SGRAG: A Self-Improving Framework for Retrieval-Augmented Generation through Mento Carlo Tree Search

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) enhances large language models (LLMs) by integrating external knowledge, yet faces challenges in data quality and noise propagation. While generation-augmented paradigms mitigate these issues by synthesizing auxiliary knowledge via LLMs, existing methods rely on answer-level supervision during fine-tuning, lacking explicit optimization signals for intermediate knowledge generation. This oversight leads to error accumulation, where inaccuracies in background knowledge compromise final outputs . To address this, we propose Self-Guided RAG(SGRAG), a two-stage framework, which consists of Generation-oriented Fine-Tuning and Self-Improving Training via Ment carlo improvement. The first stage trains LLMs on retrieval corpora to establish foundational knowledge generation capabilities. The second stage employs MCTS to model decoding as a tree search process, where a reward model provides stepwise guidance, enabling iterative refinement of knowledge trajectories. By pairing these trajectories via direct preference optimization (DPO). We evaluate our proposed framework on five benchmark datasets. The experimental results demonstrate that our proposed methods consistently outperform baseline methods outperform strong baselines by a substantial margin across all datasets.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2023; Gutiérrez et al., 2024; Asai et al., 2024) enhances large language models (LLMs) for knowledge-intensive tasks like question answering by leveraging external knowledge sources. Existing studies typically retrieve textual evidence from external corpora to augment LLM inputs, offering tailored responses for long-tail queries while circumventing expensive model retraining costs (Ovadia et al., 2023). However, such retrieval-dependent approaches suffer from inherent limitations in data quality and scalability, often introducing noisy or incomplete contextual information during augmentation. Recent work thus explores generation-augmented paradigms where LLMs autonomously synthesize auxiliary knowledge, reducing external dependencies while improving output consistency.

Current generation-augmented methods primarily employ fine-tuning to elicit query-specific background knowledge. Yet due to scarce high-quality supervision, these approaches rely solely on final-task accuracy as training signals, lacking explicit optimization objectives for intermediate knowledge generation. This weak supervision leads to systematic errors models frequently produce factually flawed or irrelevant background content (Figure 1a) , propagating inaccuracies to downstream predictions and limiting overall performance gains.

To address the problem, we propose a self-improving framework for RAG. The framework enables LLMs to repeatedly generate and self-evaluate background knowledge for input queries, constructing contrastive (positive/negative) sample pairs for Direct Preference Optimization (DPO). This closed-loop system leverages the LLMâĂŹs intrinsic reward mechanism to provide targeted feedback, gradually refining its ability to produce high-fidelity knowledge.

In this framework, the generation of background knowledge is typically executed using sampling strategies or beam search, with the process unfolding on a token-by-token basis. Additionally, LLM predominantly provides an overall evaluation signal to guide the generation process. Nevertheless, the signal means that intermediate steps are not adequately guided, potentially leading to a loss of coherence and inconsistency in the final output. As shown in Figure 1(b), LLM produce errors in the intermediate stages of generating background knowledge, which ultimately result in incorrect fi-

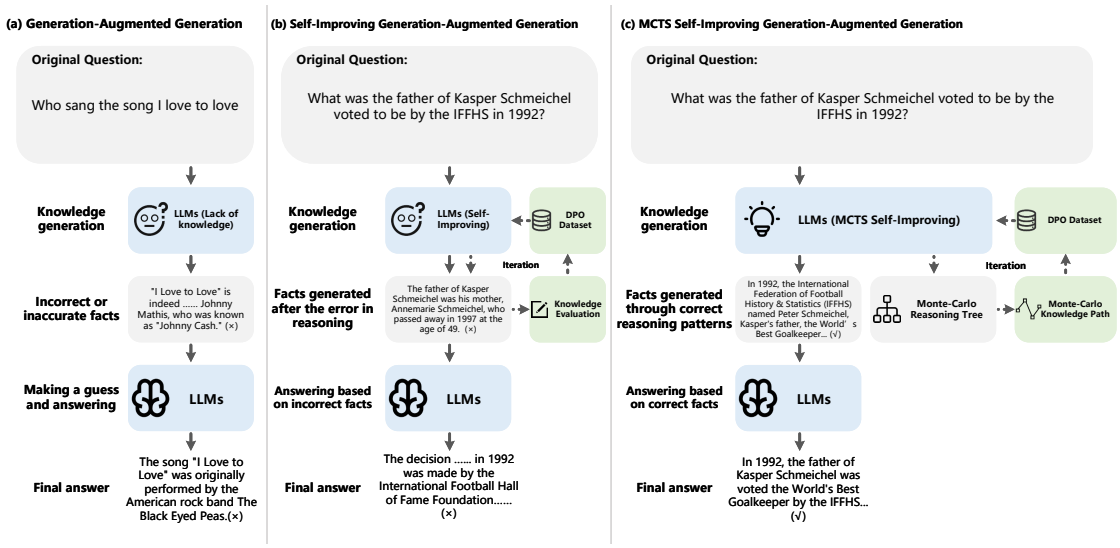

Figure 1: Comparison of Generation-Augmented Generation, Self-improving Generation-Augmented Generation and SGRAG.

nal answers.

To inject more fine-grained intermediate signals, we introduce Monte Carlo Tree Search (MCTS) into our framework. MCTS models the decoding process as a tree search, where each node represents a potential state and each edge represents a possible action. The reward model guides each step by evaluating and scoring these actions via LLM. Through iterative exploration and exploitation, MCTS refines the search tree, progressively narrowing down to the most promising trajectories.

