RAISE: Enhancing Scientific Reasoning in LLMs via Step-by-Step Retrieval

Minhae Oh[1], Jeonghye Kim[2], Nakyung Lee[1], Donggeon Seo[3], Taeuk Kim[3], Jungwoo Lee[1]

[1]Seoul National University, [2]KAIST, [3]Hanyang University

Abstract

Scientific reasoning requires not only long-chain reasoning processes, but also knowledge of domain-specific terminologies and adaptation to updated findings. To deal with these challenges for scientific reasoning, we introduce **RAISE**, a step-by-step retrieval-augmented framework which retrieves logically relevant documents from in-the-wild corpus. RAISE is divided into three steps: problem decomposition, logical query generation, and logical retrieval. We observe that RAISE consistently outperforms other baselines on scientific reasoning benchmarks. We analyze that unlike other baselines, RAISE retrieves documents that are not only similar in terms of the domain knowledge, but also documents logically more relevant.

1 Introduction

Large language models (LLMs) have shown strong potential for scientific reasoning, which demands advanced reasoning skills, domain-specific terminology, and up-to-date knowledge [45, 42, 27, 30]. Two common strategies are step-wise reasoning, which solves complex problems through structured intermediate steps [37, 47, 12, 18, 41], and retrieval-augmented generation (RAG), which mitigates hallucinations by providing external evidence [21, 1, 50, 39]. Recent work combines them, but often targets simpler multi-hop QA or assumes curated, task-specific corpora [9, 44, 5, 11, 34], unlike open-domain sources such as Wikipedia. Solving challenging scientific reasoning tasks, such as graduate-level biology or chemistry, using an in-the-wild corpus is difficult since merely retrieving superficial knowledge is insufficient. Instead, the retrieved information should contain relevant logical connections needed to solve the problem [30]. Moreover, the knowledge required for each intermediate step can vary significantly even within the same problem. Without considering the evolving information needed for each reasoning process, RAG might even deteriorate the downstream task performance. The question of *what to search for* and *how to retrieve* the appropriate external knowledge for each step when solving scientific reasoning tasks is underexplored.

To address these challenges, we introduce **RAISE** (Step-by-Step **R**etrieval-**A**ugmented **I**nference for **S**cientific r**E**asoning), a retrieval-augmented framework tailored for step-wise scientific reasoning. Our framework consists of three stages: (1) problem decomposition, where LLMs break down the original question into subquestions along with search queries; (2) logical query generation, which reformulates each search query into a logic-enriched query that captures the reasoning needed to solve the subquestion; and (3) logical retrieval, which retrieves step-specific documents from an open-domain corpus, ensuring the retrieved information is logically relevant rather than superficially domain similar. Instead of assuming task-relevant or well-curated retrieval source, such as question-answer pool of relevant domains, we retrieve from in-the-wild source such as Wikipedia, which enables applying to challenging real-world scenarios. Evaluated on GPQA, SuperGPQA, and MMLU, RAISE consistently outperforms baselines using either RAG or problem decomposition alone, demonstrating its ability to retrieve step-specific, logically relevant information essential for solving

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

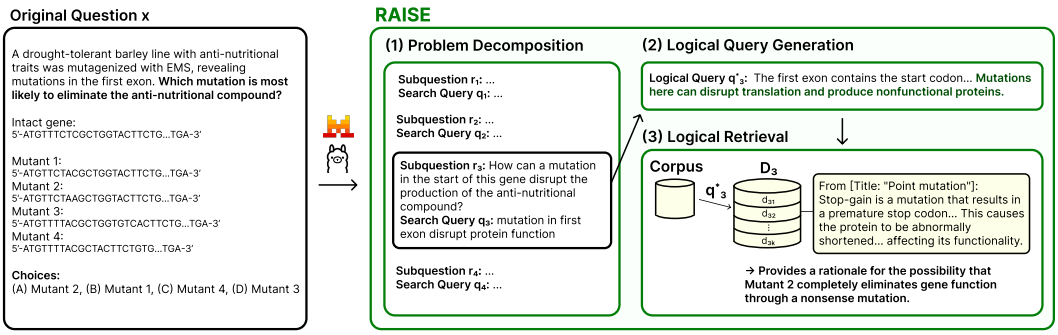

Figure 1: **Overview of RAISE.** RAISE is divided into three steps: (1) Problem Decomposition, (2) Logical Query Generation, and (3) Logical Retrieval.

complex scientific reasoning tasks. While this work focuses on scientific reasoning benchmarks, similar challenges arise in mathematical reasoning, which also requires precise, multi-step logical inference.

2 Preliminary

Step-by-Step Reasoning in LLMs. LLMs are capable of performing multi-step reasoning over complex input queries by internally chaining intermediate inferences. This step-by-step reasoning process involves decomposing a question into sub-problems, maintaining coherence across steps, and generating a final answer. Formally, given a query x, the model implicitly constructs a latent reasoning trajectory $\{r_t\}_{t=1}^{T}$, and generates the answer y conditioned on this chain:

$$p(y \mid x) = \sum_{r_1,\ldots,r_T} p(y \mid r_{1:T}, x) \cdot \prod_{t=1}^{T} p(r_t \mid r_{<t}, x).$$

However, standard LLMs rely solely on their parametric knowledge, which limits performance in scenarios requiring up-to-date or external information.

RAG for Single-Step Reasoning. We address the task of generating a response y given an input x, enhanced by retrieval from an external corpus $\mathcal{D}$. RAG combines a retriever and a generator to condition the output on both the input and relevant documents.

A standard language model defines:

$$p(y \mid x) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, x).$$

In RAG, generation is conditioned on retrieved documents $\{d_j\}_{j=1}^{k}$, typically approximated as:

$$p(y \mid x) \approx \sum_{j=1}^{k} p(y \mid x, d_j) \cdot p(d_j \mid x).$$

The retriever encodes queries and documents via $f_q(x)$ and $f_d(d)$, scoring relevance by:

$$\text{sim}(x, d) = f_q(x)^\top f_d(d).$$

Top-k documents are retrieved, and a generator (e.g., BART [20], T5 [28]) produces y based on both x and d_i.

