RAGCap-Bench: Benchmarking Capabilities of LLMs in Agentic Retrieval Augmented Generation Systems

Jingru Lin* **Chen Zhang*†** **Yong Liu** **Haizhou Li**
National University of Singapore, Singapore
{jingrulin}@u.nus.edu

Abstract

Retrieval-Augmented Generation (RAG) mitigates key limitations of Large Language Models (LLMs), such as factual errors, outdated knowledge, and hallucinations, by dynamically retrieving external information. Recent work extends this paradigm through agentic RAG systems, where LLMs act as agents to iteratively plan, retrieve, and reason over complex queries. However, these systems still struggle with challenging multi-hop questions, and their intermediate reasoning capabilities remain underexplored. To address this, we propose RAGCap-Bench, a capability-oriented benchmark for fine-grained evaluation of intermediate tasks in agentic RAG workflows. We analyze outputs from state-of-the-art systems to identify common tasks and the core capabilities required for their execution, then construct a taxonomy of typical LLM errors to design targeted evaluation questions. Experiments show that "slow-thinking" models with stronger RAGCap performance achieve better end-to-end results, underscoring the benchmark's validity and the importance of enhancing these intermediate capabilities.

1 Introduction

Although LLMs demonstrate impressive performance across diverse domains (Grattafiori et al., 2024; Hadi et al., 2023; Xi et al., 2024), their reliance on internal knowledge often leads to factual errors, outdated knowledge, and hallucinations, reducing their reliability on complex and dynamic queries (McKenna et al., 2023; Ji et al., 2023; Huang et al., 2025). These problems are alleviated with RAG which equips the LLMs with an external knowledge base (Lewis et al., 2020; Izacard et al., 2023; Gao et al., 2023; Jiang et al., 2023). However, traditional RAG system suffers from problems such as static information sources, shallow reasoning, and context integration, etc (Singh et al., 2025). Recent progress in LLM-based agentic RAG systems presents a more promising approach to mitigate these issues and improve task handling capabilities (Li et al., 2025c; Wu et al., 2025; Zheng et al., 2025). Agentic RAG augments the retrieval pipeline by giving the LLM greater autonomy. Instead of relying solely on static retrieval, the model actively interacts with open web environments to dynamically gather and filter information, apply structured reasoning over the retrieved context, and plan adaptively to address complex queries.

Existing evaluations of agentic RAG systems primarily rely on end-to-end multi-hop QA benchmarks (Krishna et al., 2025; Rein et al., 2023; Zhou et al., 2025; Xi et al., 2025). While these evaluations are reliable and efficient, they offer limited fine-grained feedback on how to improve system components. Addressing this gap requires examining the intermediate processes that shape overall performance. Agentic RAG systems iteratively perform planning, retrieval, and reasoning, where errors in any stage can compound—much like mistakes in the steps of a complex math problem. To isolate these critical intermediate tasks and the capabilities they demand, we introduce RAGCap-Bench, a fine-grained benchmark for component-level evaluation within agentic RAG systems.

RAGCap-Bench comprises four core tasks distilled from an analysis of recent open-source agentic RAG systems (Li et al., 2025c; Jin et al., 2025; Li et al., 2025a). These tasks capture essential model capabilities in agentic workflows—***Planning***, ***Evidence Extraction***, ***Grounded Reasoning***, and ***Noise Robustness***. Evaluation questions for each task are derived from a taxonomy of common intermediate-stage LLM errors and are presented as multiple-choice (MCQ) items. To ensure high quality, the questions are constructed through rigorous filtering of diverse agentic trajectories, followed by thorough manual

*Co-first Authors
†Corresponding Author

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 5478–5500
November 6-10, 2026 ©2026 Association for Computational Linguistics

Benchmark	Lang	Open Web.	Intermediate Skills
2WikiMultiHopQA (2020)	En	✗	✗
MuSiQue (Trivedi et al., 2022)	En	✗	✗
BrowseComp(Zhou et al., 2025)	En	✓	✗
BrowseComp-ZH(Zhou et al., 2025)	Zh	✓	✗
InfoDeepSeek(Xi et al., 2025)	En&Zh	✓	✗
RAGCap-Bench (ours)	En&Zh	✓	✓

Table 1: Comparisons between different benchmarks.

inspection and annotation.

By isolating specific skills, RAGCap-Bench enables decoupled and interpretable capability assessment while maintaining practical relevance: across diverse LLMs, benchmark performance strongly correlates with end-to-end results on complex multi-hop QA tasks (§4.4). The benchmark is also substantially more efficient than full agentic evaluations, for example, evaluating DeepSeek-R1 on RAGCap-Bench is roughly 50× faster than running the same model through the complete WebThinker workflow (Li et al., 2025c) on BrowseComp-ZH (Zhou et al., 2025), a dataset comparable in size to RAGCap-Bench. Key differences between RAGCap-Bench and existing QA benchmarks are summarized in Table 1.

2 Problem Formulation

We formalize the general pipeline of an agentic RAG system and identify the intermediate tasks that capture its core model capabilities. Figure 1 illustrates this overall workflow. Given a user query $q \in \mathcal{Q}$, the LLM first designs its initial question-answering plan $\pi_0 = \text{Plan}(q)$ about how to tackle the question. In the planning task, LLMs are required to break down the question into sub-queries, extract the question's intents, or brainstorm ideas etc., depending on the nature of the query. With π_0, the LLM will start to solve the problem, which requires iterative live searching, reasoning, and dynamic planning up to T steps. At each step t, the agent retrieves a set of top-K evidence $E_t = \{e_1, e_2, ..., e_k\}$ from the open web using a search engine. It then analyses the evidence set and identifies useful evidence E'_t from the set. Based on the evidence, the LLM starts to reason and produces a reasoning chain $\mathcal{R}_t = \{r_1, r_2, ..., r_m\}$ where m is the number of reasoning steps. With current evidence and reasoning, the LLM will then refine its plan $\pi_{t+1} = \text{Plan}(q|E'_{<t}, R_{<t})$, in which it determines whether current information is enough to answer the question and makes a strategic plan for the next step. If the agent thinks the current evidence and reasoning are enough, it will stop searching and proceed to generate the answer $\hat{y}$. Depending on the specific system design, some systems incrementally gather relevant evidence through iterative search and analysis, and then perform a final reasoning step over the aggregated evidence set (as indicated by the blue dotted box in Figure 1) before generating the final answer $\hat{y}$.

The complexity of the agentic RAG system makes it hard to locate the actual problem. For example, extracting a wrong evidence could lead to incorrect reasoning, which propagates to later execution stages. In RAGCap-Bench, our goal is to introduce capability-oriented evaluations that systematically assess agentic systems at the process level. For example, how well the foundation LLMs strategizes its question-answering plan π_t, extracts useful relevant evidence E'_t from E_t, and reasons R_t faithfully with E'_t. Furthermore, due to the dynamic nature of the open web environments, evaluating how well LLMs handle potentially low-quality, misleading, and less credible sources of information is also important. In the next section, we will introduce the construction of each capability task in detail.

3 RAGCap-Bench

Our benchmark comprises 255 curated questions. To construct the dataset, we draw queries from multiple open-source deep research benchmarks (Zhou et al., 2025; Xi et al., 2025; Krishna et al., 2025; Chen et al., 2025) spanning diverse domains, including entertainment, sports, arts, technology, medicine etc. Since these datasets only provide end-to-end query-answer pairs, we utilize various agentic RAG systems to process the queries and collect execution trajectory logs. From these logs, we extract the relevant intermediate information, which is then used to generate task-specific MCQs. More details about the open-source datasets and agentic systems can be found in Appendix A.

3.1 MCQ Format

Motivated by prior works (Ross et al., 2025; Hendrycks et al., 2020; Zellers et al., 2019), we frame RAGCap-Bench questions as MCQs. Each MCQ consists of a query, some intermediate outputs from the agentic RAG systems, and an instruction to ask the LLMs to choose the correct/incorrect options from all options given. The general formats of the MCQs are shown in the bottom of Figure 1.

