33	74.50	51.27	49.33	42.37	15.85	50.05
ClueAnchor	49.40	81.03	59.20	54.90	44.00	76.70	55.87	51.93	46.01	18.12	53.82
80% Noise Injection											
Vanilla RAG	42.47	74.63	47.57	46.83	36.73	63.57	43.87	43.67	38.34	11.21	45.09
RAG-DDR	45.93	78.93	49.93	50.43	41.20	73.70	51.13	48.70	42.42	15.68	49.91
ClueAnchor	50.03	81.10	59.17	55.03	44.13	76.77	54.70	52.03	46.65	18.70	53.93
100% Noise Injection											
Vanilla RAG	41.97	74.23	47.47	46.53	37.37	62.67	44.27	44.33	39.12	12.45	45.44
RAG-DDR	45.67	79.20	49.97	49.90	41.13	73.30	51.17	49.40	42.27	15.85	49.99
ClueAnchor	50.57	80.73	59.67	55.60	44.13	76.40	54.97	51.83	46.16	19.11	54.12

Table 10: Results of Noise Injection Experiments. Retrieved document contents are preserved, while varying proportions of irrelevant (noisy) passages are appended to simulate increasing retrieval noise.

Query: Who is the paternal grandfather of Leicester Devereux, 7Th Viscount Hereford?
Answer: Walter Devereux, 5th Viscount Hereford.
Passage1: " **Leicester Devereux, 7th Viscount Hereford (1674–1683) was a British Peer. He was the eldest son of Leicester Devereux, 6th Viscount Hereford (1617–1676)** and succeeded to the title as an infant. He died young and was succeeded by his younger... "
Passage2: " **Leicester Devereux, 6th Viscount Hereford (1617 – 1 December 1676) was a British Peer. He was the second son of Walter Devereux, 5th Viscount Hereford (1578–1658)**. He married Elizabeth Withipoll, daughter and sole heiress..."
Passage3: " **He was born by 1513, the son of Walter Devereux, 1st Viscount Hereford and Mary Grey** (1491-22 February 1538). **His paternal grandparents were John Devereux, 9th Baron Ferrers of Chartley and Cecily Bourchier.** His maternal grandparents were..."
Passage4: " He was the son of Edward Devereux, 11th Viscount Hereford..."
Passage5: " Robert Charles Devereux, 17th Viscount Hereford..."

Vanilla RAG Output:
" <think> To find the paternal grandfather of Leicester Devereux... **The background states that Walter Devereux, 5th Viscount Hereford was the son of Walter Devereux, 1st Viscount Hereford.** So, the paternal grandfather of Leicester Devereux, 7th Viscount Hereford is indeed Walter Devereux, 1st Viscount Hereford. </think> <answer> **Walter Devereux, 1st Viscount Hereford.**"
RA-DIT Output:
"**John Devereux, 9th Baron Ferrers of Chartley.**"
RADCoT Output:
" <think> To answer this question... **It also states that Walter Devereux was the son of John Devereux, 9th Baron Ferrers of Chartley and Cecily Bourchier.** Therefore, John Devereux, 9th Baron Ferrers of Chartley is the paternal grandfather of Leicester Devereux, 7th Viscount Hereford. </think> <answer> **John Devereux, 9th Baron Ferrers of Chartley** </answer> "
RAG-DDR Output:
" <think> Leicester Devereux, **7Th Viscount Hereford's paternal grandfather was Walter Devereux, 1st Viscount Hereford**. </think> <answer> **Walter Devereux, 1st Viscount Hereford**"
ClueAnchor Output:
" <think> First, I need to identify the information related to Leicester Devereux, 7th Viscount Hereford. From the background, I see that there are multiple mentions of different Viscounts Hereford, but I need to find the one that matches the given title. The first mention is of Leicester Devereux, 6th Viscount Hereford (1617–1676), but it's not the correct one since the question asks for the 7th Viscount Hereford. **Then, I see a mention of Leicester Devereux, 7th Viscount Hereford (1674–1683). It states that he was the eldest son of Leicester Devereux, 6th Viscount Hereford (1617–1676)**. So, I need to find the paternal grandfather of Leicester Devereux, 7th Viscount Hereford, which would be Leicester Devereux, 6th Viscount Hereford's father. **According to the text, Leicester Devereux, 6th Viscount Hereford's father was Walter Devereux, 5th Viscount Hereford (1578–1658)**. However, I also see another mention of Walter Devereux, 1st Viscount Hereford and his son Walter Devereux, who had a son John Devereux, son of Walter Devereux who died at the Battle of Bosworth, married Cecily Bourchier. But this does not seem to be directly related to the 7th Viscount Hereford. Therefore, I will focus on the direct lineage of Leicester Devereux, 7th Viscount Hereford. </think> <answer> **Walter Devereux, 5th Viscount Hereford (1578–1658)** </answer> "