In this study, we propose a self-improving framework for Retrieval-Augmented Generation (RAG), named SGRAG. This framework consists of two stages: Generation-Oriented Fine-tuning and Self-Improving Training via Monte Carlo Tree Search (MCTS). In the first stage, we fine-tune large language models (LLMs) on instruction-following tasks and retrieval corpora from external sources. This step equips the LLMs with essential skills for instruction-following and background knowledge generation. In the second stage, the framework includes three key modules: the knowledge-trajectory constructor, the LLM-based reward model, and the experience tool. The knowledge-trajectory constructor employs MCTS to produce multiple knowledge trajectories, which are then evaluated by the LLM-based reward model. The experience tool stores positive knowledge trajectories and constructs positive and negative example pairs. Finally, the parameters of the LLM are iteratively updated using the Direct Policy Optimization (DPO) strategy.

Our contribution can be summarized as follows:

- We propose SGRAG, a RAG framework that enhances LLM's capability for question answering on specialized domains.
- We design a novel instruction fine-tuning approach that enables LLMs to perform both question answering and question generation. This joint capability facilitates self-improvement through self-training on generated synthetic data, leading to enhanced model performance.
- We evaluate our proposed framework on five benchmark datasets, consistently outperform baseline methods outperform strong baselines by a substantial margin across all datasets..

2 Related Work

Retrieval-augmented generation. RAG (Retrieval-Augmented Generation) has emerged as a powerful tool for knowledge-intensive natural language processing (NLP) tasks, including language modeling (Borgeaud et al., 2022) and question answering (Lewis et al., 2020; Shi et al., 2024). The conventional approach involves integrating a retriever with a large language model (LLM) generator and designing a fine-tuning process to align the retriever's capabilities with those of the LLM. Recent research has sought to enhance RAG through various innovative strategies. These enhancements include the development of dynamic retrieval processes that improve the relevance of the retrieved content, as

well as methods for filtering out irrelevant contexts to strengthen RAG's robustness (Yoran et al., 2024; Yu et al., 2024, 2023; Wang et al., 2024). Furthermore, several studies have introduced instruction-tuning techniques specifically designed to enhance the search and RAG capabilities of LLMs (Liu et al., 2024; Lin et al., 2024; Dong et al., 2024; Wei et al., 2024).

Self-training. Self-training, also known as pseudo-labeling, is one of the earliest methods in semi-supervised learning (Rosenberg et al., 2005). This approach involves using a teacher model to generate new labels, which are then used to train a student model. Self-training has found widespread application across various natural language processing (NLP) tasks, including text classification (Du et al., 2021), natural language understanding (Vu et al., 2021), and ranking. Recently, the concept of self-training has been extended to instruction fine-tuning for large language models (LLMs) (Yuan et al., 2024; Li et al., 2024), reasoning tasks (Pang et al., 2024), and alignment processes (Gulcehre et al., 2023). However, to the best of our knowledge, this pipeline has not been extensively explored in the context of RAG applications. A significant drawback of self-training is its susceptibility to label noise (Arazo et al., 2020). To mitigate this issue, several strategies have been proposed to stabilize the self-training process, including sample selection techniques (Li et al., 2024) and reweighting methods (Wang et al., 2021).

3 Methodology

In this section, we will describe the details of our framework, as shown in Figure **??**. The framework consists of two stages, Generation-oriented Fine-Tuning and Self-improving training via MCTS. The first stage fine-tunes the LLM with generation-oriented data. In the second stage, the LLM updates

3.1 Stage-I: Generation-oriented Fine-Tuning

To start with, we leverage instruction fine-tuned LLMs as the backbone (e.g. `meta-llama/Meta-Llama-3-8B-Instruct`). Although these models have been instruction finetuned, they still exhibit a deficiency in generating background knowledge to answer domain-specific questions.

Instruction Fine-tuning (SFT) data. To help maintain the models' ability to comprehend and follow instructions, we leverage the SFT data including OpenAssistant (Köpf et al., 2023) and Self-Instruct (Wang et al., 2023).

Selective Background Knowledge data. To enhance the LLM's general skills of generating background knowledge, we fine-tune them on a diverse array of general domain question-answering datasets. For each sample, a query q its background knowledge $\mathcal{D}$ is given, and the LLM is trained to generate background knowledge to the query.

Note that we make sure there is no overlap between above data and test data from target tasks.

3.2 Stage-II: Self-Improving Training via MCTS

In stage II, we propose a self-improving framework to address the problem of lack immdieate signals. It consists of thhree module, MCTS-based-knowledge-synthesizer and LLM-based reward model. For each sample, we utilize mcts-based-knowledge-synthesizer to generate several background-knowledge contexts and co by LLM-based reward model.

3.2.1 Knowledge-Trajectory Constructor

We utilize MCTS method to construct knowledge trajectories. Currently, MCTS has demonstrated its potential in reasoning tasks which employs an additional reward model to better preference data at each reasoning step. MCTS operates through four main procedures: selection, expansion, evaluation, and back-propagation. Specifically, our tree construction process is as follows:

- **Selection**: We select the node to be expanded using Equation 1.

$$UCB_i = v_i + \alpha \times \sqrt{2 \times \ln\left(\frac{N_i}{n_i}\right)}, \quad (1)$$

  where n_i and N_i represent the visit count and the parent visit count of the node, respectively. α is a scalar that balances exploration and exploitation. v_i denotes the value of the node, which is derived from the average reward provided by the reward model.

- **Expansion**: For each node to be expanded, we generate several child nodes using a sampling-based algorithm.

- **Evaluation**: In terms of evaluating each node, we assess each node using the value provided by a reward model.