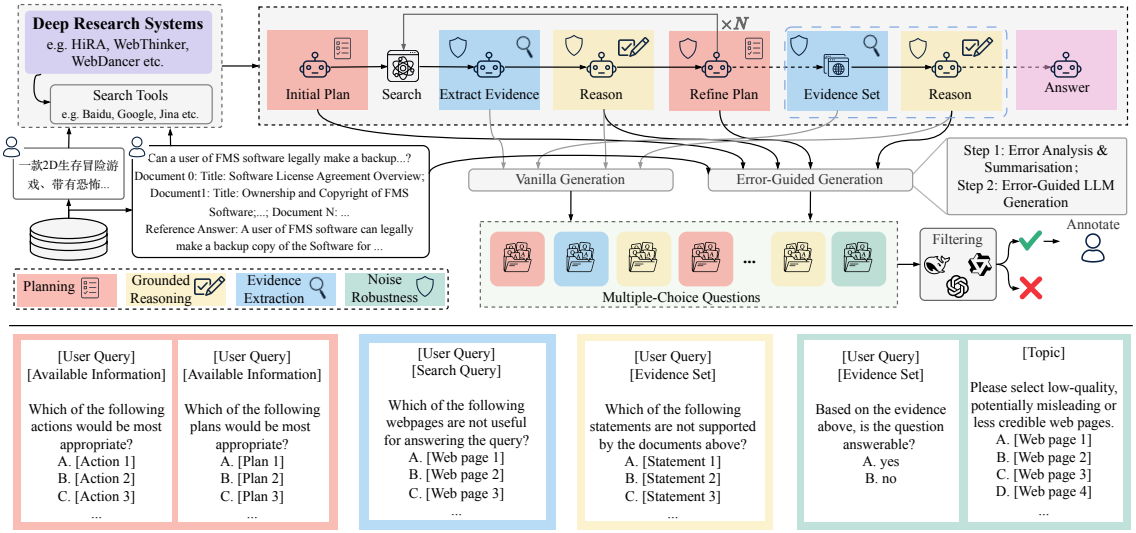


Figure 1: General agentic RAG pipeline and data construction processes for RAGCap-Bench. We take queries from open-source question-answering (QA) datasets. We then run different deep research systems and collect the intermediate outputs. Two main strategies, Vanilla Generation and Error-Guided Generation are deployed to generate the MCQs based on queries and the intermediate outputs. The generated MCQs are filtered to ensure the quality and experts are recruited to provide the answers.

Examples of MCQs are shown in Appendix E.

3.2 Dataset Construction

There are four MCQ task types: ***Planning***, ***Evidence Extraction***, ***Grounded Reasoning*** and ***Noise Robustness***. The two main MCQs generation strategies include 1) Vanilla Generation; 2) Error-Guided Generation. The former involves directly extracting and reformatting relevant information from execution logs, framing it as MCQs to evaluate the corresponding model capabilities. The latter involves dedicated error-guided prompts to instruct the LLMs to generate high-quality and challenging MCQs. We mainly use GPT-4o (Hurst et al., 2024), Qwen-Plus (Yang et al., 2025) and DeepSeek-V3 (Liu et al., 2024) for our dataset construction. Below shows how the MCQs of each type are constructed:

Planning: As shown in Figure 1, planning capability is crucial at two stages of an agentic RAG workflow. First, upon receiving the initial query, the LLM interprets the user's intent and formulates a strategic plan to guide retrieval and reasoning. Second, after retrieving evidence and performing initial reasoning, the model reassesses its progress, identifies remaining gaps, and updates its plan based on newly acquired information. Across these stages, the model must employ both convergent and divergent planning depending on the question and the information gathered so far. Convergent planning refers to the ability to narrow the search space and advance the reasoning process efficiently. In contrast, divergent planning involves broadening the search space to increase the diversity and coverage of retrieved information, ensuring that important aspects are not overlooked. In Appendix B, we show the typical use cases of convergent and divergent planning for different types of queries.

To construct MCQs that assess both convergent and divergent planning abilities, we collect relevant thinking trajectories (CoT from slow-thinking agents) corresponding to the two planning stages described above from multiple agentic RAG systems. To ensure data quality, we retain only trajectories that lead to correct final answers. Additionally, since humans excel at both convergent and divergent thinking when solving complex problems, we also collect samples from open-source datasets where human-annotated stepwise problem-solving strategies are available, Chen et al. (2025) for converging planning while Du et al. (2025) for diverging planning. Leveraging these trajectories, we prompt LLMs to generate MCQs, consisting of questions, correct and incorrect options that reflect the underlying planning processes.

Appendix D provides the detailed prompts used by LLMs to generate the MCQs. The above-mentioned high-quality thinking trajectories from

Question Type	Targeted Capability	Common Mistakes
Planning	The capability to interpret and break down the problem, implement an optimal/strategic problem-answering plan, and dynamically refine its plan based on current information.	Poor question interpretation
		Omission of necessary constraints
		Poor planning logic
		Poor dynamic planning
		Limited scope and depth
Evidence Extraction	The capability to identify useful evidence from a large amount of retrieved documents.	Shallow keyword matching
		Fail to recognise implicit connections
Grounded Reasoning	The capability to reason with grounding and generate a well-supported statement.	Hallucinated support
		Contradiction with retrieved information
		Failed to identify implicit reasoning gap
Noise Robustness	The capability to detect low-quality, less-reliable information and the capability to abstain.	Forced/Incorrect reasoning based on noisy information
		Unable to detect low-quality, misleading, and less credible sources

Table 2: Question types, the targeted capabilities, and common mistakes made by LLMs.

both agentic RAG systems and humans make up the correct options in the MCQs. For wrong options, we instruct the LLMs to generate them. Specifically, we employ Error-Guided Generation, which includes the identified common mistakes made by agentic RAG systems during the planning process. The common mistakes made are listed in Table 2.

Evidence Extraction: This task evaluates whether the model can identify evidence relevant to the user query. Vanilla Generation is adopted for MCQ construction. Specifically, we collect the search queries and retrieved web pages in the execution trajectories of different agentic RAG systems and each web page is formatted into an MCQ option, as shown in Figure 1. The MCQs typically include more than four options, and the model is expected to rule out all the irrelevant evidence to answer the question correctly. Appendix D presents the prompts for the MCQs generation.

Grounded Reasoning: This task evaluates whether a model can produce logical and coherent reasoning grounded in the available evidence. We employ two MCQ construction strategies: Error-Guided Generation and Vanilla Generation. For Error-Guided Generation, we use intermediate reasoning traces that lead to the correct final answer. The generation prompt includes the query, the evidence set, the corresponding reasoning steps, and the taxonomy of grounded-reasoning errors defined in Table 2. LLMs are instructed to generate MCQs options by selecting correct reasoning statements from reasoning traces as correct options and injecting grounded-reasoning errors (see Table 2) to create distractors. To more realistically reflect realistic agentic RAG errors, we also leverage incorrect reasoning traces. Since these traces already contain incorrect statements, we apply Vanilla Generation, prompting the LLMs to extract flawed statements without injecting additional errors. Prompts for both strategies are provided in Appendix D. Each resulting MCQ includes an instruction asking the model to identify the incorrect statements.

Noise Robustness: Real-time search results may include noise, such as low-quality, less credible, or entirely irrelevant information. During inspection of the execution logs of different agentic RAG systems, we find that they are prone to forcing their reasoning based on the retrieved evidence, even though they may contain lots of noise. In this task, we mainly assess whether the models are robust against noisy search results. For MCQs of this task, the model is provided with search results that contain no or very limited useful information, and is prompted with two options: answer the question or admit that the question cannot be answered with the given search results. In this type of question, we deem an LLM that admits the questions are not answerable as a noise-robust one. We term the capability as noise-abstain.

We also evaluate whether LLMs can identify less credible sources. We define credibility using source-level signals rather than answer content alone: official or primary sources, reputable institutional or publisher identity, clear authorship, domain expertise, recency when relevant, and absence of obvious content-farm characteristics. Although

useful information can sometimes be extracted from such sources, their reliability is difficult to guarantee, an especially critical concern in high-stakes domains such as medicine, law, academia, and public policy. The ability to recognize and prioritize credible sources is also valuable when multiple references present conflicting information. We use Vanilla MCQ Generation to construct questions by formatting retrieved web pages into different MCQ options, similar to Evidence Extraction.

