Improving Retrieval Augmented Language Model with Self-Reasoning

Yuan Xia[1], **Jingbo Zhou**[2,*], **Zhenhui Shi**[1], **Jun Chen**[1], **Haifeng Huang**[1]

[1]Baidu Inc., China
[2]Baidu Research, China
{xiayuan,zhoujingbo,shizhenhui,chenjun22,huanghaifeng}@baidu.com

Abstract

The Retrieval-Augmented Language Model (RALM) has demonstrated remarkable performance on knowledge-intensive tasks by integrating external knowledge during inference, which mitigates the factual hallucinations inherited in large language models (LLMs). Despite these advancements, challenges persist in the implementation of RALMs, particularly in terms of reliability and traceability. Specifically, the irrelevant document retrieval may result in unhelpful responses or even deteriorate the performance of LLMs, while the lack of appropriate citations in outputs complicates efforts to verify the trustworthiness of the models. To this end, we propose a novel self-reasoning framework aimed at improving the reliability and traceability of RALMs, whose core idea is to leverage reasoning trajectories generated by the LLM itself. The framework involves constructing self-reasoning trajectories through three processes: a relevance-aware process, an evidence-aware selective process, and a trajectory analysis process. We evaluated our framework across four public datasets (two short-form QA datasets, one long-form QA dataset, and one fact verification dataset) to demonstrate its superiority. Our method can outperform existing state-of-the-art models and achieve performance comparable with GPT-4, using only 2,000 training samples.

Introduction

The Retrieval-Augmented Language Model (RALM), also known as Retrieval-Augmented Generation (RAG), has become a crucial enhancement for Large Language Models (LLMs) by integrating external knowledge during inference. Despite their advanced capabilities in language understanding and generation (Brown et al. 2020; Touvron et al. 2023), LLMs are prone to producing hallucinated and inaccurate content, especially in knowledge-intensive tasks (Ji et al. 2023). Augmenting LLMs with relevant information obtained from external sources like Wikipedia and search engines has proven effective in reducing these inaccuracies (Guu et al. 2020; Lewis et al. 2020; Borgeaud et al. 2022; Izacard et al. 2022; Asai et al. 2024). This approach has proven effective in mitigating the factual hallucinations that are inherent in LLMs (Kwiatkowski et al. 2019; Petroni et al. 2021; Ram et al. 2023).

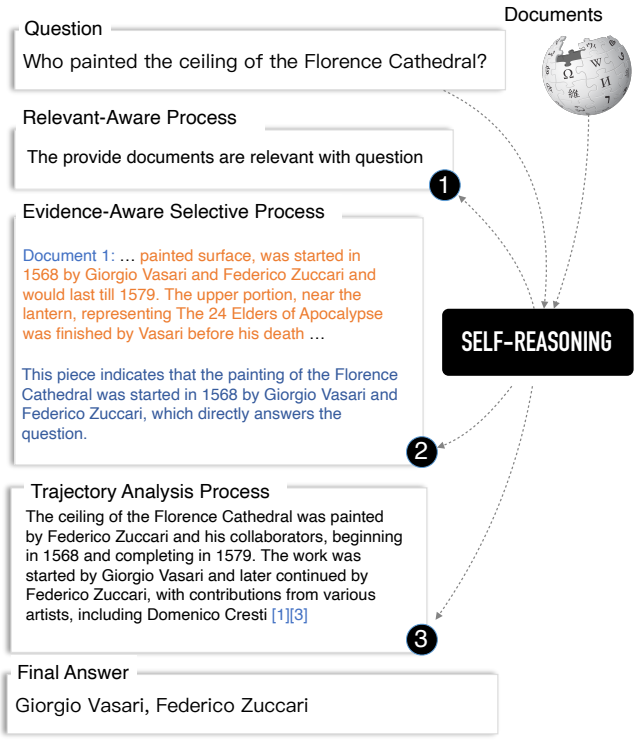

Figure 1: An example of how SELF-REASONING framework generates reasoning trajectories.

Nevertheless, there are still limitations associated with RALMs, particularly concerning reliability and traceability. Firstly, the reliability of the retrieved information remains a substantial concern. Previous studies have shown that noisy retrieval can adversely affect the performance of an LLM (Menick et al. 2022; Li et al. 2023), as irrelevant data can lead to misguided responses and disturb the model's ability to leverage its intrinsic knowledge effectively. Secondly, the interpretability and traceability of outputs generated by RALMs need to be improved. Although RALMs incorporate retrieved documents during both the training and inference

*Jingbo Zhou is the corresponding author.

phases, they may fail to explicitly cite these documents, thus complicating the process of tracing and verifying the claims made by LLMs. To improve the retrieval robustness, recent studies have explored incorporating external tools such as natural language inference (NLI) models (Honovich et al. 2022) and document summarization models during inference (Yoran et al. 2023; Xu et al. 2024). However, the effectiveness of these external tools largely influences the overall performance of RALMs. Additionally, training and optimizing these auxiliary models require additional costs. Consequently, identifying the most appropriate training and selection methods for NLI and summarization models remains a critical challenge in leveraging these approaches.