Retrieval in In-the-Wild Settings. We use the term "in-the-wild" to refer to open-domain corpora like Wikipedia that are not tailored for specific tasks or domains. Unlike curated corpora, they require retrieving logically relevant evidence from a large, diverse, and often tangential pool of content, making retrieval and reasoning more challenging.

	GPQA	SuperGPQA			MMLU		
	Overall	science-hard	science-middle	engineering-hard	(Pro) Chemistry	(Pro) Biology	(STEM) College Chemistry
Direct							
CoT	42.42	4.52	15.08	6.53	<u>25.44</u>	51.88	<u>49.50</u>
Direct+RAG							
CoT+RAG	45.96	<u>7.54</u>	12.56	7.54	25.18	54.39	43.00
Decomposed							
Least-to-Most	44.95	6.03	14.57	<u>10.05</u>	24.56	53.97	45.40
Step-Back	44.44	5.03	15.08	6.03	22.70	56.49	43.00
Decomposed+RAG							
Least-to-Most+RAG	45.95	6.03	14.57	8.04	22.97	<u>58.02</u>	46.00
Step-Back+RAG	43.43	5.53	<u>15.58</u>	9.05	23.06	56.34	43.00
HyDE	<u>46.46</u>	<u>7.54</u>	13.07	7.04	22.97	57.88	49.00
Ours							
RAISE	**51.01** (+9.8%)	**10.05** (+33.3%)	**19.60** (+25.8%)	**10.55** (+5.0%)	**28.36** (+11.5%)	**59.27** (+2.2%)	**51.00** (+3.0%)

Table 1: Comparison of various reasoning strategies across GPQA, SuperGPQA, and MMLU. The underscore marks the best baseline, boldface the best overall, and parentheses show RAISE's gain over the top baseline. RAISE consistently outperforms other approaches for scientific reasoning benchmarks.

3 RAISE

We propose **RAISE** (Step-by-Step **R**etrieval-**A**ugmented **I**nference for **S**cientific r**E**asoning), a retrieval-augmented generation framework for scientific reasoning designed to support multi-step reasoning through fine-grained, step-aware retrieval. The method consists of three main stages: (1) Problem Decomposition, (2) Logical Query Generation, (3) Logical Retreival. The overview of RAISE is provided in Figure 1 and Algorithm 1 in Appendix A.

Problem Decomposition. RAISE decomposes the problem x into subquestions $r_1, \ldots, r_n$ with corresponding search queries $q_1, \ldots, q_n$, forming a structured sequence for step-wise retrieval, unlike conventional single-query approaches. These queries are not used directly for retrieval but rather serve as an initial query for the next stage. As a result, this stage outputs subquestion-query pairs $\{(r_i, q_i)\}_{i=1}^{n}$, forming the basis for step-wise retrieval and generation.

Logical Query Generation. In the second stage, each initial search query q_i and its corresponding subquestion r_i are jointly used to generate a logically enriched **logical query** q_i^*. Since initial queries q_i lacks reasoning context and subquestions r_i alone can be noisy or overly specific, neither q_i nor r_i alone is sufficient for effective retrieval. By combining both, we generate logical queries that better capture the reasoning intent and retrieve logically relevant knowledge for solving each step. The model is prompted with both q_i and r_i, along with a reformulation prompt p_2. Even if the reformulated query q_i^* contains factual inaccuracies, it tends to retrieve passages from a corpus $\mathcal{C}$ that are logically relevant and supportive of the reasoning required for solving the original problem. Figure 4 in Appendix C.2 presents example queries generated by RAISE, Step-Back+RAG, and HyDE, illustrating RAISE's ability to generate logical queries that are well-aligned with the reasoning intent.

Logical Retrieval. External knowledge D_i is retrieved for each subquestion r_i from in-the-wild corpus $\mathcal{C}$ (e.g., Wikipedia) and used to generate the subanswer a_i. We retrieve background knowledge for each subquestion using a similarity threshold T to filter irrelevant documents. After retrieval, for each subquestion r_i, the model predicts its solution a_i using D_i, the original question x, and the previous steps. Finally, all subanswers are combined to generate the final answer y.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate on three scientific benchmarks: **GPQA**, **SuperGPQA**, **MMLU**, which cover graduate-level STEM and professional science tasks that require multi-step scientific reasoning.

Retriever and Language Models. We adopt Dense Passage Retrieval (DPR) [14] trained on the Natural Questions (NQ) dataset [17] as our retriever. For GPQA, our primary benchmark, we use Mistral Small 3.1-Instruct-2503 [25] (24B), while for SuperGPQA and MMLU we use the lighter LLaMA 3.1-8B model [4] due to computational limits.

Baselines. To assess the importance of multi-step reasoning and step-aware retrieval, we conduct experiments with four groups of baselines: **Direct Reasoning**(CoT [37]), **Direct Reasoning with RAG**(CoT+RAG [21]), **Decomposed Reasoning**(Least-to-Most [51], Step-Back [49]), and **Decomposed Reasoning with RAG**. The last group retrieves evidence for each subquestion and solves them step-by-step. This group includes Least-to-Most+RAG [21], Step-Back+RAG, and HyDE [3], with the latter two improving retrieval relevance through query reformulation, making them strong baselines. Further details about datasets, retriever, model settings, and baselines are provided in Appendix B.

4.2 Main Results

As shown in Table 1, our proposed method, RAISE, consistently outperforms all baseline reasoning strategies across three benchmark datasets of varying difficulty: GPQA, SuperGPQA, and MMLU, achieving an average performance improvement of 13% over the best baseline scores. Unlike other baselines whose performance varies depending on the dataset's difficulty or type, RAISE consistently demonstrates robust performance and outperforms them across different domains, types, and levels of difficulty. Furthermore, we confirm that these improvements hold across models of different scales, including smaller LLaMA-8B and GPT-4o mini, as shown in Appendix C.1, demonstrating that RAISE's effectiveness is not tied to a specific LLM architecture or size.