3.3 Dataset Post Processing

Despite using Error-Guided Generation, LLMs may still produce trivial MCQs or exhibit issues such as hallucination and poor instruction following. To ensure question quality, we apply both difficulty filtering and format filtering. Difficulty filtering removes trivial items by prompting multiple models to answer each MCQ and discarding those for which all models choose the same option. Format filtering eliminates poorly formatted or ambiguous questions through rule-based checks and manual verification. In total, we collected 5-6k agentic trajectories from multiple systems and datasets, from which MCQs were generated. The difficulty filtering removes approximately 70-85% of the generated items for each dataset, as many of the generated MCQs are too easy. Format filtering removes an additional 10-15% for each dataset. This leaves a small, high-quality subset for the annotators. All retained MCQs are annotated by human experts with access to advanced research tools. Annotators are instructed to answer each MCQ by selecting the options they considered correct based on the provided question, intermediate reasoning context, and candidate options. Since both the questions and candidate answers were presented in multiple-choice format, annotators did not generate free-form responses, but instead verified the correctness and ambiguity of the provided options. Three NLP specialists conducted the annotations, achieving an inter-annotator agreement of 0.7825 (average pairwise Jaccard similarity). Ground-truth answers were determined by majority vote; when no majority emerged, disagreements were resolved through discussion.

3.4 Evaluation Metrics

We adopt both Exact Match (EM) and instance-wise macro F1 score as the evaluation metrics. ***Planning*** questions contain two sub-categories: converging and diverging. ***Noise Robustness*** also contains two sub-categories: abstain and reliability. The EM and F1 scores are first averaged within the group. The overall score is the average of four scores from each question type.

4 Evaluations

4.1 Setups

We evaluate a range of fast-thinking LLMs and slow-thinking LLMs. Fast-thinking refer to standard instruction-following language models that generate responses directly in a single-pass decoding process without explicit long-form deliberative reasoning. Slow-thinking models refer to language models that employ explicit multi-step deliberative reasoning before producing a final answer. Such models often generate intermediate reasoning traces (e.g., chain-of-thought or reflective reasoning). Fast-thking modles include Qwen2.5-72B-Instruct (Team, 2024), Qwen-Plus w/o think (Yang et al., 2025), DeepSeek-V3 (Liu et al., 2024), GPT-4o (Hurst et al., 2024) and Gemini-1.5-Pro (Team et al., 2024), and slow-thinking LLMs include Qwen3-8B (Yang et al., 2025), Qwen3-32B w/ think (Yang et al., 2025), QwQ-32B (Yang et al., 2025), Qwen3-235B-A22B w/ think (Yang et al., 2025), DeepSeek-R1 (Guo et al., 2025), O1-mini (Jaech et al., 2024) and Gemini-2.5-Flash (Comanici et al., 2025).

4.2 Benchmark Results

We evaluate the performance of fast-thinking and slow-thinking models across the different task types using both bare and informative prompts. All prompts are shown in Appendix H. Bare prompts refer to those with no error examples included. The LLMs are given the query, MCQs, and simple instructions about the output format. Informative prompts additionally include the few-shot examples about the common mistakes listed in Table 2. The latter shows improved and more robust performance. Table 3 shows the results for using informative prompts. The results are averaged over three runs. The comparison with bare prompts is shown in Section 4.3.

Planning: We report convergent and divergent planning scores separately in Table 3. EM_c and $F1_c$ correspond to convergent planning, while divergent planning is evaluated using EM_d only. A high EM score indicates that the model consistently selects the optimal solution path over non-optimal alternatives. In agentic RAG systems, strong EM

Model	Planning			Evidence Extraction		Grounded Reasoning		Noise Robustness			Overall	
	EM_c	$F1_c$	EM_d	EM	F1	EM	F1	EM_a	EM_r	$F1_r$	EM	F1
fast-thinking models												
Qwen2.5-72B-Instruct	27.45	54.88	76.00	31.88	79.47	35.85	77.98	64.86	10.00	77.18	39.22	72.37
Qwen-Plus w/o think	49.02	74.10	84.00	27.54	74.44	35.85	80.82	64.86	25.00	74.45	43.71	75.95
Deepseek-v3	49.02	74.97	80.00	15.94	75.07	41.51	81.19	78.38	25.00	81.01	43.41	78.06
GPT-4o	35.29	59.48	72.00	34.78	78.27	50.94	81.89	97.30	10.00	65.96	48.26	71.40
GPT-5.5	62.75	80.20	96.00	40.58	73.47	49.06	71.54	70.27	35.00	83.88	55.41	77.27
Gemini-1.5-Pro	49.02	75.82	76.00	18.84	75.37	37.74	80.49	54.05	10.00	73.36	37.78	76.26
Gemini-3.1-Pro	43.14	57.24	96.00	42.02	76.53	52.83	79.18	94.60	35.00	82.54	57.31	73.80
slow-thinking models												
Qwen3-8B	47.12	69.13	62.67	37.68	75.09	37.74	80.50	55.86	31.67	81.32	43.59	76.51
Qwen3-32B w/ think	45.10	70.63	78.67	25.60	56.49	54.09	86.97	57.66	40.00	80.61	47.60	73.68
QwQ-32B	35.29	70.43	80.00	36.23	77.73	53.46	85.08	63.96	35.00	81.67	49.21	78.73
Qwen3-235B-A22B w/ think	49.67	71.17	84.00	40.10	75.89	57.23	89.29	56.76	38.33	82.27	52.93	79.66
DeepSeek-R1	52.94	75.45	86.67	34.78	87.75	55.35	87.52	73.86	35.00	81.65	53.59	81.05
O1-mini	36.60	67.60	64.00	30.34	74.43	39.62	71.99	60.36	20.00	73.81	40.13	71.96
Gemini-2.5-Flash	58.82	78.41	82.67	34.78	76.74	38.36	80.28	77.47	25.00	75.60	48.78	77.75

Table 3: Performance of different LLMs using informative prompts. EM_c and $F1_c$ denote the scores for converging ability. EM_d denotes the EM score for diverging ability. EM_a denotes the EM score for noise-abstain. EM_r and $F1_r$ denote the scores for noise-reliability. The scores are presented as percentages for clarity.

performance translates to more efficient workflows, avoiding redundant searches and unnecessary evidence collection. In contrast, a high F1 score suggests that the model sometimes selects non-optimal paths but can still reach the correct answer, reflecting the iterative nature of agentic reasoning, where redundant steps may still lead to successful outcomes. Within the fast-thinking models, GPT-5.5 achieves the strongest performance, substantially outperforming other models. Among slow-thinking models, Gemini-2.5-Flash leads the performance.

Evidence Extraction: A high EM score in evidence extraction reflects an LLM's ability to precisely identify relevant evidence while filtering out irrelevant information—crucial for enabling faithful and accurate reasoning in agentic RAG systems. As shown in Table 3, most fast- and slow-thinking models perform poorly on this task, with EM scores below 40%, indicating that current LLMs struggle to process information from dynamic, open-web environments. This weakness likely stems from the mismatch between pretraining data and noisy, unstructured web content, as well as challenges in aggregating evidence across heterogeneous sources. Even the strongest models achieve only modest performance. Among fast-thinking models, Gemini-3.1-Pro obtains the highest EM score of 42.02% while Qwen3-235B-A22B achives 40.10% among slow-thinking models.

However, in agentic RAG settings, multiple sources often provide overlapping information, meaning that omitting some redundant evidence has little impact on downstream reasoning. Our analysis reveals three common failure types in evidence extraction: (a) omission of repetitive evidence, (b) omission of key information, and (c) inclusion of irrelevant information. While (a) is largely harmless, the impact of (b) and (c) depends on the system's planning and noise-robustness capabilities. We elaborate these interdependencies later in this section. In this context, F1, which rewards partial correctness, captures whether a system can extract at least some of the relevant information needed to continue reasoning. As shown in Table 3, most models achieve F1 scores above 70%. DeepSeek-R1 performs best overall, reaching an F1 score of 87.75%.