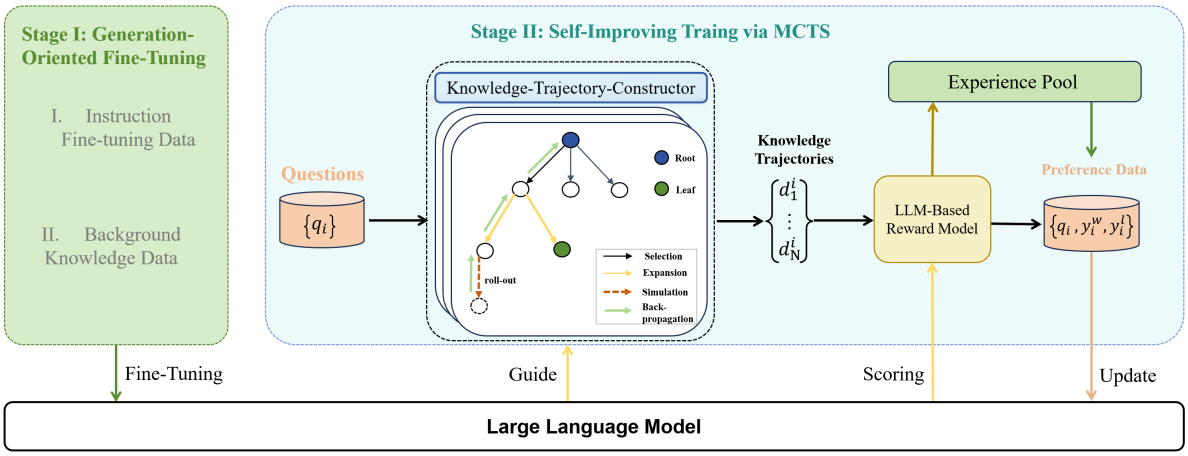

Figure 2: An overview of our framework SGRAG

- **Back-propagation**:We update the parent node by incorporating the values of the leaf nodes and adjusting the parent node's visit count accordingly..

Specifically, for a given query q, during the expansion phase, the node at layer t is represented as s_t. The new node s_{t+1} is generated using Equation 2:

$$s_{t+1} = \pi_\theta(q \oplus s_1 \oplus s_2 \oplus \cdots \oplus s_t), \quad (2)$$

where π_θ is the generator, and $\oplus$ represents the concatenation operation.

In evaluation phase, we use LLM-based Reward Model 3.2.2to calculate its value, the results is:

$$r_{s_{t+1}} = \Theta(q \oplus s_1 \oplus s_2 \oplus \cdots \oplus s_t, s_{t+1}), \quad (3)$$

where $r_{s_{t+1}}$ represents the average reward of the seven principles, and Θ is the reward model used to evaluate the reward of s_{t+1} as the suffix. Upon reaching each leaf node, the back-propagation phase is executed. During each selection phase, we utilize Equation 1 to make selections.

To be specific, we collect the chosen candidates in each preference pair with average rewards below the threshold η for refinement, as shown in Equation 4.

$$S_{\text{refinement}} = \{s_{win} \mid r_{s_{win}} \leq \eta\}, \quad (4)$$

where s_{win} and $r_{s_{win}}$ represent the chosen candidate of the collected preference pair and the corresponding average reward.

3.2.2 LLM-based Reward Model

To quantify the quality of synthesized knowledge trajectories, we utilize LLM to design two metrics as follow:

Intrinsic Perplexity Metric:

$$\mathcal{P}(D) = \exp\left(-\frac{1}{|D|}\sum_{x_i \in D} \log P_\theta(x_i|x_{<i})\right) \quad (5)$$

where D represents the knowledge trajectory and $|D|$ its token length.

Extrinsic LLM-as-Judge:

$$S_{\text{judge}} = \mathop{\mathbb{E}}_{b \sim p_\theta(b|q)} \left[\text{LLM}_{\text{judge}}(b \parallel q, \mathcal{D})\right] \quad (6)$$

where $S_{\text{judge}} \in \{1, 2, 3, 4, 5\}$.

3.2.3 Experience Pool

Due to the potential for synthesizing low-quality data during the Self-improving process, we propose and maintain an iterable experience pool to store positive experiences from this process and optimize the preferred dataset in each iteration. Specifically, the iterable global experience pool stores the optimized experience for each sample up to the current iteration. Its update and utilization process comprise two phases: 1) Evaluation, where we introduce a reasoning knowledge evaluation module based on the rewarding model in the Self-improving method. This module evaluates and compares the reasoning knowledge combinations obtained iteratively with those in the experience pool to update the global experience pool; 2) Training, where the updated global experience pool serves as the optimized experience set for the current iteration, which we use to train the LLM.

3.3 Training Objective

We leverage the positive samples from experience pool and negative samples to construct preference

pairs, and use DPO training strategy to update the parameter of LLM. The loss is as :

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(q,y_w,y_l)\sim D}\Big[\log\sigma\big(\beta\log\frac{\pi_\theta(y_w|q)}{\pi_{ref}(y_w|q)} - \beta\log\frac{\pi_\theta(y_l|q)}{\pi_{ref}(y_l|q)}\big)\Big], \quad (7)$$

where y_w and y_l is the chosen and rejected response, respectively and π_{ref} is the reference model. D is the pair-wise preference dataset, σ is the sigmoid function, and β controls the degree of deviation from the reference model.

4 Experimental Setup

4.1 Datasets and Metrics

We evaluate our model across five question-answering datasets: HotpotQA(Yang et al., 2018), 2WikiMultiHopQA (2WikiQA)(Ho et al., 2020), Natural Questions (Kwiatkowski et al., 2019), PopQA (Rabinovich et al., 2023), and WebQuestion (Wang et al., 2014). We utilize two primary metrics to evaluate the performance of our LLM's generated responses: Cover-EM (EM) and F1 score.