To assess the effectiveness of our logical query generation, we compare RAISE with three RAG-based decomposed reasoning baselines that differ in how they construct retrieval queries. Least-to-Most+RAG uses the subquestion itself as the query, Step-Back+RAG abstracts a general principle from the subquestion, and HyDE generates a hypothetical answer to use as the retrieval query. RAISE consistently outperforms all baselines across benchmarks, demonstrating the advantage of generating logically grounded queries that better align with the reasoning required to solve each subquestion. These results confirm that RAISE's queries go beyond retrieving documents that are merely domain-relevant or superficially similar, enabling access to knowledge that is logically aligned with the problem-solving process. Qualitative examples further support this finding, as shown in Appendix C.2, where RAISE retrieves passages containing essential scientific mechanisms while conventional RAG often returns vague or unrelated content.

Unlike RAISE, decomposed reasoning methods do not always yield better performance, particularly for smaller open-source LLMs that lack sufficient background knowledge [7, 40]. While decomposition can help structure reasoning, without access to relevant external knowledge, smaller models may produce hallucinations or unsupported intermediate steps, sometimes leading to worse performance than direct reasoning. Moreover, even when retrieval is added, naive RAG can introduce additional noise. In such cases, the retriever may surface superficially related or distracting content rather than the core principles needed for reasoning, which can ultimately harm performance. This highlights the importance of retrieving logically relevant knowledge rather than merely domain-related content, a challenge that RAISE directly addresses.

5 Analysis of RAISE

To further assess the importance of problem decomposition, we also evaluate a variant of our method that omits this step and directly performs logical query generation and retrieval without breaking the problem into subquestions, as shown in Figure 2. This version, referred to as RAISE-Direct, showed

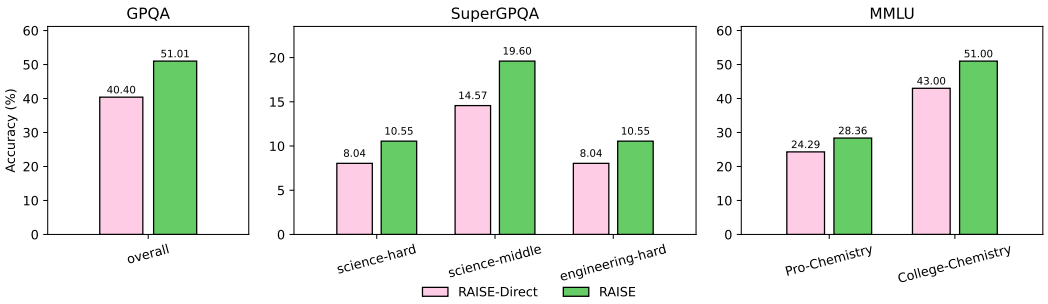

Figure 2: Performance comparison between RAISE-Direct and RAISE across datasets.

lower performance compared to the full version of RAISE. These results indicate that problem decomposition plays a critical role in guiding the retrieval process and structuring the reasoning pathway. This suggests that for complex reasoning problems, decomposing the question and retrieving logical knowledge tailored to each subquestion is more effective than retrieving once based on the original question alone. This is likely because different reasoning steps often require distinct pieces of information that may not be jointly retrievable from a single query.

We also analyze the quality of the retrieved documents. Using both an LLM-as-a-judge and a small-scale human evaluation, we find that RAISE consistently retrieves fewer irrelevant or superficial documents and more passages that directly support reasoning (Appendix C.3). These results confirm that RAISE's gains stem from retrieving logically aligned knowledge rather than merely domain-related content.

6 Conclusion

We introduce RAISE, a step-by-step retrieval framework for scientific reasoning. We first decompose the problem into multiple subquestions and search queries, and then generate logical queries and retrieve logically relevant documents from in-the-wild corpus. We demonstrate the effectiveness of RAISE on three scientific reasoning benchmarks by comparing with various baselines. Our analysis shows that RAISE retrieves documents that are not only relevant in terms of the domain (e.g. definition of specificalized terminology) but also logically relevant documents for each subquestion, assisting the step-by-step reasoning process required for scientific reasoning. Although our experiments focus on scientific reasoning, the stepwise logical retrieval in RAISE is broadly applicable to other domains such as mathematical problem solving, which also demands precise multi-step inference.

Acknowledgments

This work is in part supported by the National Research Foundation of Korea (NRF, RS-2024-00451435(20%), RS-2024-00413957(20%)), Institute of Information & communications Technology Planning & Evaluation (IITP, RS-2021-II212068(10%), RS-2025-02305453(15%), RS-2025-02273157(15%), RS-2025-25442149(10%) RS-2021-II211343(10%)) grant funded by the Ministry of Science and ICT (MSIT), Institute of New Media and Communications(INMAC), and the BK21 FOUR program of the Education, Artificial Intelligence Graduate School Program (Seoul National University), and Research Program for Future ICT Pioneers, Seoul National University in 2025.

References

[1] Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. Hallulens: Llm hallucination benchmark. *arXiv preprint arXiv:2504.17550*, 2025.

[2] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[3] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, 2023.

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

[5] Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. Deeprag: Thinking to retrieval step by step for large language models, 2025. URL `https://arxiv.org/abs/2502.01142`.

[6] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL `https://arxiv.org/abs/2009.03300`.

[7] Arian Hosseini, Alessandro Sordoni, Daniel Toyama, Aaron Courville, and Rishabh Agarwal. Not all llm reasoners are created equal. *arXiv preprint arXiv:2410.01748*, 2024.

[8] Gautier Izacard and Édouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, 2021.

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

[10] Zhihuan Jiang, Zhen Yang, Jinhao Chen, Zhengxiao Du, Weihan Wang, Bin Xu, and Jie Tang. Visscience: An extensive benchmark for evaluating k12 educational multi-modal scientific reasoning. *arXiv preprint arXiv:2409.13730*, 2024.

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

[12] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1830–1842, 2024.

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[14] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

[15] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[16] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL `https://arxiv.org/abs/2205.11916`.