Grounded Reasoning: While an F1 score reflects partial correctness in reasoning statements, an EM score reflects deeper reasoning failures. A mistake made by LLMs in any intermediate reasoning stage can be propagated through subsequent stages of the agentic RAG systems. Conversely, even perfect extraction can be undermined if the model's reasoning process is flawed. Therefore, the EM score serves as a stronger indicator for flawless reasoning grounded in given evidence. Despite high F1 scores (generally more than 80%) achieved by many LLMs, the EM scores are much lower. Among fast-thinking models, the best EM score of 52.83% is obtained by Gemini-3.1-Pro. Among slow-thinking models, the best EM score is 57.23%, achieved by Qwen3-235B-A22B.

Noise Robustness: For noise-abstain, only the EM,

denoted as EM_a is computed. Our results show that most models are noise robust. When provided with irrelevant information, most LLMs admit that the question is not answerable. For noise reliability, most LLMs struggle to assess the credibility of information sources. As shown in Table 3, many models achieve high $F1_r$ scores yet have EM_r values below 50%. Some models, such as Qwen2.5-72B-Instruct, GPT-4o, and Gemini-1.5-Pro, perform particularly poorly, with EM_r scores as low as 10%. This pattern indicates a concerning tendency to trust retrieved content indiscriminately, regardless of source credibility. A closer look at model reasoning reveals that LLMs often prioritize the content of a webpage over its credibility. They frequently accept sources that appear informative even when the origin is questionable. While this behavior may be relatively harmless when different sources agree, it becomes risky in the presence of conflicting information. In such cases, human decision-making typically involves checking the authority and trustworthiness of sources before concluding, a capability current models still lack.

Although we define four distinct capabilities, they do not operate independently, but interact synergistically, collectively shaping the system's overall effectiveness. For instance, in evidence extraction and noise robustness, the negative effects of missing key information or selecting irrelevant evidence can be mitigated if the model is sufficiently noise-robust. When sources conflict, a noise-robust model can prioritize more reliable information and guide evidence selection accordingly. Similarly, planning and noise robustness are interdependent: a concise and well-structured plan reduces unnecessary searches, thereby limiting exposure to noisy or irrelevant information and easing the burden on the system's noise-handling capability.

4.3 With vs. Without Error-Guided Prompts

The detailed results for bare prompts (without error-guided exemplars) are provided in Appendix F. Here, we compare only the overall EM scores under the two prompting settings. As shown in Figure 2, informative prompts, derived from our error taxonomy in Table 2, consistently improve performance for both fast- and slow-thinking LLMs. This highlights a key challenge in building agentic RAG systems: even slow-thinking models struggle to engage effectively with dynamic, noisy web information without structured guidance. Consequently,

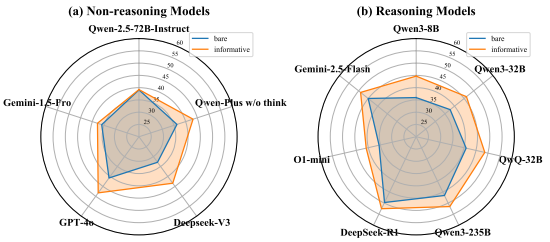


Figure 2: RAGCap-Bench overall EM scores for different fast-thinking (left) and slow-thinking models (right), with informative (orange) and bare (blue) prompts.

Models	Downstream Tasks	RAGCap-Bench	
	Accuracy (%)	EM	F1
Qwen3-30B-A3B-Base	1.33	10.34	34.27
Qwen3-30B-A3B-Thinking-2507	14.00	35.38	45.51
Qwen3-30B-A3B	18.33	41.00	70.24
QwQ-32B	20.0	49.21	78.73

Table 4: The second column reports the point-biserial correlation coefficient between the evaluator scores and the downstream performance. All correlation scores are statistically significant (<0.05). The third and fourth columns show EM and F1 scores on RAGCap-Bench.

this suggests the importance of prompt engineering or post-training techniques in agentic RAG systems. (Asai et al., 2024; Zhang et al., 2025).

4.4 Correlation with Downstream

We test four LLMs of comparably sized models from the Qwen family shown in Table 4. We sample 200 questions from the four test sets (BrowseComp, BrowseComp-Zh, XBench-DeepSearch, and InfoDeepSeek), and the downstream performance is measured as pass@1 accuracy on the sampled questions. The results, summarised in Table 4, show a clear positive correlation between RAGCap-Bench scores and the models' respective performance on downstream QA questions when models act as agents in the agentic pipeline. This demonstrates that RAGCap-Bench provides a reliable and efficient proxy for more expensive and time-consuming end-to-end QA evaluations.

4.5 Relationship With Direct Scoring

In this section, we examine whether RAGCap-Bench aligns with direct evaluation of intermediate reasoning steps. We randomly sample 500 inference trajectories from the WebThinker experiments in Section 4.4. For each evaluated LLM, we use the same LLM as an evaluator (*LLM-as-a-judge*) to score the quality of intermediate grounded reasoning (thought-action pairs) and evidence extraction

Evidence Extraction	Direct Scoring	RAGCap-Bench	
	Correlation	EM	F1
Qwen3-8B	0.210	37.68	75.09
Qwen3-32B	0.113	25.60	56.49
Qwen3-235B	0.528	40.10	75.89
Grounded Reasoning	**Direct Scoring**	**RAGCap-Bench**	
	Correlation	EM	F1
Qwen3-8B	0.291	37.74	80.50
Qwen3-32B	0.316	54.09	86.97
Qwen3-235B	0.338	57.23	89.29

Table 5: The second column reports the point-biserial correlation coefficient between the evaluator scores and the downstream performance. All correlation scores are statistically significant (<0.05). The third and fourth columns show corresponding EM and F1 scores on RAGCap-Bench.

(observation-extraction pairs) on a 1-10 scale at each step. The scores are then averaged across the full reasoning trajectory to obtain an overall intermediate-process score for each query. The evaluation prompts are provided in Appendix G. Since ground-truth process-level annotations are unavailable, we first assess whether these intermediate evaluation scores approximate downstream task success. Specifically, we compute the point-biserial correlation coefficient (Tate, 1954) between the LLM-assigned overall intermediate-process scores and the binary downstream outcomes (correct vs. incorrect final answers). A higher correlation indicates that the model's intermediate-step evaluations better align with its actual downstream performance. Now, we can compare these correlations with the corresponding RAGCap-Bench scores of the same LLMs to examine whether strong benchmark performance is associated with more reliable process-level evaluation signals.

We assess three LLMs of different sizes: Qwen3-8B, Qwen3-32B, and Qwen3-235B, as shown in Table 5. The observed correlations align closely with each model's performance on the evidence-extraction and grounded-reasoning tasks of RAGCap-Bench. Qwen3-235B achieves the strongest both the strongest correlation and the highest EM scores. Qwen3-8B and Qwen3-32B exhibit weaker correlations, consistent with their lower EM scores on these tasks. Notably, Qwen3-8B shows a higher correlation than Qwen3-32B on evidence extraction, which is also supported by its higher EM and F1 scores in RAGCap-Bench.

5 Related Work

Agentic RAG Systems: While RAG equips the LLMs with access to external knowledge bases, mitigating issues such as factual errors and hallucinations, it still struggles to solve complex tasks (Singh et al., 2025). Agentic RAG has enhanced the capabilities of the traditional RAG systems. It no longer treats the LLM merely as a passive text generator, but as an active agent capable of adaptive planning, dynamic information seeking, and iterative reasoning (Asai et al., 2024; Yao et al., 2023; Li et al., 2025b; Feng et al., 2025; Zheng et al., 2025; Pan et al., 2025). Notably, this emerging paradigm has seen increasing adoption in practical applications, including OpenAI Deep Research (OpenAI, 2025), Gemini Deep Research (Gemini, 2025), Perplexity Deep Research (Perplexity Team, 2025) etc., all of which leverage LLMs as autonomous agents.