4.2 Baselines

To evaluate the performance, we employ the following baselines: Vanilla LLM, which is based on Llama-3.1-8B; Chain of Thought (CoT), utilizing 8-shot examples extracted from the training dataset; IterDRAG (Yue et al., 2024), which decomposes questions and answers step-by-step through in-context learning; FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024), confidence-based methods that determine retrieval timing based on token importance and uncertainty; TAARE (Zhang et al., 2024), which allows the LLM to decide when retrieval is necessary; Genread, which employs knowledge generation instead of knowledge retrieval, leveraging large language models to produce relevant knowledge based on questions; and AutoRAG (Yu et al., 2024), which uses trained models to iteratively decompose questions and retrieve relevant documents for answering.

4.3 Implementation Details

We use Llama3.1-8B (Meta-AI, 2024) and Qwen2.5-7B () as our backbone models. Due to resource constraints, we apply LoRA (Hu et al., 2022) ($r = 32, \alpha = 32$) during the fine-tuning of the Llama3.1-8B model. For both stages, we set the global batch size to 64, utilize gradient accumulation with a factor of 8, and train the model for one epoch. Specifically, the learning rate is set to $5e-7$ for Stage-I and $2e-7$ for the Llama3 backbone in Stage-II, while for the Qwen2.5 backbone in Stage-II, it is set to $5e-7$. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. All experiments are conducted on 8 NVIDIA A100 GPUs. The prompt format for answer and question generation, as well as inference, is provided in the Appendix E.

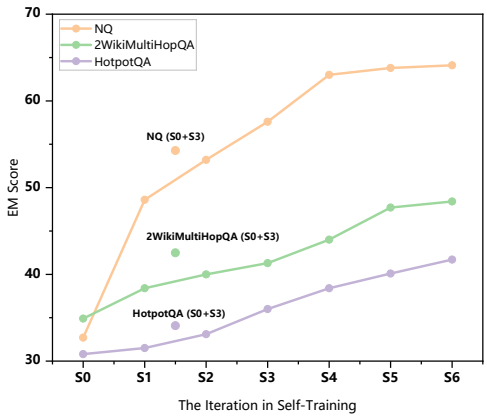

Figure 3: The iteration results in self-training

5 Experimental Results

5.1 Main Results

Table 1 present the experimental results for five benchmark datasets. From the results, we have the following findings:

(1) SGRAG consistently outperforms baselines across all six datasets , the average performance gain is 8.01%, 6.37%, 8.61% over the Llama variant and 1.19%, 3.50%, 3.20% over the Gemma variant, respectively. Our method consistently outperforms existing approaches by enabling self-improving framework. Compared to reasoning-based (i.e., CoT, CoT-Retreive and AutoRAG) and adaptive RAG baselines (i.e., UAR and TAARE), SGRAG achieves improvements across all datasets, verify the effectiveness of the generate backgroud knowledge by LLM itself through self-improving framework. Specially, GENREAD shows a poor performance on HotpotQA and 2Wiki indicates that prompt-based generation enhancement methods often struggle to produce accurate background knowledge when confronted with complex problems, such as multi-hop reasoning.

(2) SGRAG exhibits remarkable generalization

Methods	*in-distribution*			*out-of-distribution*	
	Natural Questions	HotpotQA	2WikiMultihopQA	PopQA	Web Question
Llama3.1-8B					
Genread	45.6	24.6	22.9	21.5	32.3
Auto-RAG	55.1	25.8	23.0	27.8	17.4
DRAGIN	53.4	27.6	29.1	22.6	21.2
TAARE	52.3	30.6	35.2	33.2	23.4
SGRAG	**63.4**	**42.2**	**50.3**	**44.5**	**34.3**
-w/o IEP	57.4	38.9	47.2	42.6	31.6
-w/o RM	55.8	36.8	46.2	40.2	29.5
Qwen2.5-7B					
Genread	28.0	23.0	27.1	26.4	24.6
DRAGIN	50.5	16.7	12.4	12.0	17.4
TAARE	49.5	25.3	21.3	27.0	18.2
SGRAG	**56.1**	**34.5**	**42.6**	**42.2**	**26.8**
-w/o IEP	53.9	31.2	40.2	40.0	23.3
-w/o RM	51.8	31.9	39.2	38.4	21.9

Table 1: SGRAG's main experimental results, with the best performance under each dataset **bolded** and sub-optimal performance underlined.

capabilities and robustness in out-of-distribution settings. DeepRAG achieves substantial F1 score improvements of 2.63 and 4.57 on PopQA and WebQuestions respectively, even in scenarios where relevant information may be sparse or missing from the knowledge base.

5.2 Ablation study

To evaluate the effectiveness of each model module, we perform ablation studies where we remove the Iterable Experience Pool (IEP) and Rewarding Model (RM). The results are shown in Table **??**.

After removing each module, the performance of the model deteriorated. As Table **??**, the SGRAG w/o IEP is unable to store positive experiences across iterations. Experimental results demonstrate that IEP can contribute to positive model experiences. On the other hand, SGRAG w/o RM utilizes the original reward model for data synthesis. Experimental results confirm that the reward model designed based on reasoning exploration optimizes the exploration process in the reasoning knowledge combination space.