To address the above limitations, we propose a novel end-to-end SELF-REASONING framework to improve the performance of RALMs. For convenience, we will also refer to this framework as SELF-REASONING RAG and use the terms interchangeably. Our intuition is that the explicit self-reasoning trajectory crafted by LLMs can improve both the retrieval robustness and accuracy in question answering. During the pre-training phase, while an LLM primarily focuses on knowledge acquisition, it does not learn to reason from retrieved documents to generate answers. To address this, a feasible approach is to incorporate reasoning trajectories into a post-training phase. Such an approach could potentially teach the model to reason and distinguish relevant and irrelevant documents, thereby enhancing its query response accuracy. An example of how our SELF-REASONING framework generates reasoning trajectories is illustrated in Figure 1. In contrast, as shown in the middle part of Figure 2, the conventional RALM methods gather all documents in a non-selective manner, leading to the distraction of the LLM by irrelevant content and consequently resulting in the generation of erroneous answers.

Our framework constructs self-reasoning trajectories comprising three processes: 1) a *Relevance-Aware Process* (RAP), which instructs the LLM to judge the relevance between the retrieved documents and the question, 2) an *Evidence-Aware Selective Process* (EAP), which directs the LLM to choose and cite relevant documents, and then automatically select snippets of key sentences as *evidence* from the cited documents, 3) a *Trajectory Analysis Process* (TAP), which requires the LLM to synthesize a concise analysis based on all gathered self-reasoning trajectories generated by previous two processes and subsequently provide the final inferred answer. Furthermore, we propose a gradual training method by employing stage-wise masking strategies to enhance the performance of our framework. We summarize our contributions as follows:

- We propose a novel end-to-end SELF-REASONING framework that improves the robustness of RALMs by leveraging reasoning trajectories generated by the LLM itself, without the need for external tools.
- We carefully design three processes to enhance the interpretability and traceability of RALMs by requiring LLMs to explicitly generate snippets and citations from documents, and further explain the reason why cited documents can help answer the question.
- We evaluate our framework on four public datasets (two short-form QA, one long-form QA, and one fact verification), demonstrating that our method surpasses existing state-of-the-art models in performance using only 2,000 training samples.

Related Work

Retrieval-augmented LMs

Many studies have investigated augmenting the performance of LLMs with externally retrieved information (Izacard et al. 2022; Guu et al. 2020; Borgeaud et al. 2022) and some of them pre-train language models with retrieved passages. For works focusing on RALMs with citations, Menick et al. (2022); Nakano et al. (2021) instruct or train an LLM to answer questions with retrieved documents while providing citations. Gao et al. (2023b) proposes an end-to-end system to retrieve supporting evidence and generate answers with citations, while only focusing on prompting without updating their model weights. Other works instruct or fine-tune LLMs to use external tools to retrieve dynamically (Schick et al. 2023; Yao et al. 2023; Jiang et al. 2023), which offers an adaptive method of when and what to search. Gao et al. (2023a) improves the attribution and factuality of language models by taking outputs of LLMs and applying a post-process retrieve-and-edit approach.

Robustness for RALMs

To improve the robustness of RALMs, previous works can be divided into two categories. The first category utilizes retrieved documents to enhance the Chain of Thought (CoT). For example, IRCoT (Trivedi et al. 2023) iteratively uses retrieved documents to generate CoT, which is then used to retrieve further documents in subsequent steps. ReAct (Yao et al. 2023) introduces an iterative CoT paradigm that integrates reasoning with search results. However, irrelevant retrievals may produce misguided CoT, adversely affecting LLM performance (Menick et al. 2022; Li et al. 2023).

To address the issue of irrelevant retrieval information, the second category proposes using external modules to process retrieved documents during inference. For instance, Yoran et al. (2023) utilize a natural language inference model to filter out irrelevant documents, Yan et al. (2024) employ a retrieval evaluator to classify documents based on their quality, and Xu et al. (2024) and Yu et al. (2023) apply models to filter out or compress retrieved documents. Baek et al. (2023) deploy a separate small language model as a verifier to detect and correct errors in LLMs during retrieval. A method presented by Asai et al. (2024), which appears most similar to our approach, develops a technique that instructs models to retrieve information using specifically designed reflection tokens. However, this approach needs to train extra critic models and generator models to predict the reflection tokens, which requires tens of thousands of extra training samples.