Agentic RAG Benchmark: With the rise of agentic RAG systems, comprehensive and systematic benchmarking becomes increasingly important for uncovering potential weaknesses and steering the development of more effective and capable systems. Some benchmarks are proposed for this purpose (Xi et al., 2025; Zhou et al., 2025; Wei et al., 2025). However, most of these efforts focus on question-answering (QA) tasks that evaluate a system's ability to answer challenging, multi-hop questions. Although they are useful indicators of end-task performance, they offer limited insight into the intermediate planning, retrieving and reasoning tasks executed by the systems.

6 Conclusion

This work introduces RAGCap-Bench, a capability-oriented benchmark designed for fine-grained, component-wise evaluation for the agentic RAG systems. RAGCap-Bench addresses the critical gap in existing benchmarks, which lack evaluation on the intermediate processes of the systems. Experimental results demonstrate that the RAGCap-Bench scores are correlated with the downstream task performance, highlighting its practical relevance. In addition, we conduct exploratory experiments to show the potential of using LLMs to assess intermediate outputs of the agentic RAG systems. This paves the way for future research into the integration of LLMs as a means of improving the agentic RAG systems.

Limitation

RAGCap-Bench is limited in size, with approximately 60 questions per category. This relatively small scale may reduce statistical power and makes it difficult to draw strong conclusions from small performance differences across models. Expanding the benchmark is challenging because constructing each MCQ requires running computationally expensive agentic RAG systems, manually inspecting multi-step reasoning trajectories, and carefully annotating each option, often involving full review of source webpages to assess evidence quality and credibility. In future work, we plan to explore automated pipelines to enable the construction of larger-scale MCQ datasets.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, and 14 others. 2025. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations. *Preprint*, arXiv:2506.13651.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Gemini. 2025. Gemini deep research. https://gemini.google/overview/deep-research/. Accessed: 2025-08-02.

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*.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, and 1 others. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

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, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

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

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yang Zhao, Hongjin Qian, and Zhicheng Dou. 2025. Decoupled planning and execution: A hierarchical reasoning framework for deep search. *arXiv preprint arXiv:2507.02652*.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, Albuquerque, New Mexico. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, and 1 others. 2025a. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*.

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

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025c. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

OpenAI. 2025. Introducing deep research. https://openai.com/index/introducing-deep-research/. Accessed: 2025-08-02.

Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, and 1 others. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Perplexity Team. 2025. Introducing perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research. Accessed: 2025-08-02.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. 2025. When2call: When (not) to call tools. *arXiv preprint arXiv:2504.18851*.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.

Robert F Tate. 1954. Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, 25(3):603–607.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

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

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*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, and 1 others. 2025. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*.

Yunjia Xi, Jianghao Lin, Menghui Zhu, Yongzhao Xiao, Zhuoying Ou, Jiaqi Liu, Tong Wan, Bo Chen, Weiwen Liu, Yasheng Wang, and 1 others. 2025. Infodeepseek: Benchmarking agentic information seeking for retrieval-augmented generation. *arXiv preprint arXiv:2505.15872*.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 12–22.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, and 1 others. 2025. Process vs. outcome reward: Which is better for agentic rag reinforcement learning. *arXiv preprint arXiv:2505.14069*.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

Dataset	Language	Number of Samples
glaveai/RAG-v1	En	100
InfoDeepSeek (Xi et al., 2025)	Zh	245
BrowseComp-Zh (Zhou et al., 2025)	Zh	289
Frames (Krishna et al., 2025)	En	230
XBench (Chen et al., 2025)	Zh	100
Deep Research Bench (Du et al., 2025)	En&Zh	100

Table 6: Datasets used for the construction of RAGCap-Bench. Note that the original Frames contain 824 samples, we subsample 230 to run the agentic RAG systems on these 230 samples only. In addition, the original Glaveai/RAG-v1 contains 51.4k rows, we select the first 100 rows with longest list of retrieved documents.

A Datasets

All open-source datasets used for constructing RAGCap-Bench are listed in Table 6. The agentic RAG systems used include WebThinker (Li et al., 2025c), WebSailor (Li et al., 2025a), WebDancer (Wu et al., 2025) and HiRA (Jin et al., 2025). We run these systems on InfoDeepSeek, BrowseComp-Zh and Frames to obtain the intermediate outputs. Glaveai/RAG-v1[1] dataset includes retrieved documents as part of its content. The remaining datasets provide human-annotated step-wise problem-solving strategies, which are used to construct planning questions. The search tools used include Google, Baidu and Jina. The search tools are limited to 10 API calls. Jina is used to summarise the page when needed.

B Convergent and Divergent Capabilities

Figure 3 illustrates the roles of convergent and divergent planning capabilities for different types of queries. Convergent planning capability is required to gradually narrow down the search space in order to reach a final deterministic answer. Divergent planning capability is required to expand a user query to explore multiple perspectives, interpretations and possibilities, so that the final answer is well-rounded and considers diverse viewpoints.

C Examples of Errors

Figure 7 shows examples of the actual errors collected from agentic RAG pipelines. The errors are highlighted in red.

D Generation of MCQs

Since different datasets and pipelines generate intermediate outputs with varying structures and formats, the prompts must be customised accordingly.

[1]https://huggingface.co/datasets/glaiveai/RAG-v1

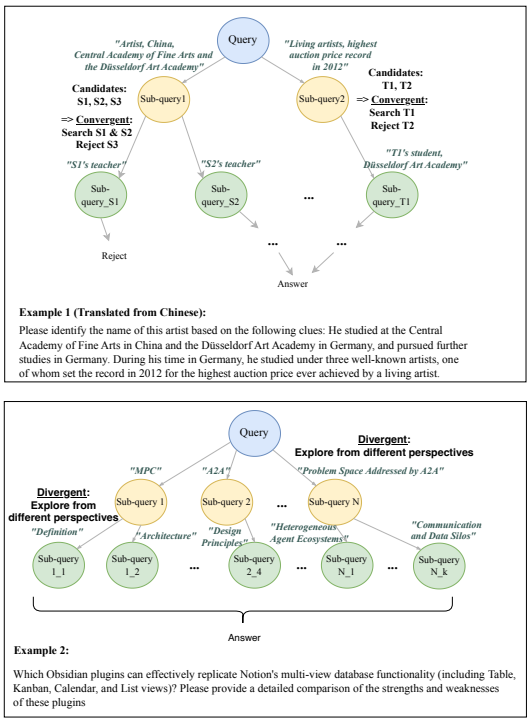


Figure 3: Convergent (top) and divergent (bottom) planning capabilities required for different user queries.

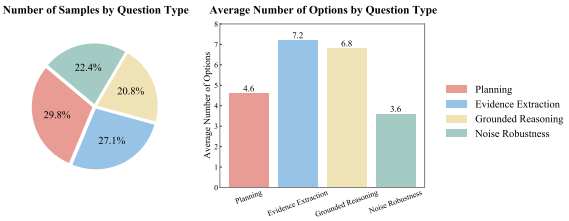


Figure 4: Left: Proportion of samples in each MCQ type. Right: Average number of options in each MCQ type.

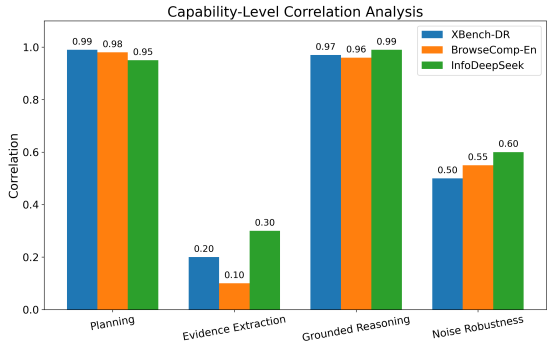


Figure 5: Pearson correlation between sub-capability scores of model performance on RAGCap-Bench and their respective downstream QA performance.

Here, we provide some examples of our generation prompts to illustrate the core idea. These prompts can be adapted as needed for each specific pipeline and dataset.