5.3 Study on Self-Improving

5.3.1 Effect of iterations

During the Self-improving process, the quality of data varies across different iteration rounds, and the changes in data quality during these iterations determine whether the model can gain positive experience and feedback through this process. Therefore, we conducted experiments and analyses on the Self-improving process of SGRAG, with iteration rounds as the unit of analysis, and demonstrated that the preference data explored and constructed through MCTS can provide positive experience and feedback for SGRAG. The experiments were based on Llama3.1, and we divided the entire Self-improving process into seven iterative stages, including supervised fine-tuning (SFT) training, to test the performance of SGRAG at the end of each iteration.

The experimental results of the iterative process are shown in Figure 5. Across three datasets, including open-domain question answering and multi-hop question answering, the Self-improving process significantly contributed to the performance improvement of SGRAG. The experimental result curves for all three datasets exhibit a monotonically increasing trend, proving that none of the iterative processes in Self-improving introduced negative experience or feedback. This validates the value of the MCTS-based data synthesizer and iterable experience pool in SGRAG for acquiring positive experience. Compared to the open-domain question answering dataset, the performance of the

multi-hop reasoning dataset stabilized in later iteration rounds, suggesting a positive correlation between performance convergence time and the complexity of dataset reasoning. In addition to grouping experiments by iteration rounds, we also merged the third iteration (S3) with the original data (S0) to train the model after the S0 iteration, to eliminate the continuous impact of iteration rounds on model performance. The experimental results showed that the model trained in this way outperformed the model after the S0 iteration, further proving that the preference data generated in each iteration round provided a positive experience for the model.

5.4 Study on MCTS

5.4.1 Effect of Different Knowledge Generation Strategy

SGRAG explores the optimal combination space for knowledge generation through MCTS. To further validate the effectiveness of our method for exploring the combination space, we compare its performance with random sampling of the combination space on NQ, HotpotQA, and 2WikiQA. In addition to SGRAG, we construct four random sampling baselines for experimental analysis, with the number of sampling iterations as the benchmark. The LLM's temperature for the random sampling baselines is set to 1.0 to enhance the randomness of their combination space exploration, while other experimental settings remain consistent with the main experiment.

The experimental results for combination space exploration are shown in Figure 5. When the number of sampling iterations exceeds 16, the performance gains of the random sampling baselines diminish marginally, indicating that their exploration range of the combination space reaches saturation after a certain number of sampling iterations, and thus their performance ceiling is reached. In contrast, SGRAG significantly outperforms the random sampling baselines across all three datasets, suggesting that the combination space explored by SGRAG is more reasonable compared to that of the random sampling baselines. This demonstrates the effective optimization of MCTS for exploring the combination space.

5.4.2 Human evaluation on generted

To demonstrate the consistency of the reasoning knowledge combinations generated by SGRAG with human thinking patterns, we conducted a human evaluation by recruiting participants to assess these combinations across five datasets that were identical to those used in the main experiments. Specifically, we recruited 50 participants and presented them with dataset questions, gold standard answers, SGRAG-generated answers, and SGRAG-generated reasoning knowledge combinations in the form of questionnaires, without disclosing the specific evaluation standard. All participants were required to evaluate the content quality of the reasoning knowledge combinations based on the provided information and select one of the following three evaluation outcomes: 1) Factually Consistent, where the reasoning knowledge combination supports the correct answer generated by SGRAG; 2) Partially Factually Consistent, where there are facts in the reasoning knowledge combination related to the SGRAG question-answering process, but it is uncertain whether they support the correct answer generated by SGRAG; 3) Factually Inconsistent, where the reasoning knowledge combination does not support the correct answer generated by SGRAG. The final evaluation result for each sample was determined by the average rating given by the 50 participants.

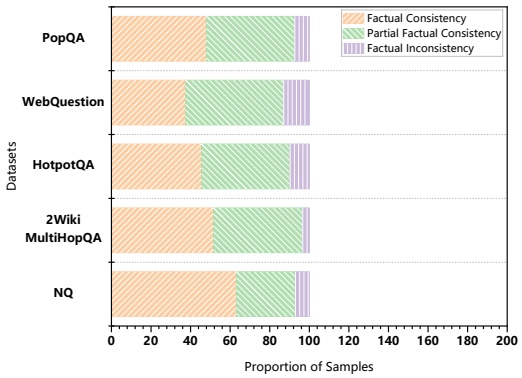

Figure 4: The statistics of factual accuracy, partial accuracy, and factual inconsistency in different datasets.

The results of the human evaluation are shown in Figure 4. Across most datasets, 50% of the reasoning knowledge samples were rated as Factually Consistent, indicating that SGRAG generated reasoning knowledge combinations that align with human thinking patterns in nearly half of the cases, and these combinations could support the correct answers generated by SGRAG. Additionally, 90% of the reasoning knowledge samples across all datasets were rated as having factual relevance, suggesting that SGRAG generated reasoning knowledge combinations related to the

question-answering process in most cases. These proportions were slightly lower in the WebQuestion dataset, which we attribute to the fact that, as an open-domain question-answering dataset, SGRAG had access to less relevant knowledge. However, compared to other baselines, SGRAG still achieved better performance in the WebQuestion dataset.

5.5 Case Studies

To better illustrate the quality of pseudo-labeled samples generated by SGRAG after Stage-I fine-tuning, we present two case studies in Table **??**, comparing the samples produced by SGRAG with those from the baseline model, Llama3-8B-it.

In the first case, where the model is asked to generate a claim supported by the context, Llama3-8B-it simply selects a sentence from the context. This results in relatively simple QA pairs, making the task less challenging for Stage-II training.