Unlike the second group of works, which rely on external tools or additional modules to eliminate irrelevant information, the SELF-REASONING RAG method integrates self-reasoning directly into the model's architecture, thereby en-

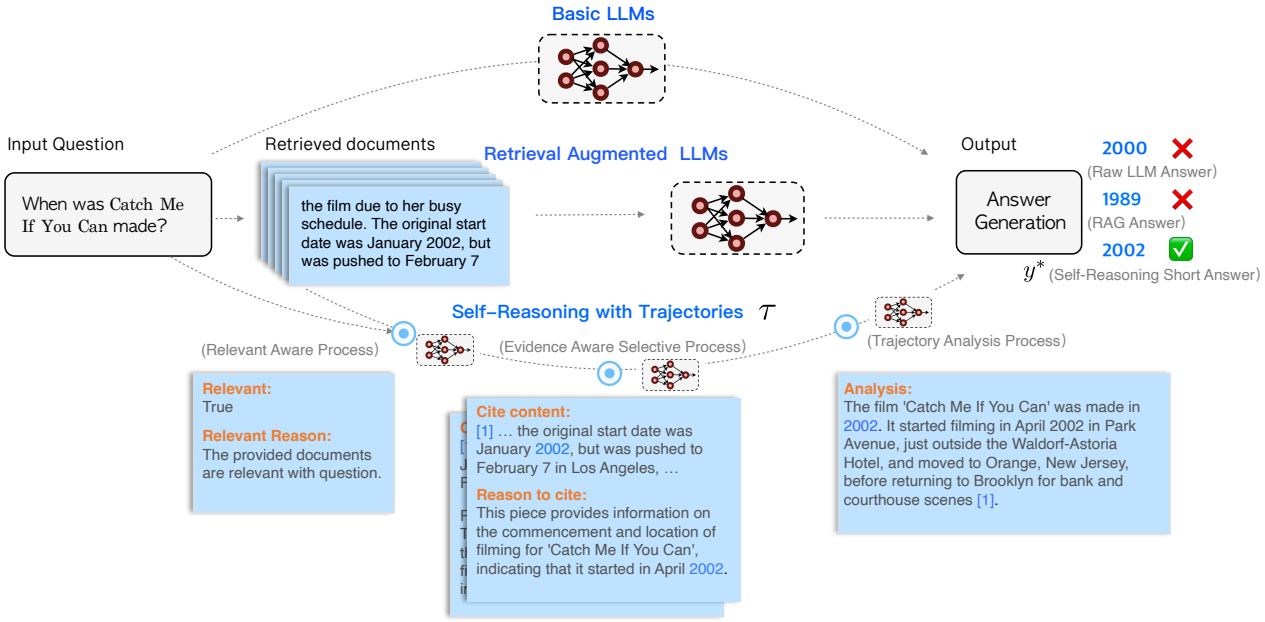

Figure 2: An illustration of the SELF-REASONING framework. The upper is the basic LLMs which answer the question by inherent knowledge. The middle is the standard retrieval augmented LMs, which use retrieved documents to help answer the question. The bottom is our SELF-REASONING framework which uses self-generated reason trajectories to output answers.

hancing the performance of LLMs and providing a more efficient and scalable solution. Further related works on LLMs for reasoning are discussed in the Appendix.

Preliminary

We formally define the problem of retrieval augmented generation with self-reasoning. Given a query q and a corpus of documents $\mathcal{D}$, an LLM-generated answer with m statements and n tokens can be defined as $y = (s_1, s_2, \cdots, s_m) = (w_1, w_2, \cdots, w_n)$, where s_i is the i-th statement and w_j is the j-th token in the generated answer. In addition, for long-form QA settings, each statement s_i should cite a list of documents $C_i = \{c_i^{(1)}, c_i^{(2)}, ...\}$, where $c_i^{(k)} \in \mathcal{D}$. In our work, we train an LLM (e.g. LLaMA2) to first generate reasoning trajectories τ through self-reasoning and then to generate answers y^* (short-form answers) on condition of τ. The model output is $y = \text{concat}(\tau, y^*)$, which is the concatenation of τ and y^*. Note that the generations of τ and y^* are done in a single pass within the SELF-REASONING framework.

Method

Here we provide a detailed implementation of the self-reasoning process which involves three processes: 1) a *Relevance-Aware Process* (RAP), 2) an *Evidence-Aware Selective Process* (EAP), and 3) a *Trajectory Analysis Process* (TAP). An illustration of our SELF-REASONING framework is shown in Figure 2. Additionally, we outline the process of data generation and quality control, and present the specifics of model training.

Relevance-Aware Process

In this work, we choose DPR (Karpukhin et al. 2020) and Contriever (Izacard et al. 2021) as default retrievers R to recall the top-k relevant documents. When presented with a question and a set of documents, people can determine whether the question is relevant to the retrieved documents. Therefore, we first instruct the model to judge the relevance between the retrieved documents $\mathcal{D}$ and the given question q. We further request the model to explicitly generate reasons explaining why given documents are identified as relevant. The output should include two fields as *relevant* and *relevant reason*, as depicted in Figure 2. If all of the retrieved documents are irrelevant, the model should provide an answer based on the internal knowledge acquired during its pre-training phase. We define the self-reasoning trajectories generated by RAP as τ_r.

Evidence-Aware Selective Process

When answering a question, people generally first identify the crucial sentences from the provided documents and then cite or highlight them as key points. This process of citing the document facilitates reading comprehension and can serve as a technique for combining multiple short answers to address various aspects. While people may carry out this selective process and citation instantaneously, LLMs need to formulate the self-reasoning trajectories explicitly.