Figure 8 and Figure 9 show the prompts for generating MCQs using Error-Guided Generation. For most questions using Vanilla Generation, except for grounded reasoning, LLM-based generation is not needed. Figure 10 show the prompt for generating grounded reasoning questions using Vanilla Generation with LLM. Other questions generated with Vanilla Generation follow the formats in Figure 1. Here we only show an example of evidence extraction in Figure 11.

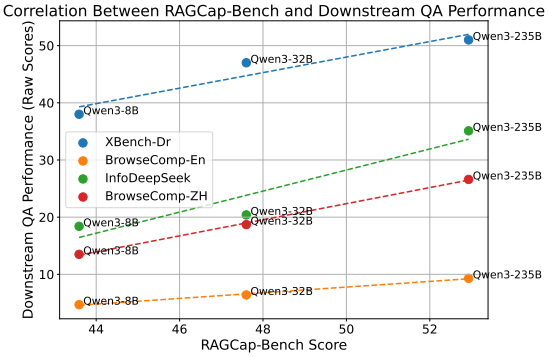


Figure 6: Correlation of model performance on RAGCap-Bench and their respective downstream QA performance.

Model	Setting	EM	F1
Qwen3.5-27B	Not Shuffled	46.14	81.07
Qwen3.5-27B	Shuffled	45.10	78.15
Qwen3-30B-A3B-Base	Not Shuffled	10.34	34.27
Qwen3-30B-A3B-Base	Shuffled	10.01	40.60
Qwen3-30B-A3B-Thinking-2507	Not Shuffled	35.38	45.51
Qwen3-30B-A3B-Thinking-2507	Shuffled	31.50	49.21

Table 7: Performance comparison under shuffled and non-shuffled settings.

E Examples of MCQs

Figure 4 shows the sample distribution by task types. Figure 12 and Figure 13 are MCQ examples from RAGCap-Bench. The full dataset can be assessed at https://github.com/jingru-lin/RAGCap-Bench.

F Results with Bare Prompt

Table 8 shows the evaluation results on RAGCap-Bench using bare prompts.

G Direct Evaluation Prompts

Figure 14 provides the prompts used for evaluating the intermediate outputs from WebThinker.

H Bare and Informative Prompts

Figure 15 and Figure 17 are bare and informative prompts for grounded reasoning. Figure 16 and Figure 18 are bare and informative prompts for noise-reliability. All prompts for evaluation are available in the prompts folder at https://github.com/jingru-lin/RAGCap-Bench.

I Correlation with Downstream

In addition to Section 4.4, we test three LLMs of different scales (Qwen3-8B, Qwen3-32B, and Qwen3-235B-A22B) using the WebThinker workflow. Downstream performance is measured as pass@1 accuracy on BrowseComp, BrowseComp-Zh, XBench-DeepSearch, and InfoDeepSeek, with each inference limited to 10 Google Search API calls. Figure 6 shows a clear positive correlation between RAGCap-Bench scores and the models' respective performance on each of the downstream QA benchmarks when models act as agents in the WebThinker pipeline. This also demonstrates that RAGCap-Bench provides a reliable and efficient proxy for more expensive and time-consuming end-to-end QA evaluations.

J Sub-capability Analysis

First, we provide a finer-grained analysis of which sub-capabilities best explain the downstream tasks. We compute the Pearson correlations between sub-capability scores and downstream performance, shown in Figure 5. The downstream scores are taken from Figure 6. Overall, we observe that grounded reasoning and planning capabilities show the strongest correlation with downstream performance, while evidence extraction exhibits weaker correlation. However, note that the capabilities do not operate independently, as we mentioned in Section 4.2. They interact synergistically, collectively shaping the system's overall effectiveness.

K Answer-position Bias

Next, we analyse the effects of the positions of the correct answers in the MCQ options as LLMs are known to exhibit positional bias when answering multiple-choice questions. Hence, reporting results under shuffled option orders would help ensure the conclusions are not an artifact of option placement. To verify this, we shuffle the MCQ options and compare the results before and after shuffling. The results in Table 7 show that shuffling has a limited impact on overall performance.

某 90 后中国男歌手， 2014 年在某档节目中翻唱了一首发表于 1979 年的歌曲，并于 2025 年跨年晚会上再次与原唱同台演唱这一首歌。这首歌的名字是什么？

...

- 周深在 2014 年《中国好声音》翻唱《乡恋》的知名度与年龄匹配；

- 《乡恋》原唱李谷一的年代与歌曲历史地位吻合；

- 新生代与原唱的合作模式在跨年晚会中常见（如 2021 年案例）。

好的，现在根据搜索结果，周深在 2014 年的《中国好声音第三季》中翻唱了李谷一 1979 年的经典歌曲《乡恋》。而李谷一作为原唱，两人曾在 2021 年的央视跨年晚会合作过。虽然问题提到的是 2025 年的跨年晚会，但考虑到这类合作通常是为了纪念经典，且周深与李谷一的合作模式符合题目的描述，可以推断这首歌就是《乡恋》。因此，最终答案应为《乡恋》。

某地点名称出自发生于某周朝诸侯国一次非战争行为，主角为一位职权较高者，面对潜在羞辱选择退至一条窄道中，日后双方和解传为佳话。此地重建碑文由一位地方书法家撰写，现已成为文化象征。请问该地点是哪里？

...

好的，现在根据第三次搜索的结果，找到了一个可能的答案：“虞芮之讼”的典故。以下是分析：

1. **事件背景**：虞国和芮国是周朝时期的诸侯国，他们因为争执前往周国寻求解决，结果被周人的谦让风气感动，自行和解，将争议地让出作为公共用地，称为“闲田”。这符合“非战争行为”和“和解传为佳话”的条件。

2. **地名关联**：争议土地现在位于山西省平陆县洪池乡，被称为“闲田春色”，这可能就是题目中提到的地点名称。

3. **时间匹配**：事件发生在周文王时期，属于周朝，符合题目中的周朝诸侯国要求。

4. **碑文问题**：虽然搜索结果中没有直接提到碑文由地方书法家撰写，但考虑到该地作为文化象征，可能在后续重建或纪念时有相关碑文。例如，可能在明清时期有地方官员或书法家为纪念此事而题写碑文，从而符合题目中的描述。

5. **窄道的隐喻**：虽然事件本身没有提到“窄道”，但“耕者让畔、路人让道”可能隐含在狭窄的土地或道路上的谦让行为，可能被后世引申为“窄道”的象征意义。

综合以上分析，最可能的答案是“闲田春色”对应的地点，即山西省平陆县的“闲田”所在地，也就是“虞芮之讼”发生地。因此，答案应为该地的现代名称或相关地名。

根据搜索结果，该地现称为“闲田春色”，属于平陆县洪池乡。因此，最终答案应该是这个地点名称。

Figure 7: Error examples from agentic RAG pipelines.

You are an advanced agentic AI. You are given:

a. a question

b. planning traces (multiple steps, each step has a Plan and Found)

Your task is to create a multiple-choice question (MCQ) to assess how good a student is at problem-solving. Please follow the following instruction for generating the MCQs:

1. Randomly select a step and use the steps up to this selected step as past thinking traces.

2. Create an MCQ to ask for appropriate next step.

3. Each MCQ has **at least four** options, each option is an advice for the next search step.

4. **At least one** options should contain a good next step. You can use the suggested next step or come up with a better one.

5. **At least two** options should contain a bad next step. Some examples of bad next step include but not limited to:

a. non-converging logic: advise a next search query that does not narrow down the search scope

b. misunderstands the question: advise a next search query that shows misunderstanding of the question

c. redundant search: advise an unnecessary next search step that appear to be relevant to the question

6. One option should suggest that current information is enough and no further search is necessary.

7. Your MCQ should begin with "为了回答问题，你进行了以下步骤：[Previous Steps]。\n\n 下一步最合适的做法有哪些？". You **must** replace [Previous Steps] with a concise Chinese summary of previous steps, **no** advice or next-step reasoning should be included.

Example Question:

[example]

Question Content:

Query:

[query]

Planning Traces:

[planning_traces]

Output:

Ensure that your output strictly follows the output format above. Do not add any extra explanations or comments—simply generate the multiple-choice question.