In the second case, the model is tasked with generating an answer first, and then formulating a question based on the context and the answer. While Llama3-8B-it does not copy a sentence exactly, it generates a lengthy question that closely paraphrases the context. This makes the question overly dependent on the original text, making it difficult to interpret without it. Additionally, the model misinterprets the context by implying that the research was focused on children when actually adults are the focus. In contrast, after fine-tuning on answer generation and query generation in Stage-I, SGRAG generates higher-quality QA pairs that are self-contained and understandable without relying on the context. These QA pairs also present more challenging tasks, as they require deeper comprehension of the context, providing harder and more effective training data for Stage-II.

6 Conclusion

We introduce SGRAG, an instruction fine-tuning framework designed to enhance LLMs for domain-specific question-answering tasks. By equipping LLMs with joint capabilities for both question answering and question generation, SGRAG enables the generation of diverse, high-quality synthetic questions from unlabeled domain-relevant corpora. This approach facilitates effective adaptation to specialized fields, where distribution shifts and limited domain-specific data typically pose challenges. Extensive experiments across 11 datasets in three domains show that SGRAG consistently outperforms

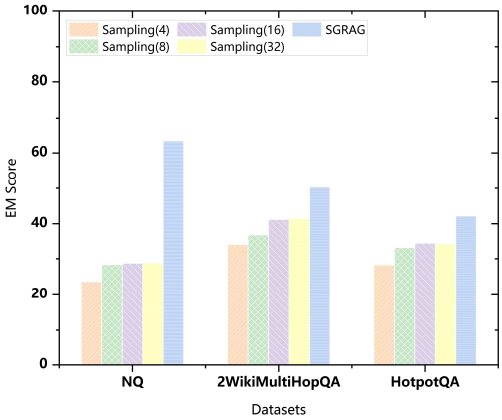

Figure 5: Experiment results on different sampling numbers.

baseline models, demonstrating its effectiveness in tackling the challenges of retrieval-augmented, domain-specific question-answering tasks.

Limitation

While SGRAG demonstrates notable improvements, there are some limitations to our approach:
Single Round Pseudo-Label Generation: Our current method relies on a single round of query generation from the corpus, which may restrict the refinement of pseudo label quality. Iterative refinement of generated synthetic queries could potentially lead to better results.
Additional Training Time: The incorporation of synthetic query generation and filtering adds time complexity compared to baseline models, which may affect efficiency in environments with limited computational resources. However, we would like to note that our method *will not increase the inference time complexity* compared to the existing RAG approaches with the same backbone models.
Stronger Query Generation Models: Although we achieved strong performance with Llama3 8B and Gemma2 27B models, leveraging more powerful query generation models, such as Llama-3.1-70B-it (Meta-AI, 2024), could yield further gains. However, using larger models would incur higher computational costs beyond our current budget.

References

Asma Ben Abacha, Eugene Agichtein, Yuval Pinter, and Dina Demner-Fushman. 2017. Overview of the medical question answering task at trec 2017 liveqa. In *TREC*, pages 1–12.

Asma Ben Abacha, Yassine Mrabet, Mark Sharp,

Travis R Goodwin, Sonya E Shooshan, and Dina Demner-Fushman. 2019. Bridging the gap between consumers medication questions and trusted answers. In *MEDINFO 2019: Health and Wellbeing e-Networks for All*, pages 25–29. IOS Press.

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

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

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Zhicheng Dou, and Ji-Rong Wen. 2024. Understand what llm needs: Dual preference alignment for retrieval-augmented generation. *arXiv preprint arXiv:2406.18676*.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv preprint*, abs/2312.10997.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *ArXiv preprint*, abs/2308.08998.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Minh Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Alexandrovich Glushkov, Arnav Varma Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Julian Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

Patrick S. H. 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2024. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *ICLR*.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. In *NeurIPS*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Meta-AI. 2024. Llama 3 model card.

OpenAI. 2022. Introducing ChatGPT.

OpenAI. 2023. GPT-4.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *ArXiv preprint*, abs/2312.05934.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *CHIL*.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. In *NeurIPS*.

Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby-Tavor. 2023. Predicting question-answering performance of large language models through semantic consistency. *arXiv preprint arXiv:2311.01152*.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)*.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust LLM specialists for evidence-based question-answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1931, Bangkok, Thailand. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Replug: Retrieval-augmented black-box language models. In *NAACL*.

Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, et al. 2024. Cs-bench: A comprehensive benchmark for large language models towards computer science mastery. *ArXiv preprint*, abs/2406.08587.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv preprint*, abs/2408.00118.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*.

Tu Vu, Minh-Thang Luong, Quoc Le, Grady Simon, and Mohit Iyyer. 2021. STraTA: Self-training with task augmentation for better few-shot learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5715–5731, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Wadden, Kejian Shi, Jacob Morrison, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, Shannon Zejiang Shen, et al. 2024. Sciriff: A resource to enhance language model instruction-following over scientific literature. *ArXiv preprint*, abs/2406.07835.

Haoyu Wang, Tuo Zhao, and Jing Gao. 2024. Blendfilter: Advancing retrieval-augmented large language models via query generation blending and knowledge filtering. In *EMNLP*.

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021. Meta self-training for few-shot neural sequence labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1737–1747.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *ACL*.

Zhenghao Wang, Shengquan Yan, Huaming Wang, and Xuedong Huang. 2014. An overview of

microsoft deep qa system on stanford webquestions benchmark. *2018-09-15]. https://www.microsoft. com/en-us/research/publication/an-overview-of-microsoft-deep-qa-system-on-stanford-webquestionsbenchmark*.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation with explicit denoising. *ArXiv preprint*, abs/2406.13629.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *ICLR*.

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

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *NeurIPS*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *ICML*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. RAFT: Adapting language model to domain specific RAG. In *COLM*.