In our work, we require the LLM to explicitly state the reason why the selected sentence is supportive and plausible in answering the question. We define the selected sentence as *evidence* in our paper. Specifically, after retrieving the top-k documents, the self-reasoning method for *Evidence-*

Aware Selective Process can be formulated as follows: First, we instruct the LLM to choose relevant documents and automatically select snippets of key sentences for the selected documents. Then, we request the LLM to output the reason why the selected snippets can answer the question. The intermediate output is a list containing multiple contents, each content should include two fields, as *cite content* and *reason for cite*, which is illustrated in Figure 2. We define the self-reasoning trajectories generated by EAP as τ_e.

Trajectory Analysis Process

Finally, we consolidate all the self-reasoning trajectories (τ_r and τ_e) in the previous processes together to form a chain of reasoning snippets, thereby enhancing the overall performance of the retrieval augmentation generation. Specifically, we ask the LLM to analyze the reasoning trajectories within itself and ultimately to output a concise analysis and a short answer. We instruct the LLM to output content with two fields as *analysis* and *answer*, which is shown in Figure 2. We define the self-reasoning trajectories generated by TAP as τ_a. In this work, the *analysis* output is defined as a long-form answer, and the *answer* output is defined as a short-form answer. In the experiment section, we further explored the performance of long-form and short-form QA settings.

Data Generation and Quality Control

Training Data Generation. For the *Relevance-Aware Process* data generation, as manually labeling the relevant and irrelevant documents is label-intensive, we request GPT-4 (OpenAI 2023) to generate answers as ground truth. Specifically, we instruct GPT-4 to generate labels regarding irrelevant fields, and further to output the reasons why the given documents cannot answer the question. We concatenate the given question and the retrieved documents as positive samples. For negative samples, we randomly select a different question from the training set and retrieve the top-k documents related to it. These documents are then concatenated with the initial question to form negative samples. To avoid order bias in the training data, we shuffle the order of the documents.

For the EAP and TAP data generation, manually annotating the citation and writing the self-reasoning process for each question is not feasible in practice. Therefore, we follow a similar process to RAP, we first instruct GPT-4 to generate a snippet of selected documents and subsequently output the reasoning process as trajectories. The method for constructing the EAP training data is the same as RAP except that the instructions given to GPT-4 are different. The details of the instructions are shown in the Appendix.

Data Quality Control. For training data generation, correct and comprehensive reasoning trajectories are very important. When training an LLM, the quality of the training samples is more important than the quantity (Zhou et al. 2023). As we cannot guarantee the correctness of self-reasoning trajectories and citations by GPT-4, we develop two efficient methods to control the quality of data generation: 1) The first method is to use the off-the-shelf tools Gao et al. (2023b) to automatically verify the performance of data generation for document citations. We calculate the citation precision and recall score for each training sample and filter out scores lower than our pre-defined thresholds δ_p and δ_r, for citation precision and recall, respectively. 2) Second, though the validation of self-reasoning trajectories and citations generated by GPT-4 is challenging, verifying the correctness of the final answer is straightforward. Therefore, we filter out the trajectories that lead to the incorrect answers and only keep the correct ones. We totally generate 10,000 training samples by GPT-4, after the filtering strategy by quality control, we finally keep 2,000 training samples with high quality. More details and pseudo-codes can be found in the Appendix.

Model Training

We train the self-reasoning RAG model ϕ by our constructed corpus which is augmented with self-reasoning trajectories τ using the standard language modeling objective, maximizing likelihood:

$$\max_{\phi} \mathbb{E}_{(q,\tau,y)\sim\mathcal{D}_{sr}} \log p_\phi(y \mid \tau, q) p_\phi(\tau \mid q) \tag{1}$$

where $\tau = \tau_r \oplus \tau_e \oplus \tau_a$ are the self-reasoning trajectories, $\oplus$ is a concatenation operator, τ_r, τ_e, τ_a are trajectories generated by above three processes respectively. q is the provided question, and y is the model output, including the intermediate reason trajectories and the final answer. $\mathcal{D}_{sr}$ is the training corpus augmented with self-reasoning trajectories.

During training, we observed that it is more challenging to ensure the correctness of an LLM with 13B parameters when generating long reasoning trajectories than short ones. We hypothesize that an LLM's effective reasoning length is limited and exceeding this limit might lead to error accumulation during the inference stage. Therefore, we propose a gradual training method by employing stage-wise masking strategies to gradually learn to generate long trajectories.

Specifically, we propose a stage-wise training process while we train the LLM stage by stage. In the first stage, we mask the trajectories produced by the next two stages (EAP and TAP) and train the model with a learning rate r_a. Then in the second stage, we only mask the trajectories generated by TAP and train the model with a learning rate r_b. Finally, we concatenate the reasoning trajectories from all stages and put them into a self-reasoning LLM for end-to-end training with a learning rate r_c. Hyper-parameters for training are described in the Appendix.