[17] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[18] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

[19] Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401, 2024.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

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

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

[23] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. *arXiv preprint arXiv:2502.12442*, 2025.

[24] Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, Aixin Sun, Hany Awadalla, et al. Sciagent: Tool-augmented language models for scientific reasoning. *arXiv preprint arXiv:2402.11451*, 2024.

[25] Mistral AI. Mistral Small 3.1. `https://mistral.ai/news/mistral-small-3-1`, 2025.

[26] OpenAI. Hello gpt-4o, 2024. URL `https://openai.com/index/hello-gpt-4o/`.

[27] Vignesh Prabhakar, Md Amirul Islam, Adam Atanas, Yao-Ting Wang, Joah Han, Aastha Jhunjhunwala, Rucha Apte, Robert Clark, Kang Xu, Zihan Wang, and Kai Liu. Omniscience: A domain-specialized llm for scientific reasoning and discovery, 2025. URL `https://arxiv.org/abs/2503.17604`.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[29] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[30] Alice Rueda, Mohammed S. Hassan, Argyrios Perivolaris, Bazen G. Teferra, Reza Samavi, Sirisha Rambhatla, Yuqi Wu, Yanbo Zhang, Bo Cao, Divya Sharma, and Sridhar Krishnan Venkat Bhat. Understanding llm scientific reasoning through promptings and model's explanation on the answers, 2025. URL `https://arxiv.org/abs/2505.01482`.

[31] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science, 2022. URL `https://arxiv.org/abs/2211.09085`.

[32] M-AP Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *CoRR*, 2025.

[33] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, 2023.

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

[35] Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models, 2024. URL https://arxiv.org/abs/2307.10635.

[36] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

[37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022.

[38] Geemi P Wellawatte, Huixuan Guo, Magdalena Lederbauer, Anna Borisova, Matthew Hart, Marta Brucka, and Philippe Schwaller. Chemlit-qa: a human evaluated dataset for chemistry rag tasks. *Machine Learning: Science and Technology*, 6(2):020601, 2025.

[39] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine, 2024. URL https://arxiv.org/abs/2402.13178.

[40] Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, Shuohang Wang, Weijian Xu, Jianfeng Gao, and Weizhu Chen. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math, 2025. URL https://arxiv.org/abs/2504.21233.

[41] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[42] Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning. *arXiv preprint arXiv:2401.07950*, 2024.

[43] Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024.

[44] Ningning Zhang, Chi Zhang, Zhizhong Tan, Xingxing Yang, Weiping Deng, and Wenyong Wang. Credible plan-driven rag method for multi-hop question answering. *arXiv preprint arXiv:2504.16787*, 2025.

[45] Qiang Zhang, Keyang Ding, Tianwen Lyv, Xinda Wang, Qingyu Yin, Yiwen Zhang, Jing Yu, Yuhao Wang, Xiaotong Li, Zhuoyi Xiang, Kehua Feng, Xiang Zhuang, Zeyuan Wang, Ming Qin, Mengyao Zhang, Jinlu Zhang, Jiyu Cui, Tao Huang, Pengju Yan, Renjun Xu, Hongyang Chen, Xiaolin Li, Xiaohui Fan, Huabin Xing, and Huajun Chen. Scientific large language models: A survey on biological & chemical domains, 2024. URL https://arxiv.org/abs/2401.14656.

[46] Xinyu Zhang, Yuxuan Dong, Yanrui Wu, Jiaxing Huang, Chengyou Jia, Basura Fernando, Mike Zheng Shou, Lingling Zhang, and Jun Liu. Physreason: A comprehensive benchmark towards physics-based reasoning, 2025. URL https://arxiv.org/abs/2502.12054.

[47] Zilong Zhao, Yao Rong, Dongyang Guo, Emek Gözlüklü, Emir Gülboy, and Enkelejda Kasneci. Stepwise self-consistent mathematical reasoning with large language models. *arXiv preprint arXiv:2402.17786*, 2024.

[48] Zilong Zhao, Yao Rong, Dongyang Guo, Emek Gözlüklü, Emir Gülboy, and Enkelejda Kasneci. Stepwise self-consistent mathematical reasoning with large language models. *arXiv preprint arXiv:2402.17786*, 2024.

[49] Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3bq3jsvcQ1.

[50] Xianrui Zhong, Bowen Jin, Siru Ouyang, Yanzhen Shen, Qiao Jin, Yin Fang, Zhiyong Lu, and Jiawei Han. Benchmarking retrieval-augmented generation for chemistry, 2025. URL https://arxiv.org/abs/2505.07671.

[51] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

A RAISE Algorithm

Algorithm 1: RAISE Inference Procedure

Input: Original question x, prompts $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$, corpus $\mathcal{C}$
Output: Final answer y
Step 1: Problem Decomposition
Generate subquestions and initial queries:
$\{(r_i, q_i)\}_{i=1}^{n} \sim P_\theta(\cdot \mid x, p_1)$
for $i = 1$ **to** n **do**
  Step 2: Logical Query Generation
  Reformulate initial query:
  $q_i^* \sim P_\theta(\cdot \mid r_i, q_i, p_2)$
  Step 3: Knowledge Retrieval
  Retrieve top-k documents:
  $D_i = \mathcal{R}(q_i^*, \mathcal{C}, k)$
  Step 4: Subquestion Answering
  if $i = 1$ **then**
    $a_i \sim P_\theta(\cdot \mid x, r_1, D_1, p_3)$
  else
    $a_i \sim P_\theta(\cdot \mid x, \{(r_j, a_j)\}_{j=1}^{i-1}, r_i, D_i, p_3)$
  end
end
Step 5: Final Answer Composition
Generate final answer using all subanswers:
$y \sim P_\theta(\cdot \mid x, \{(r_i, a_i)\}_{i=1}^{n}, p_4)$

B Experiment Details

B.1 Dataset Details

GPQA [29] This dataset consists of physics, biology, and chemistry questions written by domain experts. We use GPQA diamond subset, which consist of 198 high-quality questions selected based on human performance. Specifically, this subset includes questions that both experts answer correctly while the majority of non-experts fail to solve. Each question typically demands multi-step reasoning, precise formula manipulation, and access to external scientific facts (e.g., physical constants, definitions). Due to its alignment with our target setting, GPQA serves as the primary evaluation benchmark throughout our experiments.