Table 11: Case Study I. This case illustrates the outputs of various methods in a **multi-hop scenario**, using Llama-3.1-8B-Instruct as the foundational model.

Query: when was the seat moved to the capitol?
Answer: November 17, 1800.
Passage1: " For several decades, beginning when the federal government moved to Washington in the **fall of 1800**... **In 1801 the House moved to temporary quarters in the south wing**, called the Öven, **which it vacated in 1804**, returning to the north wing for three years... "
Passage2: " **Construction on the North Wing began in 1793 with the laying of the cornerstone by President George Washington**..."
Passage3: " ... From December 6, 1790, to **May 14, 1800**, the same block hosted federal, state, county, and city government offices. Congress Hall, which was originally built to... "
Passage4: " the Old Supreme Court Chamber from 1810 through 1860... **In 1810, the Supreme Court first occupied the Old Supreme Court Chamber in the Capitol**..."
Passage5: " ... However, the House of Representatives moved early into their House wing in 1807. **Though the Senate wing building was incomplete, the Capitol held its first session of the United States Congress with both chambers in session on November 17, 1800. The National Legislature was moved to Washington prematurely**, at the urging of President John Adams, in hopes of securing enough Southern votes..."

Vanilla RAG Output:
" <think> To answer this question... However, based on the context, it is reasonable to conclude that **the seat was moved to the capitol in the fall of 1800, specifically after May 14, 1800, when Congress Hall in Philadelphia ceased to be the capitol building**. </think> <answer> The seat was moved to the capitol **in the fall of 1800**."
RA-DIT Output:
"**May 14, 1800**. The federal government moved to Washington, D.C. in 1800... "
RADCoT Output:
" <think> The background... I need to recall the specific event mentioned in the text that relates to the seat being moved to the Capitol. </think> <answer> **May 14, 1800** </answer> "
RAG-DDR Output:
" <think> The background information... that **the federal government moved to Washington, D.C. in May 14, 1800**... </think> <answer> **May 14, 1800 (or fall of 1800)**"
ClueAnchor Output:
" <think> First, I need to determine if the background contains information related to the problem. The background discusses the history of the United States Capitol building, including the various locations where Congress and other government institutions were held before moving to Washington D.C. It mentions the Residence Act of 1790, which empowered President George Washington to locate a permanent capital along the Potomac River. It also talks about the construction of the Capitol building and the various expansions and renovations that took place over the years.The question asks when the seat was moved to the Capitol. **Based on the background, it seems that the federal government moved to Washington, D.C. in 1800, specifically on November 17, 1800, when the first session of the United States Congress with both chambers in session was held in the Capitol building.** However, the background also mentions that the Residence Act of 1790 was passed, which led to the decision to move the capital to Washington, D.C. So, there might be some ambiguity in the question. To clarify, I will look for specific information about when the seat of government was moved to the Capitol building. **Upon re-reading the background, I found that the Residence Act of 1790 was passed on July 16, 1790, and the federal government moved to Washington, D.C. in 1800.** Therefore, I will provide an answer based on this information. </think> <answer> **November 17, 1800 (for the first session of Congress)** </answer> "

Table 12: Case Study II. This case illustrates the outputs of various methods in a **fact-intensive scenario**, based on Llama-3.1-8B-Instruct as the foundational model.