Experiments

Datasets and Settings

To demonstrate the effectiveness of our proposed SELF-REASONING framework, we conduct an extensive experimental evaluation on two short-form QA datasets (NaturalQuestion (Kwiatkowski et al. 2019) and PopQA (Mallen et al. 2023)), one long-form QA dataset (ASQA (Stelmakh et al. 2022)), and one fact verification dataset (FEVER (Thorne et al. 2018)). Detailed descriptions of the datasets can be found in the Appendix. We explore off-the-shelf retrievers. We use DPR (Karpukhin et al. 2020) and

Models	NaturalQuestion	PopQA	FEVER	ASQA		
	(acc)	(acc)	(acc)	(em-recall)	(precision)	(recall)
Baselines without retrieval						
$\text{LLaMA2}_{\text{7B}}$	19.2	18.4	23.2	10.2	-	-
$\text{LLaMA2}_{\text{13B}}$	24.0	22.6	25.3	15.3	-	-
$\text{LLaMA2}_{\text{7B-chat}}$	20.2	21.5	26.5	16.3	-	-
$\text{LLaMA2}_{\text{13B-chat}}$	23.2	25.9	28.4	18.3	-	-
Baselines with retrieval						
$\text{LLaMA2}_{\text{7B}}$	27.8	47.8	39.8	28.5	13.6	9.59
$\text{LLaMA2}_{\text{13B}}$	34.0	48.1	35.2	26.8	21.8	16.3
$\text{LLaMA2}_{\text{7B-chat}}$	27.4	52.9	43.4	25.3	34.5	33.2
$\text{LLaMA2}_{\text{13B-chat}}$	32.7	53.5	53.4	26.4	39.4	38.4
$\text{Vicuna}_{\text{7B}}$ (Chiang et al. 2023)	28.0	55.2	62.4	24.3	45.7	40.8
$\text{Vicuna}_{\text{13B}}$ (Chiang et al. 2023)	35.4	56.1	60.6	27.3	51.3	50.2
$\text{LLaMA2-FT}_{\text{7B}}$	36.8	54.4	67.5	28.5	47.2	45.4
ReAct (Yao et al. 2023)	-	-	64.6	-	-	-
RECOMP (Xu et al. 2024)	38.4	-	-	-	-	-
$\text{Self-RAG}_{\text{7B}}$ (Asai et al. 2024)	37.2	54.9	70.2	30.0	66.9	67.8
$\text{Self-RAG}_{\text{13B}}$ (Asai et al. 2024)	38.8	55.8	72.1	31.7	70.3	71.3
$\text{SELF-REASONING}_{\text{7B}}$	38.0	54.2	78.6	33.9	66.3	70.8
$\text{SELF-REASONING}_{\text{13B}}$	**41.4**	**57.3**	**83.9**	**35.2**	**71.2**	**72.3**
GPT-4	**46.6**	**62.5**	**87.7**	**41.3**	**75.6**	68.5

Table 1: Performance comparisons with different baseline models on two short-form QA datasets, a long-form QA dataset, and a fact verification dataset. The numbers with bold black represent the best results excluding GPT-4. The results are averaged over five runs, and presented with standard variance values omitted (all $\leq 2\%$).

Contriever-MS MARCO (Izacard et al. 2021) to retrieve the top five documents from Wikipedia.

By default, we use DPR as a retriever for the NQ, as DPR has been fine-tuned on the high-quality NQ data. On the PopQA, where question and answer pairs are created based on Wikipedia in 2022, therefore, for the PopQA, we use the December 2020 preprocessed Wikipedia corpus provided by (Izacard et al. 2022) and use Contriever as a retriever. For the ASQA dataset, we use GTR (Ni et al. 2022) as a retrieval that corresponds to the experimental settings in (Gao et al. 2023b). More settings can be found in the Appendix.

Evaluation Metrics

We use different evaluation metrics for short-form QA, long-form QA, and fact verification tasks.

Short-form QA metrics. We report *accuracy* for short-form QA tasks, which is based on whether ground-truth answers are included in the model predictions instead of strictly requiring exact matching, following Mallen et al. (2023); Schick et al. (2023).

Long-form QA metrics. For long-form QA tasks, we report the *EM recall* as a correctness metric, and the *citation recall* and the *citation precision* for citation quality, which are the same as the metrics in (Gao et al. 2023b).

Fact verification metrics. For the fact verification task, we report the *accuracy* as a metric, which is a three-class classification accuracy, following Thorne et al. (2018).

Baseline Models

Baseline models without retrieval. We evaluate strong open-source pre-trained LLMs as baseline models. For basic LLMs, we test LLaMA2-7B, LLaMA2-13B (Touvron et al. 2023) and its instruction-tuned chat version LLaMA2-Chat-7B, LLaMA2-Chat-13B.

Baseline models with retrieval. First, we benchmark the models using the LLaMA2 and the Vicuna (Chiang et al. 2023) series models for baselines. Additionally, for a fair comparison, we also include LLaMA2-FT, where LLaMA2 is fine-tuned on all the training samples generated by GPT-4 except the self-reasoning trajectories. To establish strong baselines, we compare our method against RECOMP (Xu et al. 2024), ReAct (Yao et al. 2023), and Self-RAG (Asai et al. 2024), all of which are trained with extra GPT-4 generated samples or external tools. We also compare our framework with GPT-4 (OpenAI 2023). We include categorical comparisons with the baseline models in the Appendix.