For GPQA, the original dataset does not include standardized multiple-choice labeled as (A), (B), (C) and (D). To ensure consistency during evaluation, we proprocessed each question by randomly shuffling the correct answer along with the three distractors, and assigning them uniformly to choice labels (A) through (D).

SuperGPQA [32] SuperGPQA is a large-scale benchmark designed to evaluate graduate-level reasoning across 13 disciplines, 72 fields, and 285 graduate-level disciplines. In alignment with the scientific reasoning focus of our work, we select science and engineering domains for evaluation. Each domain is further divided by three difficulty levels(easy, medium, and hard). To reduce computational overhead while maintaining consistency, we randomly sample 199 questions per subset using a fixed seed (42). Specifically, our experiments include 199 examples each from science-hard, science-middle, and engineering-hard subsets.

MMLU [6, 36] The MMLU benchmark covers a wide range of subjects across multiple domains. For out experiments, we focus on **STEM** and **Pro**fessional categories. The STEM contains university-level science and engineering subjects such as college mathematics and computer science, while the Professional category covers specialized fields that typically require professional training or advanced education, including law, medicine, and chemistry. We specifically select three subsets: college

chemistry from MMLU-STEM and professional chemistry and biology from MMLU-Pro. These subsets are chosen to evaluate our method's ability to perform scientific reasoning in both academic and professional contexts involving complex domain knowledge.

B.2 Baseline Details

CoT [37, 16] We apply Chain-of-Thought prompting for direct reasoning, where the model is encouraged to explicitly generate intermediate reasoning steps through prompting (Think step by step).

CoT+RAG [21] We implement CoT+RAG by combining Chain-of-Thought prompting with retrieval, where the model is prompted to solve the problem step-by-step while also leveraging external knowledge. Specifically, we provide the model with a CoT-style prompt encouraging step-by-step reasoning, alongside the original question and documents retrieved using the original question as the search query.

Least-to-Most [51] Least-to-Most is a decomposed reasoning strategy that breaks down a complex problem into a sequence of simpler subquestions, which are then solved sequentially without retrieval augmentation. This subquestion decomposition pipeline serves as the foundational structure for other decomposed reasoning methods as well.

Step-Back (Decomposed reasoning) [49] We implement Step-Back for decomposed reasoning by applying the Step-Back prompting method to each subquestion in a decomposed reasoning framework. While the original Step-Back paper does not cover the application of this method to decomposed subquestions, we extend it for a fair comparison with our approach. Specifically, after decomposing the original question into subquestions, we use the Step-Back prompting strategy to extract a high-level principle for each subquestion, and then provide the subquestion along with its corresponding principle to guide the model's reasoning.

Least-to-Most+RAG (Decomposed reasoning with RAG) [23] We implement RAG by first decomposing the original problem into subquestions and then retrieving documents using each subquestion as a query. The retrieved documents are provided to the model along with the corresponding subquestion to support its reasoning.

Step-Back+RAG (Decomposed reasoning with RAG) [49] We extend the Step-Back prompting strategy to a retrieval-augmented setting for fair comparison with our method. After decomposing the original question into subquestions, we generate a principle abstraction for each subquestion using Step-Back prompting, and use it as a query to retrieve evidence. The retrieved documents are then provided alongside the original subquestion to guide the model's reasoning.

HyDE (Decomposed reasoning with RAG) [3] We apply the HyDE approach to each subquestion in a decomposed reasoning framework. For each subquestion, the model first generates a hypothetical answer, which is then used as a query to retrieve supporting documents. The retrieved evidence, together with the subquestion, is provided to the model to support step-by-step reasoning.

B.3 Retriever Configuration

We use the pre-trained DPR encoder from the 'facebook/dpr-question_encoder-single-nq-base' model [14], which is a BERT-based encoder trained for open-domain question answering. This encoder is trained on the Natural Question (NQ) dataset [17] and is designed to map questions into 768-dimensional dense vector representations for retrieval.

For the retrieval corpus, we use the preprocessed Wikipedia passages provided by 'facebook/wiki_dpr' [14], a corpus widely used to evaluate DPR-based retrieval models. This corpus is constructed from the December 20, 2018 Wikipedia dump, where each article is split into multiple, disjoint text blocks of 100 words, resulting in approximately 21 million passages. Each passage is accompanied by the title of the wikipedia page it comes from along with DPR embedding.

To enable efficient retrieval over the passage embeddings, we use an exact FAISS index. FAISS (Facebook AI Similarity Search) [13, 2] is widely used library for fast similarity search over dense vectors.

Throughout all experiment, we retrieve top-10 documents per query. To reduce the impact of potentially irrelevant documents by DPR, we apply a similarity threshold T in RAISE. Specifically, we discard any retrieved passage whose DPR similarity score falls below T. DPR similarity is computed as the inner product between L2-normalized query and passage embeddings. Higher scores indicate greater semantic similarity, with values closer to 1 representing stronger alignment between the query and passage. We set $T = 0.84$ for GPQA, SuperGPQA, and MMLU-Pro, which are composed of more challenging reasoning problems. For MMLU-STEM (college chemistry), we use a slightly lower threshold of $T = 0.80$, considering that the questions are generally simpler than those in other datasets.

C Additional Results

C.1 Applying RAISE to various LLMs.

To assess the generalizability of RAISE across different LLM scales, we evaluate its performance on GPQA using LLaMA 3.1-8B [4] and GPT-4o mini [26], in addition to Mistral (used in our main experiments). As shown in Table 2, RAISE demonstrates consistent improvements over other baselines, exhibiting a similar trend to our main results with Mistral-24B. This shows that the effect of RAISE is not limited to a specific type of LLM, but can be applied to various LLMs with different scales.