A Training Data Details

We include the training dataset, the number of examples used in each stage, as well as the instruction format in Table 2.

B Test Data Details

We evaluate on 11 datasets in total from the medical, scientific and computer science domain. (1) **Medical:**

- MMLU-med (Hendrycks et al., 2021) is a subset of six tasks related to biomedicine, including anatomy, clinical knowledge, professional medicine, human genetics, college medicine, and college biology. It contains 1089 questions in total.
- MedMCQA (Pal et al., 2022) includes multiple-choice questions derived from Indian medical entrance exams, covering 2400 healthcare topics across 21 medical subjects. We use the 4,183-question development set from MedMCQA, as the test set lacks provided ground truths.
- MedQA (Jin et al., 2021) is collected from the US Medical Licensing Examination, containing 1273 four-option multiple-choice questions focused on real-world scenarios from professional medical board exams.
- BioASQ (Tsatsaronis et al., 2015) includes 618 questions constructed from biomedical literature without providing the ground truth snippets, challenging RAG systems to infer answers independently.
- PubMedQA (Jin et al., 2019) is a biomedical research QA dataset consisting of 1000 manually annotated questions based on PubMed abstracts. Answers in PubMedQA are structured as yes/no/-maybe to reflect the validity of the questions.
- LiveQA (Abacha et al., 2017) and MedicationQA (Abacha et al., 2019) are two QA datasets focusing on answering consumer health questions about medications, including 100 and 674 question-answer pairs, respectively.

(2) **Scientific:**

- SciQ (Welbl et al., 2017) is a scientific question-answering dataset containing 13,679 crowd-sourced science exam questions about Physics, Chemistry, and Biology, among others.

Table 2: The blending ratio of different datasets with their specific prompt format in Stage-I and Stage-II fine-tuning. For Stage-II Pseudo-labeled QA Samples, the two numbers represent the # sample for the Llama and Gemma backbones, respectively.

Dataset	Specific Instruction	Stage-I # Samples	Stage-I Blending Ratio	Stage-II # Samples	Stage-II Blending Ratio
Instruction Fine-tuning					
ChatQA SFT Data	—	60000	0.18	128000	0.12
Question Answering					
DROP	Answer the following question with a short span.	12000	0.034	29195	0.04
NarrativeQA		12000	0.034	40000	0.04
Quoref		4800	0.014	10996	0.015
ROPES		4800	0.014	10924	0.015
Squad1.1		16000	0.045	40000	0.035
Squad2.0		16000	0.045	52474	0.05
OpenbookQA	Answer the following question by selecting one of the provided options with A, B, C, or D. Please answer with the capitalized alphabet only, without adding any extra phrase or period.	2000	0.006	82092	0.005
LogiQA		4000	0.012	7376	0.006
NQ	Answer the following question with a short phrase.	16000	0.045	46426	0.04
TatQA-arithmetic	Answer the following question with a number from context or the math arithmetic using +,-,*, or /.	8325	0.045	24975	0.034
TatQA-others	Answer the following question with a short span, or a full and complete answer.	3176	0.023	9528	0.013
WebGLM	Please give a full and complete answer for the question using only the provided search results (some of which might be irrelevant) and cite them properly. Use an unbiased and journalistic tone. When citing several search results, use [1][2][3].	12000	0.034	43579	0.023
StrategyQA	Answer the following question with Yes or No.	1526	0.005	4578	0.006
BoolQ		4000	0.012	9427	0.013
FaVIQ	Answer the following question with Yes or No. Is the statement {claim} correct?	2000	0.006	10906	0.01
FEVER		2000	0.006	10444	0.01
Pseudo-labeled Question Answering					
Short-span QA	Answer the following question with a short span.	—	—	150,000 / 45,000	0.2625
Multiple-choice QA	Answer the following question by selecting one of the provided options with A, B, C, or D. Please answer with the capitalized alphabet only, without adding any extra phrase or period.	—	—	50,000 / 15,000	0.0875
Claim Verification	Answer the following question with Yes or No. Is the statement {claim} correct?	—	—	100,000 / 30,000	0.175
Answer Generaion					
Squad1.1	Based on the context, generate candidate spans within the passage that are likely to be answers to a question. Separate different candidate answers with a semicolon (';').	18877	0.063	—	—
Squad2.0		18863	0.059	—	—
DROP		4984	0.023	—	—
WebQuestions		1084	0.012	—	—
Query Generaion					
NQ	Based on the context, please generate a question. The answer to the question should be {answer}.	20000	0.068	—	—
Squad1.1		20000	0.068	—	—
StrategyQA		131	0.023	—	—
WebQuestions		24000	0.068	—	—
FaVIQ	Based on the context, please generate a claim that can be supported/refuted by the context.	10000	0.028	—	—
FEVER		10000	0.028	—	—

- ARC-easy/challenge (Clark et al., 2018) contains 7,787 authentic multiple-choice science questions at the grade-school level, designed to foster advanced question-answering research. The dataset is divided into a Challenge Set, with questions that stumped both a retrieval-based and a word co-occurrence algorithm, and an Easy Set.
- MMLU-Sci (Hendrycks et al., 2021) is the Massive Multitask Language Understanding dataset, designed to test a wide range of language understanding abilities across 57 tasks. In this work, we select 14 subjects to ensure the evaluation is not limited to certain fields.

(3) **Computer Science:**

- CS-Bench (Song et al., 2024) is a recently-proposed benchmark specifically designed to as-

sess the performance of large language models (LLMs) in computer science. It contains around 5,000 carefully selected test samples that span 26 subfields within four major areas of computer science, covering various task forms and divisions of knowledge and reasoning.