Main Results

Table 1 shows the performance comparisons with different methods on the four public datasets. For short-form QA evaluations, the performance of LLMs with augmented retrieval is consistently better than that of basic ones, affirming the effectiveness of the augmented approach. Notably, under the same order of magnitude parameters, our SELF-REASONING framework outperforms most of the strong baseline LLMs. Specifically, compared to the Self-RAG, our framework is an end-to-end system trained with only 2,000

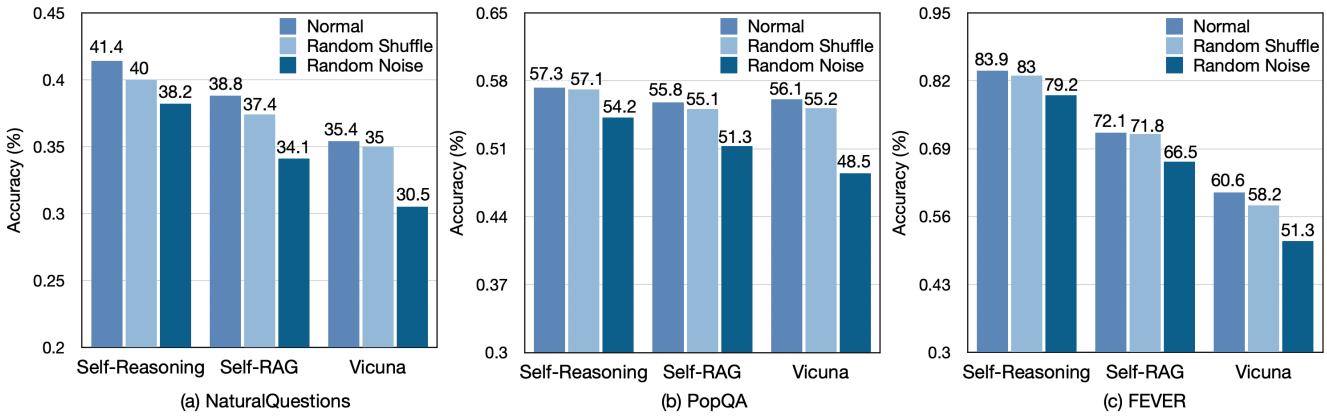

Figure 3: Noise robustness experiment results on three different datasets: (a) On the left is the NQ dataset, (b) in the middle is the PopQA dataset, and (c) on the right is the FEVER dataset. The Self-RAG and Vicuna are 13B parameter size models.

self-reasoning trajectory samples. In contrast, the Self-RAG requires training additional critic LMs to predict reflection tokens using an additional 46,000 instances generated by GPT-4. This efficiency not only simplifies the training process but also significantly reduces resource consumption.

In the context of long-form QA evaluations, for the metrics of *EM recall*, it needs to comprehend multiple documents and merge answers. The EAP and TAP are specifically designed for multi-document reading comprehension, enabling our performance to surpass other baselines. In terms of citation evaluation metrics, the SELF-REASONING RAG can achieve better results than GPT-4 in ASQA citation recall metrics (72.3 vs. 68.5). This is largely due to the reasoning trajectories generated in the EAP, which can enhance the recall and precision of citation evaluation, leading to more interpretable and traceable generations.

For fact verification evaluations, we observed that SELF-SEASONING is dominantly superior to all baseline models. Our method achieves a much higher accuracy rate than the Self-RAG model (83.9 vs. 72.1). The RAP in our framework is designed to judge the relevance between the retrieved documents and the question, which leads to a notable enhancement in accuracy for this fact verification task.

To clearly demonstrate the practical applications and benefits of our SELF-REASONING framework, we provide a case study for a more in-depth analysis in Appendix, which illustrates how our framework operates in real-world scenarios.

Models	NQ	PopQA	FEVER
	(acc)	(acc)	(acc)
ORIGIN	41.4	57.3	83.9
w/o (RAP)	39.9	54.3	72.2
w/o (EAP)	37.2	53.2	78.4
w/o (TAP)	38.2	53.4	81.2
w/o (GL)	39.5	55.3	81.2
w/o (QC)	37.7	54.2	80.8

Table 2: The ablation study on two short-form QA datasets and a fact verification dataset with 13B parameter size models. In the table, the ORIGIN represents our self-reasoning model enhanced with self-generated trajectories.

Models	NQ	PopQA	FEVER
	(acc)	(acc)	(acc)
LLaMA2	32.7	53.5	53.4
+ *trajectory*	38.3	54.2	79.2
Vicuna	35.4	56.1	60.6
+ *trajectory*	38.5	56.4	79.6

Table 3: The analysis on the effectiveness of self-reasoning trajectories with 13B parameter size models. In the table, the *+trajectory* indicates the result of the baseline model is enhanced with self-generated trajectories by our framework.

Analysis

Ablation Study

We conduct an ablation study on two short-form QA datasets and a fact verification dataset to analyze the individual contributions of each process within our proposed SELF-REASONING framework. We further explore the effectiveness of the gradual learning (GL) method and the quality control (QC) of data generation (a detailed analysis described in the Appendix). The main ablation study results are shown in Table 2 and Table 3.