	LLaMA	**GPT**	**Mistral**
Direct			
CoT	22.22	40.91	42.42
Direct+RAG			
CoT+RAG	23.23	40.40	45.96
Decomposed			
Least-to-Most	26.26	<u>45.45</u>	44.95
Step-Back	<u>28.28</u>	42.42	44.44
Decomposed+RAG			
Least-to-Most+RAG	24.24	42.93	45.95
Step-Back+RAG	21.72	42.42	43.43
HyDE	25.75	38.89	<u>46.46</u>
Ours			
RAISE	**30.30** (+7.1%)	**47.98** (+5.3%)	**51.01** (+9.8%)

Table 2: Evaluation on GPQA with various LLMs with different scales: LLaMA 3.1-8B, GPT-4o mini, and Mistral Small 3.1. Underscore marks the best baseline; bold indicates the best overall. Values in parentheses under RAISE show gains over the top baseline. RAISE consistently shows the best performance across all settings.

C.2 Qualitative Evaluation of Retrieved Documents

We qualitatively demonstrate the examples when RAISE retrieves logically relevant documents compared to convential RAG in Figure 3. While RAG often retrieves documents that are topically related yet fail to address the reasoning needs of the subquestion, RAISE consistently identifies documents that include essential scientific principles, mechanisms, or equations. For instance, in questions involving chemical reactions, RAISE surfaces materials that explain the specific reactivity

Subquestion	RAG	RAISE	Explanation
What is the product of the reaction of 2,8-dimethylspiro[4.5] decan-6-ol with sulfuric acid?	Carbylamine reaction ... synthesis of an isocyanide by the reaction of a primary amine, chloroform, and base.	The alkene acts as a nucleophile and attacks the proton, following Markovnikov's rule. In the second step, an HO molecule bonds to the more substituted carbon...	The RAISE-retrieved document explains the acid-catalyzed dehydration mechanism of alcohols, directly aligning with the transformation of 2,8-dimethylspiro[4.5]decan-6-ol to a ketone.
What is the concentration of OH^- ions in a solution of 0.3 M $Ba(OH)_2$?	Normality is an ambiguous measure of the concentration of a solution. It needs a definition of the equivalence factor...	Barium hydroxide is a chemical compound with the formula $Ba(OH)_2(H_2O)$. Barium hydroxide can be prepared by dissolving BaO in water... The Ba centers adopt a square anti-prismatic geometry.	The RAISE-retrieved document clearly identifies barium hydroxide as $Ba(OH)_2$ and explains its dissociation behavior in water, directly supporting the calculation of $[OH^-]$ concentration.

Figure 3: Examples where RAISE-retrieved documents provide logically relevant information for scientific reasoning compared to baseline RAG retrieval.

or the retarded time calculation, whereas RAG may return vague definitions or unrelated economic concepts. These cases illustrate how RAISE's retrieval is not only domain-aware but also aligned with the logical demands of solving complex scientific problems.

SubQuestion	Step-back+RAG	HyDE	RAISE
What is the reduced mass of the diatomic molecule XY?	Reduced Mass: The reduced mass (μ) of a two-body system is a quantity that appears in the two-body problem in physics and astronomy.	The atomic masses of X and Y are 20 u and 30 u respectively ✗ ... atomic masses of X and Y are 20 u and 30 u respectively ✗	The reduced mass of a diatomic molecule XY is calculated using the formula μ = (m1 * m2) / (m1 + m2) ...
What is the energy of the first excited state of the diatomic molecule XY?	Quantum Mechanics: The energy levels of a diatomic molecule are quantized, meaning they can only take on specific discrete values. ...	The bond length of XY is 1.2 Å, ✗ and the vibrational frequency is 500 cm^-1. The reduced mass of XY is 10 amu. ✗	The energy levels of a quantum harmonic oscillator are quantized by $E_n = (n + 1/2)\hbar\omega$, ...The first excited state corresponds to n = 1
	Provides a broad overview of concepts and principles	Includes problem-specific variables or values, which can lead to incorrect or overly narrow retrieval	Provides a **logically relevant knowlege** while avoiding distractions in the subquestions

Figure 4: Examples comparing query generation methods (Step-Back+RAG, HyDE, and RAISE) for the same subquestion. Both Step-Back+RAG and HyDE are methods that reformulate the original query to retrieve more relevant documents. These methods are included as baselines in the main comparison table.

C.3 Evaluation of Logical Relevancy of Retrieved Documents

To further investigate our hypothesis that RAISE retrieves documents that are logically more relevant compared to other baselines, we use LLM-as-a-judge (GPT-4o-mini) to evaluate the logical relevancy of the retrieved documents. Conditioned on the question, subquestion for a specific step, and the retrieved documents, the evaluator model evaluates the logical relevancy among 4 levels of logical relevancy: (1) *Not relevant at all*, (2) *Superficially relevant* (topically related but logically unhelpful), (3) *Partially logically relevant* (some useful reasoning content), and (4) *Fully logically relevant* (logically sufficient to solve the subquestion).

The results are illustrated in Figure 5. Compared to other baselines that also applies RAG, RAISE has the lowest ratio of documents that are irrelevant at all or only superficially relevant (relevant in terms of domain knowledge, but not relevant logically) and highest ratio of documents that are at

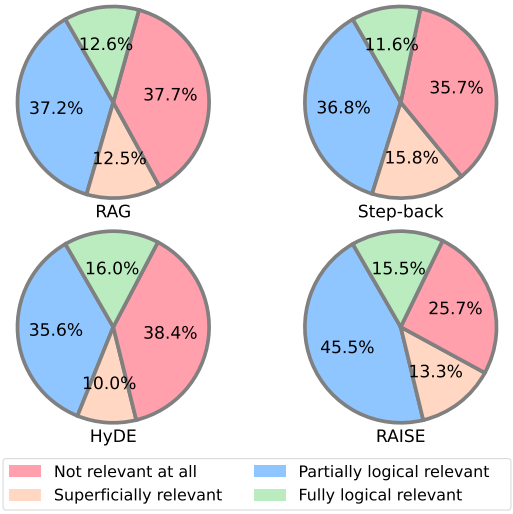

Figure 5: **Logical Relevancy of Retrieved Documents.** Unlike other baselines, RAISE has higher ratio of documents that are logically relevant and lower ratio of documents that are irrelevant or superficially relevant.