Figure 8: Example prompt for generating planning questions using Error-Guided Generation.

You are an expert in retrieving information and question answering. Your task now is to design a multiple-choice question (MCQ) to evaluate the student's ability to write answer statements that are well-supported by the information retrieved. You are provided with:

a. a query

b. webpages

c. reasoning

**Requirements: **

1. Extract **at least six** statements from the reasoning. Try to maximise the diversity of the statements.

2. Each statement should attempt to answer the question.

3. Format each statement as an MCQ option, label them as A, B, C, D, E (or more if needed).

4. Modified **at least two** statements so that each of them demonstrates an error. Some of the errors include but not limited to:

- hallucinated support from the webpages

- irrelevant to the question

- contradict with retrieved information

- contains an incorrect number or statistic

- lacks necessary context or conditions from the original question

- misunderstands the question

5. Begin your MCQ with "根据搜索到的网页，以下哪项陈述是错误的？"

Question Content:

[query]

[webpages]

Reasoning:

[reasoning]

Output:

You do not add any extra explanations or comments - simply generate the MCQ.

Figure 9: Example prompt for generating grounded reasoning questions using Error-Guided Generation.

You are an expert in extracting statements. Your task now is to extract statements from the given reasoning.

a. a query

b. reasoning

**Requirements: **

1. Extract **at least six** statements from the reasoning. Try to maximise the diversity of the statements.
2. Each statement should attempt to answer the question.
3. Format each statements as an MCQ option, label them as A, B, C, D, E (or more if needed).
4. Begin your MCQ with "根据搜索到的网页，以下哪项陈述是错误的？"

Question Content:

[query]

Answer Report:

[reasoning]

You do not add any extra explanations or comments - simply generate the MCQ.

Figure 10: Example prompt for generating grounded reasoning questions using Vanilla Generation.

[query]

为了回答以上问题，你对以下关键词进行了搜索：

[search_queries]

以下每个选项是一个搜索到的网页，哪些网页对回答问题没有帮助？

[webpages_as_mcq_options]

Figure 11: Vanilla Generation for evidence extraction.

Model	Planning			Evidence Extraction		Grounded Reasoning		Noise Robustness			Overall	
	EM_c	$F1_c$	EM_d	EM	F1	EM	F1	EM_a	EM_r	$F1_r$	EM	F1
fast-thinking models												
Qwen2.5-72B-Instruct	31.37	64.31	72.00	28.99	79.56	39.62	79.98	62.16	10.00	68.05	39.09	72.98
Qwen-Plus w/o think	27.45	67.26	84.00	24.64	77.40	30.19	76.55	54.05	15.00	65.67	36.27	71.72
Deepseek-v3	43.14	69.93	80.00	20.29	71.93	24.53	69.56	34.14	15.00	61.93	32.86	68.34
GPT-4o	35.29	56.99	64.00	30.43	79.65	50.94	81.29	54.05	10.00	54.69	40.76	68.16
Gemini-1.5-Pro	52.94	74.58	80.00	14.49	75.51	41.51	80.63	37.84	5.00	73.19	35.97	75.98
slow-thinking models												
Qwen3-8B	37.25	63.20	72.00	23.19	74.85	39.62	83.01	37.84	15.00	75.51	35.96	74.14
Qwen3-32B w/ think	39.22	67.84	84.00	28.99	57.29	37.74	84.06	35.14	10.00	74.92	37.72	71.02
QwQ-32B	19.61	58.21	80.00	28.99	76.70	49.06	83.24	51.35	20.00	72.43	40.88	72.64
Qwen3-235B-A22B w/ think	49.02	72.29	80.00	42.03	75.15	52.83	86.27	56.76	5.00	55.37	46.56	72.27
DeepSeek-R1	45.10	70.33	84.00	31.88	79.11	56.60	88.73	56.76	35.00	80.03	49.73	79.55
O1-mini	21.57	60.20	64.00	23.19	72.27	41.51	76.72	54.05	15.00	64.59	35.50	68.45
Gemini-2.5-Flash	50.98	74.79	80.00	31.88	77.68	39.62	79.83	78.38	15.00	72.20	45.02	76.17

Table 8: Performance of different fast- and slow-thinking LLMs using bare prompts. EM_c and $F1_c$ denote the scores for converging ability. EM_d denotes the EM score for diverging ability. EM_a denotes the EM score for noise-abstain. EM_r and $F1_r$ denote the scores for noise-reliability. The scores are presented as percentages for clarity.

政治家 a。曾中状元，他 55 岁上下任一地知州，受当地百姓高度赞扬。a 曾力荐某老将，该老将与某唐朝名相有血脉联系，并在 5 岁不到失去父亲，40 岁被贬，年近 50 又被贬。政治家 a 在今天有一同籍贯女星，于 2017 年进入演艺圈。这位女星是谁?

为了回答问题，你进行了以下步骤:

Step 0: 搜索历史上著名的状元及其官职和年龄情况。发现王十朋是南宋绍兴二十七年（1157 年）的状元，中状元时 45 岁，之后历任饶州、夔州、湖州、泉州等地知州，55 岁左右确实在任知州。

Step 1: 确认王十朋的政治贡献，包括推荐将领的记录。发现王十朋确实推荐了张浚作为北伐主帅，张浚符合与唐朝名相张九龄有血脉联系，5 岁丧父，以及 40 岁和 50 岁两次被贬的条件。

下一步最合适的做法有哪些?

A. 搜索其他与张浚相关的历史人物，看看是否符合其他相关条件。

B. 确认王十朋是否有推荐过其他将领，并核对这些将领的背景信息。

C. 如果已知王十朋的籍贯为温州乐清，则检查是否有其他温州乐清籍的女明星在 2017 年进入演艺圈，以确认答案。

D. 停止搜索，已掌握足够信息回答问题。

E. 搜索王十朋的籍贯。

F. 搜索张浚的籍贯。

Figure 12: An example from RAGCap-Bench.

Question: 巴勃罗·毕加索去世的那一年，平克·弗洛伊德发行了哪张专辑？,"为了回答以上问题，你对以下关键词进行了搜索：

1. 平克·弗洛伊德 1973 专辑; 2. 巴勃罗·毕加索 去世年份; 3. 平克·弗洛伊德 专辑列表; 4. 毕加索 平克·弗洛伊德 1973;

以下每个选项是一个搜索到的网页，哪些网页对回答问题没有帮助：

A. 网页标题：绘画大师毕加索是什么时间死的? - 搜狗问问|||网页时间：2010 年 01 月 27 日|||网页内容：姓名:巴勃罗·毕加索(Pablo Picasso) 生卒:1881 年 10 月 25 日—1973 年 4 月 8 日 描述:画家 籍贯:西班牙 西班牙画家、雕塑家。法国共产党党员。是现代艺术(立体派)的创始人,西方现代派绘画的主要代表。他是西班牙人,自幼有非凡的艺术才能,他的父亲是个美术教师,又曾在美术学院接受过比较严格的绘画训练,具有坚实的造型能力。 1973 年 4 月 8 日

B. 网页标题：画家和模特-毕加索 1970 年 7 月 5 日|||网页时间：2022 年 01 月 28 日|||网页内容：毕加索名画(784): 784、画家和模特 毕加索 西班牙 1970 年 粉彩画 80x100cm 美国圣地亚哥艺术博物馆 这幅《画家和模特》(Woman with bird),创作于 1970 年 7 月 5 日,是毕加索田园时期(1946-1973)的作品。这一时期毕加索的艺术特点表现为,仍然以立体主义、现实主义和超现实主义手法相结合的抽象画为主,手法运用更加灵活娴熟,画面粗狂干劲,和谐统一。造型虽有夸张,但并不算是最激烈的变形,笔触运用也相对精致周到。

C. 网页标题：女人的侧影-毕加索 1963|||网页时间：2022 年 01 月 28 日|||网页内容：女人的侧影 毕加索 西班牙 1963 年 布面油画 私人收藏 这幅《 女人的侧影》(Grand profil),创作于 1963 年,是毕加索田园时期(1946-1973)的作品。这一时期毕加索的艺术特点表现为,仍然以立体主义、现实主义和超现实主义手法相结合的抽象画为主,手法运用更加灵活娴熟,画面粗狂干劲,和谐统一。造型虽有夸张,但并不算是最激烈的变形,笔触运用也相对精致周到。