Effectiveness of RAP. First, we evaluate the effect of the RAP. The removal of the RAP causes the overall performance to drop in two short-form QA datasets and a fact verification dataset, suggesting that preliminary consideration of the relevance between questions and retrieved documents can help improve performance. We notice that the performance declines most significantly in the FEVER dataset. Detecting irrelevant documents is critical in the fact-verification task. Our model will immediately output *NotEnoughInfo* if it detects that all documents are irrelevant.

Effectiveness of EAP. Then we evaluate the effect of the EAP. Removing the EAP causes the overall performance of the average *accuracy* to decline from 60.9 to 56.3 in three short-form QA datasets, which indicates that snippets of key sentences and document citations generated through self-reasoning are instrumental in boosting accuracy.

Effectiveness of TAP. Finally, we evaluate the effect of the TAP. When excluding the TAP, we can observe a performance decline on all three datasets, demonstrating that self-analysis based on two previous processes generated trajectories can also improve the performance of LLMs. Note that the *analysis* content generated by TAP is indispensable for the long-form QA evaluation.

Effectiveness of Self-Reasoning Trajectory. To verify whether the trajectories generated by the self-reasoning framework are truly effective, we put the trajectories generated by our SELF-REASONING framework into the original baseline models as input prompts, and then use the baseline models to regenerate the answers. We observe that incorporating self-generated trajectories can significantly enhance performance in short QA tasks and fact verification tasks.

Retrieval Robustness Analysis

Retrievers are not perfect and past work has shown that noisy retrieval can have negative effects on the performance of LLMs (Petroni et al. 2020; Li et al. 2023). In this section, we design two kinds of settings to validate the robustness of RALMs. In the first setting, we test whether the order of the retrieved documents will affect the performance of the RALMs. Specifically, after retrieving the top-k documents using retrievals with a descending relevance score, we randomly shuffle the order of the retrieved documents and then input them to an LLM. In the second setting, we test how noisy documents impact the performance of LLMs. When retrieving the top-k documents from the given question, we randomly replace 50% of the retrieved documents with other documents sampled from a different question in the dataset.

Figure 3 shows the noise robustness experiment results on three datasets. Our SELF-REASONING framework consistently outperforms the Self-RAG and Vicuna models. We observe that random shuffling of retrieved documents has a minimal impact on the performance of RALMs. If the provided documents are supportive, it is trivial for a RALM to determine the correct answer. However, when presented with noisy documents, all models experience a decline in performance. The performance drop in our self-reasoning framework is relatively minimal, demonstrating the robustness of our method even when handling noisy documents.

Citation Analysis

As the automatic evaluation by the NLI model cannot detect partially supported citations, we discuss the analysis of citations with human evaluation in this section. Similarly to Liu, Zhang, and Liang (2023), we conduct a human evaluation on two dimensions: 1) *citation recall*: annotators are given a statement and all documents that the statement refers to and are asked to judge whether the documents fully support the given statement; 2) *citation precision*: given a statement and one of its citations, annotators are asked to validate whether the citation *fully supports*, *partially supports* or *does not support* the statement. As shown in Figure 4, the relative rankings by human evaluation align well with those from the automatic evaluation, and the human evaluation often yields a closely higher score when compared with the automatic evaluation. Details of human annotation can be found in the Appendix.

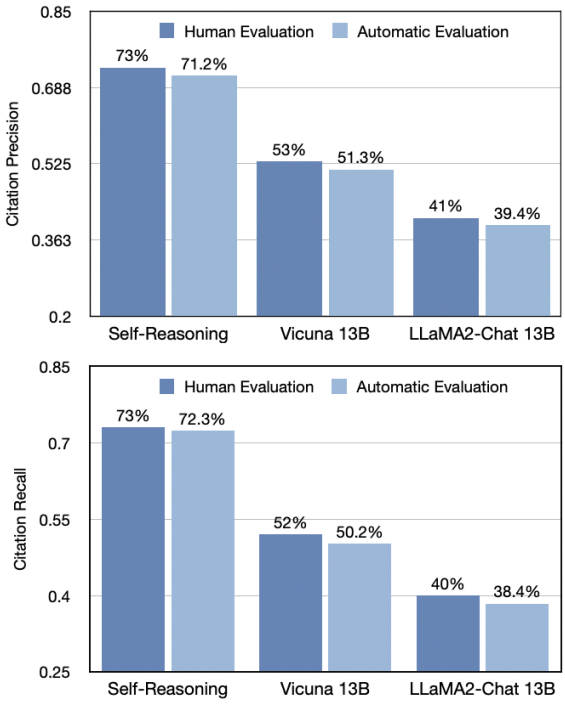

Figure 4: Human citation quality evaluation vs. automatic citation evaluation on the long-form ASQA dataset.

Latency Analysis

We also compared the inference latency of SELF-REASONING RAG with that of Self-RAG and GPT-4. The results show that our method maintains comparable latency to Self-RAG while delivering substantial performance gains. Detailed results are available in the Appendix.