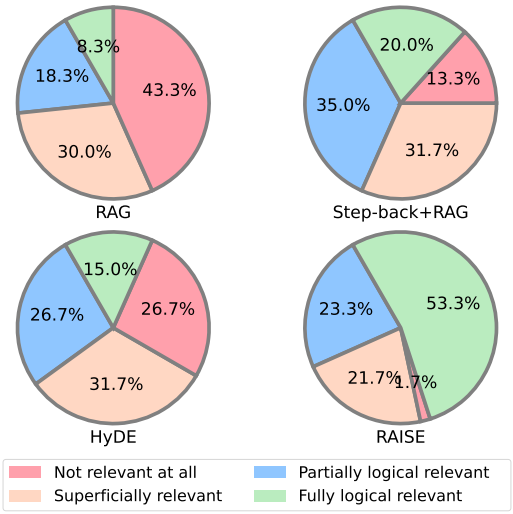

Figure 6: **Human Evaluation of the Logical Relevance of Retrieved Documents** Aligned with the results from the LLM-as-a-judge evaluation of logical relevancy, RAISE shows a higher proportion of logically relevant documents and a lower proportion of irrelevant or superficially relevant ones.

least partially logically relevant. This indicates that RAISE avoids retrieving documents that may interrupt the reasoning process for scientific reasoning through logical query generation.

Since our domain includes complex, expert-level questions, and LLM-based evaluations may overlook domain-specific reasoning and often rely on surface-level features, we supplemented our analysis with a small-scale human evaluation of 20 subquestion–document pairs. Each pair was assessed by at least three annotators, including Ph.D. students and a faculty member in chemistry, with the method provenance concealed to maintain objectivity. As also discussed in the LLM-as-a-judge results, the human evaluation indicates that RAISE produces significantly fewer irrelevant documents compared to all other methods, while achieving the highest proportion of logically relevant documents. Although limited in scale due to time and cost constraints, we believe this evaluation provides meaningful human validation of RAISE's effectiveness and serves as a valuable complement to the LLM-based assessments.

C.4 Further Analysis on GPQA

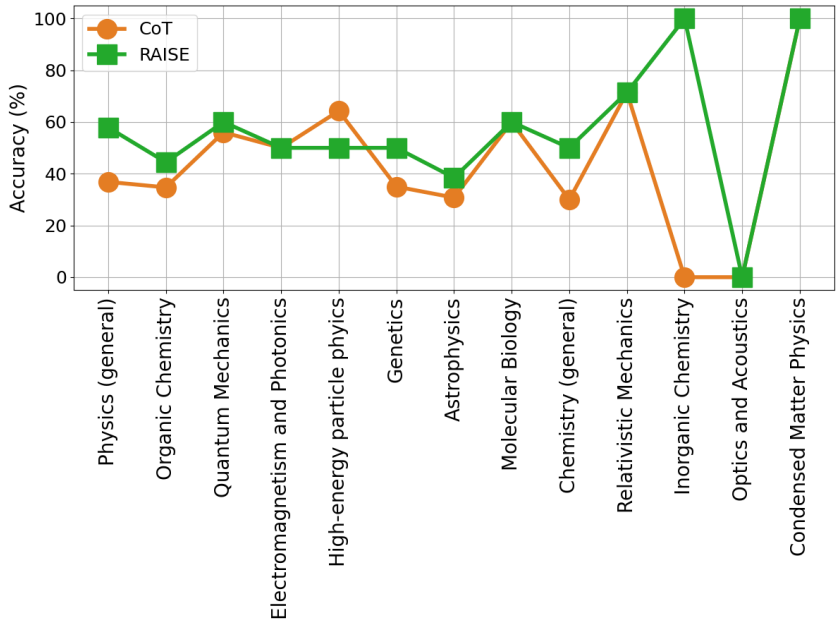

Figure 7: Domain-wise accuracy comparison between CoT and RAISE on the GPQA Diamond subset.

Figure 7 shows the domain-wise accuracy on the GPQA Diamond dataset. We compare the performance of RAISE against Chain-of-Thought (CoT) prompting across all domains. RAISE outperforms or matches CoT in nearly all domains, with only one domain where CoT shows higher accuracy. These results demonstrate RAISE's robustness and its ability to generalize across diverse areas of graduate-level scientific reasoning.

D Related Works

LLMs for Scientific Reasoning. Recent works have shown that LLMs can be applied for challenging scientific reasoning tasks. Unlike other domains, scientific reasoning requires not only step-by-step thinking, but also knowledge of specialized terminology and adaptation to continually evolving knowledge. Due to this challenging nature, many benchmarks have been proposed recently to tackle scientific reasoning with LLMs [29, 50, 46, 35, 10]. Many works enhance scientific reasoning capabilities of LLMs through domain-specific training [31, 27, 43], step-by-step reasoning [30, 29], or retrieval of external knowledge or tools [24, 50, 38, 22]. Unlike previous works, we focus on applying step-by-step document retrieval from in-the-wild corpus without assuming access to well-curated and domain-specific corpus.

Step-wise Reasoning. A growing body of research has shown that decomposing complex problems into structured intermediate steps can enhance the reasoning abilities of LLMs. An influential early approach, Chain-of-Thought prompting [37], introduced explicit, sequential reasoning steps, making the model's thought process more transparent and coherent. This inspired methods such as Plan-and-Solve [33], which emphasizes high-level planning before answering, and Step-Back Prompting [49], which encourages abstraction by prompting the model to reflect before solving. Least-to-Most prompting [51] extends this by breaking down tasks into simpler subproblems, solved in increasing order of difficulty.

While prior work has focused on prompting strategies that help LLMs better use their internal reasoning capabilities, our work addresses a complementary challenge: enabling LLMs to retrieve and apply information from in-the-wild sources like Wikipedia, particularly during step-wise problem solving. We investigate how external evidence can be integrated at each step to improve reasoning beyond what internal knowledge alone can achieve.