C Baseline Descriptions

- Self-RAG (Asai et al., 2024) utilizes instruction fine-tuning to adaptively retrieve passages based on the question and determine if the passage contains useful information for answering the question.
- ChatQA (Liu et al., 2024) is a fine-tuning pipeline tailored for RAG and conversational QA tasks via aggregating multiple QA and dialogue datasets.
- RAFT (Zhang et al., 2024) is a domain-specific fine-tuning approach that incorporates top-k passages as context during fine-tuning, helping to address discrepancies between training and testing data.
- EvidenceRAG (Schimanski et al., 2024) leverage off-the-shelf LLMs (GPT-4) to generate context-aware question answering datasets, which is then used to fine-tune the student model.

D Additional Experimental Results

We list the per-task results of MMLU-sci in Table 3.

E Prompt Details

E.1 Answer Generation

```
[System]

[Context]

Based on the context, generate
several candidate spans within
the passage that are likely to be
 answers to a question. The
answers can be entities, verbs or
 even numbers. Make sure that the
 answers are different and
diverse. Separate different
candidate answers with a
semicolon (';').
```

E.2 Query Generation

```
[System]

[Context]

Based on the context, please
generate a question that is
relevant to the information
provided. The question should
stand alone and not refer back to
 the context explicitly. The
question should be clear and
understandable without needing
the context. The answer to the
question should be [Answer].
```

E.3 Inference

```
[System]

[Top 10 Contexts]

[Specific Instruction]

[Question]
```

The [Specifc Instruction] for each evaluation dataset depends on their question type and can refer to those in Table 2.

Table 3: Results of our proposed method and baselines in the scientific domain.

Models	**astronomy**	**college biology**	**college chemistry**	**college physics**	**computer security**	**high school geography**	**high school macroeconomics**	**high school microeconomics**	**high school psychology**	**high school US history**	**high school world history**	**human sexuality**	**nutrition**	**virology**	**Avg.**
Metrics	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	ACC	—
Proprietary LLMs, For Reference Only															
GPT-3.5 (OpenAI, 2022)	66.45	65.28	35.00	46.53	65.00	77.27	91.54	64.29	83.12	78.43	72.15	70.99	66.01	47.59	66.40
GPT-4 (OpenAI, 2023)	93.42	93.75	61.00	73.27	91.00	94.95	97.95	94.54	96.15	95.59	94.51	93.13	89.22	56.02	87.46
Scientific LLMs															
SciTulu 7B (Wadden et al., 2024)	69.74	63.89	31.00	18.63	62.00	70.20	56.58	57.08	77.43	53.06	57.38	65.65	54.90	45.78	55.95
SciTulu 70B (Wadden et al., 2024)	83.55	80.56	36.00	28.43	83.00	89.39	80.26	79.83	91.19	77.55	77.22	78.63	68.95	50.60	71.80
Retrieval-Augmented LLMs															
Self-RAG 13B (Asai et al., 2024)	55.26	58.33	24.00	21.57	60.00	61.11	32.89	45.49	67.89	58.67	58.23	53.44	43.79	40.96	48.69
ChatQA 8B (Liu et al., 2024)	60.53	54.17	29.00	33.33	70.00	64.65	51.32	58.37	74.86	49.49	54.85	59.54	57.19	45.18	54.46
ChatQA 70B (Liu et al., 2024)	82.89	79.17	46.00	48.04	83.00	84.85	80.26	84.98	91.74	**86.73**	82.28	74.05	77.78	51.20	75.21
Backbone: Llama3-8B-Instruct															
Llama3-8B-it (Meta-AI, 2024)	78.29	71.53	38.00	40.20	83.00	82.32	63.16	72.96	84.04	65.31	72.15	69.47	70.26	49.40	67.15
RAFT 8B (Zhang et al., 2024)	80.26	75.69	37.00	42.16	84.00	79.80	65.79	74.68	83.67	72.45	77.22	73.28	71.24	51.81	69.22
EvidenceRAG 8B (Schimanski et al., 2024)	77.63	78.47	44.00	45.10	85.00	84.85	72.37	74.68	86.24	74.49	79.32	74.05	74.84	51.20	71.59
SGRAG 8B	85.53	81.94	47.00	50.98	**88.00**	89.90	76.32	84.55	92.66	83.16	81.43	84.73	**81.37**	54.82	77.31
w/o Stage II	84.87	81.25	**49.00**	49.02	87.00	88.89	73.68	82.83	90.64	80.61	81.01	83.21	79.41	51.81	75.95
Backbone: Gemma2-27B-Instruct															
Gemma2-27B-it (Team et al., 2024)	82.89	84.03	47.00	55.88	84.00	89.39	77.63	81.12	91.93	80.61	84.81	81.68	72.22	52.40	76.11
RAFT 27B (Zhang et al., 2024)	84.87	88.89	47.00	63.73	86.00	90.91	**86.84**	84.55	93.58	81.12	85.65	81.68	76.47	51.81	78.79
EvidenceRAG 27B (Schimanski et al., 2024)	84.87	87.50	**49.00**	60.78	86.00	91.41	**86.84**	85.41	93.94	81.63	86.08	81.68	76.80	51.81	78.84
SGRAG 27B	**90.13**	**91.67**	**49.00**	**68.63**	87.00	**92.42**	85.53	**87.98**	**95.05**	84.18	**86.92**	**85.50**	78.43	**55.42**	**81.28**
w/o Stage II	84.21	87.50	**49.00**	59.80	84.00	89.90	84.21	82.83	93.58	83.16	86.50	81.68	76.14	54.82	78.38