Conclusion

RALMs can effectively enhance the performance of LLMs in handling knowledge-intensive tasks. Despite their effectiveness, notable concerns about their reliability and traceability persist. To address these limitations, we propose a novel SELF-REASONING framework to improve the performance of RALMs by using reasoning trajectories generated by the LLM itself. It is comprised of a relevance-aware process, an evidence-aware selective process, and a trajectory analysis process. We conduct extensive experiments on four public datasets to demonstrate the superiority of our framework over existing state-of-the-art models.

References

Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; and Hajishirzi, H. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*.

Baek, J.; Jeong, S.; Kang, M.; Park, J.; and Hwang, S. 2023. Knowledge-Augmented Language Model Verification. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 1720–1736. Singapore: Association for Computational Linguistics.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G. B.; Lespiau, J.-B.; Damoc, B.; Clark, A.; et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, 2206–2240. PMLR.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Gao, L.; Dai, Z.; Pasupat, P.; Chen, A.; Chaganty, A. T.; Fan, Y.; Zhao, V.; Lao, N.; Lee, H.; Juan, D.-C.; and Guu, K. 2023a. RARR: Researching and Revising What Language Models Say, Using Language Models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 16477–16508. Toronto, Canada: Association for Computational Linguistics.

Gao, T.; Yen, H.; Yu, J.; and Chen, D. 2023b. Enabling Large Language Models to Generate Text with Citations. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 6465–6488. Singapore: Association for Computational Linguistics.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: retrieval-augmented language model pretraining. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Honovich, O.; Aharoni, R.; Herzig, J.; Taitelbaum, H.; Kukliansy, D.; Cohen, V.; Scialom, T.; Szpektor, I.; Hassidim, A.; and Matias, Y. 2022. TRUE: Re-evaluating factual consistency evaluation. *arXiv preprint arXiv:2204.04991*.

Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Izacard, G.; Lewis, P.; Lomeli, M.; Hosseini, L.; Petroni, F.; Schick, T.; Dwivedi-Yu, J.; Joulin, A.; Riedel, S.; and Grave, E. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Jiang, Z.; Xu, F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Active Retrieval Augmented Generation. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7969–7992. Singapore: Association for Computational Linguistics.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. Online: Association for Computational Linguistics.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Li, D.; Rawat, A. S.; Zaheer, M.; Wang, X.; Lukasik, M.; Veit, A.; Yu, F.; and Kumar, S. 2023. Large Language Models with Controllable Working Memory. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 1774–1793. Toronto, Canada: Association for Computational Linguistics.

Liu, N.; Zhang, T.; and Liang, P. 2023. Evaluating Verifiability in Generative Search Engines. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 7001–7025. Singapore: Association for Computational Linguistics.

Mallen, A.; Asai, A.; Zhong, V.; Das, R.; Khashabi, D.; and Hajishirzi, H. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9802–9822. Toronto, Canada: Association for Computational Linguistics.

Menick, J.; Trebacz, M.; Mikulik, V.; Aslanides, J.; Song, F.; Chadwick, M.; Glaese, M.; Young, S.; Campbell-Gillingham, L.; Irving, G.; et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al.

2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ni, J.; Qu, C.; Lu, J.; Dai, Z.; Hernandez Abrego, G.; Ma, J.; Zhao, V.; Luan, Y.; Hall, K.; Chang, M.-W.; and Yang, Y. 2022. Large Dual Encoders Are Generalizable Retrievers. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9844–9855. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

OpenAI, R. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2: 13.

Petroni, F.; Lewis, P.; Piktus, A.; Rocktäschel, T.; Wu, Y.; Miller, A. H.; and Riedel, S. 2020. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*.

Petroni, F.; Piktus, A.; Fan, A.; Lewis, P.; Yazdani, M.; De Cao, N.; Thorne, J.; Jernite, Y.; Karpukhin, V.; Maillard, J.; Plachouras, V.; Rocktäschel, T.; and Riedel, S. 2021. KILT: a Benchmark for Knowledge Intensive Language Tasks. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tur, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2523–2544. Online: Association for Computational Linguistics.

Ram, O.; Levine, Y.; Dalmedigos, I.; Muhlgay, D.; Shashua, A.; Leyton-Brown, K.; and Shoham, Y. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11: 1316–1331.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Stelmakh, I.; Luan, Y.; Dhingra, B.; and Chang, M.-W. 2022. ASQA: Factoid Questions Meet Long-Form Answers. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 8273–8288. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819. New Orleans, Louisiana: Association for Computational Linguistics.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10014–10037. Toronto, Canada: Association for Computational Linguistics.

Xu, F.; et al. 2024. RECOMP: Improving Retrieval-Augmented LMs with Context Compression and Selective Augmentation. In *The Twelfth International Conference on Learning Representations*.

Yan, S.-Q.; Gu, J.-C.; Zhu, Y.; and Ling, Z.-H. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.

Yoran, O.; Wolfson, T.; Ram, O.; and Berant, J. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Yu, W.; Zhang, H.; Pan, X.; Ma, K.; Wang, H.; and Yu, D. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.