Retrieval Augmented Generation. Retrieval-Augmented Generation (RAG) [21] was initially proposed to improve LLMs' factual accuracy and knowledge by retrieving relevant external documents during generation [21, 14, 8, 15].

Recently, RAG has been extended for multi-hop reasoning, performing retrieval iteratively at multiple reasoning steps [23, 48, 49]. In parallel, query reformulation and expansion techniques have been developed to enhance retrieval. Instead of using the original question, models generate enriched queries through prompting, such as intermediate answers or summaries. For example, HyDE [3] and CSQE [19] demonstrate that carefully crafted queries greatly improve retrieval in complex, multi-step tasks.

Building on this line of work, we redesign query expansion techniques with the specific goal of retrieving documents that contain the key logic or underlying principles required at each step of a step-wise reasoning process. This enables the model to supplement its limited internal knowledge with external sources, leading to more complete problem solving, especially in complex, multi-step tasks.

E Prompts

E.1 Baseline Prompts

You are solving a multiple choice question. Think step by step and show your reasoning clearly.
At the end, state your answer in the format: `"The final answer is (X)"`.
Here, X must be the correct letter choice.
Question: [Problem here]
Answer Choices: [Answer choices here]
Solution:

Figure 8: Prompt for CoT

You are an expert at Science. You are given a Science problem.
Your task is to extract the Science concepts and principles involved in solving the problem.
What are the principles behind this question?
End your response with `"End of generation"` after you answer the instructions.
Question: [Subquestion here]
Principles Involved:

Figure 9: Prompt for Step-Back Principle Abstraction

You are an expert at Science. You are given a Science problem and a set of principles involved in solving the problem.
Solve the problem step by step by following the principles.
At the end, state your answer in the format: `"The final answer is (X)"`.
Here, X must be the correct letter choice.
Question: [Problem here]
Principles: [Principles here]
Answer Choices: [Answer choices here]
Solution:

Figure 10: Prompt for Step-Back

Generate a paragraph that answers the question.
End your response with `"End of generation"` after you answer the instructions.
Question: [Subquestion here]
Explanation:

Figure 11: Prompt for HyDE Query Generation

E.2 RAISE Prompts

You are given a multiple-choice question.

Break this problem into essential subquestions that directly help solve the original problem.

Each subquestion MUST also include its search query.

Each search query should reflect scientific or mathematical knowledge needed to answer the subquestion.

STRICT FORMAT REQUIREMENTS:

1. For each subquestion, you MUST provide exactly two parts in this order:

- The subquestion

- A search query for that subquestion

2. Use EXACTLY this format for each subquestion:

Subquestion 1: [your specific subquestion]

Search Query for Subquestion 1: [Write a search query someone might realistically use to learn how to answer this subquestion]

Question: [Problem here]

Answer Choices: [Answer choices here]

Figure 12: Prompt for Problem Decomposition

You are given a subquestion and a search query.

The search query is a realistic phrase that someone might use to find knowledge or reasoning support to answer the subquestion.

Your task is to anticipate what essential scientific or mathematical explanation the search result would contain, and write it concisely (2–3 sentences).

Focus only on the core concept or principle that would help answer the subquestion.

Avoid restating the subquestion, and do not include unrelated or overly general information.

Subquestion: [Subquestion resulting from Problem Decomposition]

Search Query: [Search query resulting from Problem Decomposition]

Explanation:

Figure 13: Prompt for Logical Query Generation

You are solving a multiple-choice question. The question is decomposed into several subquestions. You will be given:

1. The original multiple-choice question
2. Previous subquestions and their solutions (if any)
3. The current subquestion to solve
4. Documents that are relevant to the current subquestion

Your task:

- Carefully read the original question, any previous subquestions and their solutions, and the current subquestion.

- Use the information from the retrieved documents to solve the current subquestion.

- Also use your existing knowledge to solve the current subquestion.

- Your solution should be detailed and logically structured.

Documents: [Retrieved document]

Question: [Problem here]

Answer Choices: [Answer choices here]

Previous subquestions and their solutions:

[Previously generated subquestions and solutions]

Current subquestion to solve:

Subquestion [Step num]: [Subquestion]

Subquestion [Step num] Solution:

Figure 14: Prompt for Solving Subquestions with Documents

You are solving a multiple-choice question. The question is decomposed into several subquestions. Each subquestion has already been solved. Your task is to carefully read the original question and the several subquestion solutions, then use them to determine the final answer. Think step by step and then finish your answer with `"The final answer is (X)"` where X is the correct letter choice.

Original Question:

Question: [Problem here]

Answer Choices: [Answer choices here]

Subquestions and Solutions:

[Generated stepwise subproblems and solutions]

Final Solution:

Figure 15: Prompt for Generating Final Answer

You are given the following three items:

- Original Problem: [Problem here]

- Subquestion: [Subquestion here]

- Retrieved Document: [Document here]

Your task is to evaluate how helpful the retrieved document is for answering the subquestion.

Please follow these instructions:

- Do not just check if the topic is related.

- Instead, check if the document includes information that helps someone reason through and solve the subquestion.

- Focus on whether the document supports actual thinking or steps needed to get the answer.

Give your final judgment using only one of the following ratings:

- **"No relevance at all"** – does not have any domain similarity

- **"Superficially relevant"** – has domain similarity (only superficially) but does not have any logical relevance to the subquestion. For example, the document might mention the same topic as the subquestion, but it does not provide any information that helps solve the subquestion.

- **"Partially relevant"** – has domain similarity and has some logical relevance to the subquestion. For example, the document might provide some information that helps solve the subquestion, but it does not provide all the logical steps needed.

- **"Fully relevant"** – has domain similarity and has almost all logical relevance to the subquestion. For example, the document provides enough relevant logical steps to solve the subquestion.

Then explain your reasoning briefly.

Output Format:

Helpfulness Rating: `<one of the 4 options above>`

Explanation: `<your short explanation>`

Figure 16: Prompt for Evaluation with GPT