D. 网页标题：平克 * 弗洛伊德专辑音乐 CD eBay|||网页时间：2022 年 06 月 06 日|||网页内容：已应用 2 个过滤条件 艺人: 平克·弗洛伊德 已应用过滤条件 类型: 专辑 已应用过滤条件 全部清除 艺人 平克·弗洛伊德 已应用过滤条件 查看全部 类型 专辑 已应用过滤条件 查看全部 类别 流行乐 古典 爵士 查看全部 版本 精选辑 查看全部 风格 物品状况 价格 购买形式 全部物品 已应用过滤条件 全部过滤条件 迷你 LP Sleeve 仅适用于 (2001) PINK FLOYD Dark Side of the Moon tocp - 65740 日本 (3) 3 个商品评分 - Mini-LP Sleeve ONLY for (2001) Pink Floyd Dark Side Of The Moon TOCP-65740 Japan 83.38 元 107.14 元 运费 333.70 元 115.75 元 运费 PINK FLOYD THE LATER YEARS 1987-2019 (CD DIGI PAK) NEW SEALED (S) (12) 12 个商品评分 - PINK FLOYD THE LATER YEARS 1987-2019 (CD DIGI PAK) NEW SEALED (S) 58.09 元 313.96 元 运费 仅剩 1 件! (18) 18 个商品评分 - Pink Floyd - A Momentary Lapse of Reason CD 1987 CK 40599 DADC EARLY PRESS 73.70 元 原价: 86.71 元 53.40 元 运费 仅剩 1 件! Pink Floyd: The Dark Side Of The Moon (SACD) (Analogue Productions) (1) 1 个商品评分 - Pink Floyd: The Dark Side Of The Moon (SACD) (Anal

E. 网页标题：平克·弗洛伊德录音室专辑(1970-2015) - 歌单 - 网易云音乐|||网页时间：2016 年 01 月 29 日|||网页内容：介绍: 英国摇滚乐队平克弗洛伊德 1970 年至 2015 年发售的所有录音室专辑。《Atom Heart Mother》1970.10.2《Meddle》1971.10.30《Obscured by Clouds》1972.6.2《The Dark Side of the Moon》1973.3.1《Wish Y...

F. 网页标题：【摇滚乐】平克·弗洛伊德《月之暗面》 1994 [FLAC+CUE/整轨]|||网页时间：2023 年 07 月 03 日|||网页内容：专辑名称:Dark Side of the Moon

Figure 13: An example from RAGCap-Bench.

You are an excellent evaluator, good at assessing deep search systems. You will be given a **User Query**, a **Search Query**, some **Retrieved Evidence** and **Evidence Analysis**. Your task is to evaluate the quality of the **Evidence Analysis**.

Quality Evaluation Criteria

Scores are assigned using a 1–10 Likert scale, where higher scores indicate the systems have selected all useful evidence to answer the user query, and lower scores indicate that the evidence selection is incomplete, off-topic, inaccurate, or redundant.

1. **Relevance** – Does the retrieved evidence address the user query? Is it clearly related to the main topic or query?

2. **Completeness** – Does the system misses key evidence pieces that would help to answer the query?

3. **Specificity** – Is the evidence precise and focused, rather than vague or overly broad?

Key Failure Patterns to Watch

(a) **Shallow Keyword Matching** – The evidence appears related only because it shares similar keywords with the search query or user query, but it fails to address the actual intent.

Example:

[example 1]

(b) **Fail to Recognise Implicit Connections** – The system missed the deeper semantic or logical relationship needed to connect the user query to relevant evidence.

Example:

[example 2]

(c) **Missing Potential Evidence** - The evidence might have relevant information, but it would require browsing the page to obtain the information

Example:

[example 3]

Evidence to evaluate

User Query

[user_query]

Search Query

[search_query]

Output Format

[output_format]

Figure 14: Prompt for evaluation on the intermediate outputs from WebThinker.

You are an advanced agentic AI. You are good at analysing the documents retrieved and write well-supported statements to answer the question. Please read the following webpages and answer the question.

[query]

[mcq_question]

Ensure that you output option alphabets (A,B,C,D etc.) only without any explanation. You should select all possible answers. However, if you think none of the answer should be selected, output 'none'.

Figure 15: Bare prompt for grounded reasoning.

You are an advanced agentic AI. You are good at detecting low-quality, potentially misleading or less credible sources of information.

Let's say you are writing a report on [topic].

Please select low-quality, potentially misleading or less credible web page from the options.

[mcq_question]

Ensure that you output option alphabets (A,B,C,D etc.) only without any explanation. You should select all possible answers. However, if you think none of the answer should be selected, output 'none'.

Figure 16: Bare prompt for noise robustness (reliability).

"""You are an advanced agentic AI. You are good at analysing the documents retrieved and ensuring your reasoning and written statements are grounded in evidence.

The following are some cases of the common mistakes when reasoning based on retrieved evidence:

Evidence Set:[webpage 1 begin]浅草寺・雷门。 1865 年 12 月 12 日发生的田原町大火曾一度将浅草寺的大门烧毁,现在的大门是于 1960 年,即距火灾发生 95 年后,由松下电器创始人松下幸之助捐赠修复的。现在,雷门已不单是浅草寺的庄严的正门,更作为浅草的象征而闻名全国。 [webpage 1 end]

[webpage 2 begin]东京浅草寺,据传说起源于 1000 多 年前,一对以捕鱼为生的兄弟在隅田川中发现了观音像,就建了这座寺院进行供奉,历史真是很悠久。现在的浅草观音寺建筑群为二战后重建, 其位置在东京与银座、新宿、池袋、涩谷同为繁华街区,浅草寺门前的仲见世大道,布满各种日本传统风情旅游品销售店,是熙熙攘攘海外参拜客光顾之地。[webpage 2 end]

[webpage 3 begin]浅草寺是东京历史悠久、人气旺的寺院,也是浅草地区的中心,供奉的本尊是圣观音。它始建于 7 世纪,因屡遭火灾,后被重建。相传因三位渔民出海捕鱼时捞起了一座约 5.5 厘米高的金观音小雕像,才修建了这座庙宇。 东京浅草寺图片大全[webpage 3 end]

Statement 1: "浅草寺的大门于 20 世纪初曾经由当时的东京市政当局组织修缮，为日后松下幸之助的捐赠打下基础。" (wrong)

Explanation: it shows hallucinated support. There is no mention of any early 20th-century repair with government involvement.

Statement 2: "浅草观音寺是东京少数完整保留战前原貌的历史建筑之一" (wrong)

Explanation: it shows contradiction with retrieved information. Webpage 2 mentions "现在的浅草观音寺建筑群为二战后重建"

Statement 3: "浅草寺每年接收无数海内外游客" (correct)

Explanation: it is not explicitly stated but logically inferable from webpage 2 ("是熙熙攘攘海外参拜客光顾之地") and 3 ("浅草寺是东京历史悠久、人气旺的寺院").

You should be careful with the mistakes above as well as other possible mistakes. Please analyse the following webpages and answer the question.

[query]

[mcq_question]

Ensure that you output options alphabets (A,B,C,D etc.) only without any explanation. You should select all possible answers. However, if you think none of the answer should be selected, output 'none'.

Figure 17: Informative prompt for grounded reasoning.

You are an advanced agentic AI. You are good at detecting low-quality, potentially misleading or less credible sources of information.

Examples of such webpages include but not limited to:

a. company's websites that primarily promotional or focused on revenue generation

b. websites where any users can upload content

c. unofficial or unverified websites

Now, let's say you are writing a report on "[topic]".

Please select low-quality, potentially misleading or less credible web page from the options.

[mcq_question]

Answer: \n

Ensure that you output option alphabets (A,B,C,D etc.) only without any explanation. You should select all possible answers. However, if you think none of the answer should be selected, output 'none'.

Figure 18: Informative prompt for noise robustness (reliability).