Optimizing Knowledge Integration in Retrieval-Augmented Generation with Self-Selection

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) has proven effective in enabling LLMs to produce more accurate and reliable responses. However, it remains a significant challenge how to effectively integrate external retrieved knowledge with internal parametric knowledge in LLMs. In this work, we propose a novel **Self-Selection** RAG framework, where the LLM is made to select from pairwise responses generated with internal parametric knowledge solely and with external retrieved knowledge together to achieve enhanced accuracy. To this end, we devise a **Self-Selection-RGP** method to enhance the capabilities of the LLM in both generating and selecting the correct answer, by training the LLM with Direct Preference Optimization (DPO) over a curated Retrieval-Generation Preference (RGP) dataset. Experimental results with three open-source LLMs (i.e., Llama2-13B-Chat, Mistral-7B and Qwen2.5-7B) well demonstrate the superiority of our approach over other baseline methods on Natural Questions (NQ), TrivialQA and HotpotQA datasets.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various tasks (Brown et al., 2020; Touvron et al., 2023a; OpenAI, 2024). However, their reliance on static parametric knowledge (Kasai et al., 2023; Mallen et al., 2023) often leads to inaccuracy or hallucinations in responses (Welleck et al., 2020; Min et al., 2023). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020; Ram et al., 2023; Asai et al., 2023) supplements LLMs with relevant knowledge retrieved from external sources, attracting increasing research interest. One critical challenge for existing RAG systems is how to effectively integrate internal parametric knowledge with external retrieved knowledge to generate more accurate and reliable results.

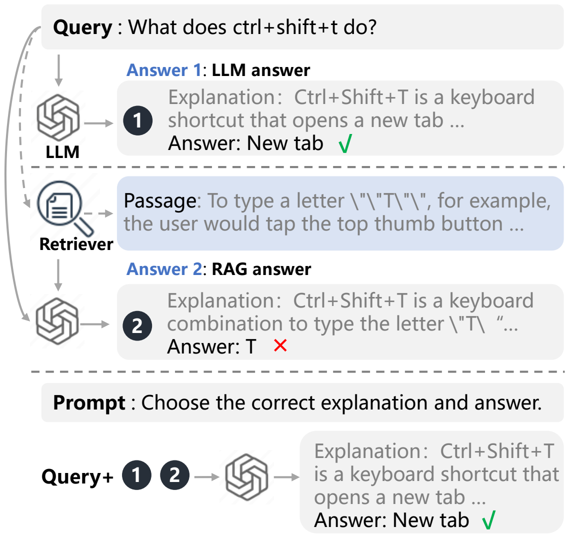

Figure 1: An illustration of the proposed **Self-Selection** framework. Given a query, an LLM is prompted to generate answers and corresponding explanations both with and without external knowledge. These outputs are then fed back into the LLM, which selects one as the final answer along with its explanation.

In existing RAG approaches, LLMs depend either *highly* or *conditionally* upon external knowledge. The former consistently uses the retrieved content as supporting evidence (Lewis et al., 2020; Guu et al., 2020; Trivedi et al., 2023), which often introduces irrelevant or noisy information and overlooks the valuable internal knowledge in LLMs, resulting in sub-optimal results. In comparison, the latter integrates external knowledge into LLMs conditionally based on specific strategies, such as characteristics of input query (Mallen et al., 2023; Jeong et al., 2024; Wang et al., 2023), probability of generated tokens (Jiang et al., 2023b; Su et al., 2024), or relevance of retrieved content (Zhang et al., 2023; Xu et al., 2024; Liu et al., 2024). The query-based and token-based strategies generally utilize a fixed question set or a predefined threshold to decide whether to incorporate external knowledge, limiting their effectiveness due to **incomplete**

information; the relevance-based strategy employs an **additional validation module** to assess the retrieved content, with its accuracy largely determining the quality of the final responses.

In this work, we explore empowering the **LLM itself** to determine the correct result by **holistically** evaluating the outputs generated with and without external knowledge. As illustrated in Figure 1, given a query "What does Ctrl+Shift+T do?", we instruct the LLM to generate the *LLM Answer* (i.e., "New tab") and corresponding explanation (i.e., reasoning steps) with its internal parametric knowledge. Meanwhile, we employ a retriever to obtain the relevant passages from external knowledge bases and feed the query and retrieved passages to LLMs to produce the *RAG Answer* (i.e., "T") and the corresponding explanation. Next, we instruct LLMs to take the query, *LLM Answer* with its explanation and *RAG Answer* with its explanation as input to choose the more accurate one (i.e., "New tab"). In this manner, both internal and external knowledge related to the query are comprehensively considered, enabling the LLM to generate more accurate responses, while the RAG framework remains simple by avoiding the need for additional modules.

Accordingly, we devise a novel **Self-Selection** RAG framework that leverages the LLM itself to identify the more accurate answer to a query by evaluating both *LLM Answer* and *RAG Answer*, along with their respective explanations. We validate the performance of the proposed **Self-Selection** framework with three open-sourced LLMs (see Section 3.2) and find that it tends to fail in some scenarios, which we attribute to its limited capacity in distinguishing the correct answer from the incorrect one. To enhance the accuracy of the LLM selecting the right one among multiple responses generated from different knowledge sources, we develop a **Self-Selection-RGP** method, leveraging Direct Preference Optimization (DPO) (Rafailov et al., 2023) to fine-tune the LLM with a curated Retrieval-Generation Preference (**RGP**) dataset. To construct this RGP dataset, we employ GPT-3.5 (OpenAI, 2024) to generate an *LLM Answer* and an *RAG Answer* for each query sampled from WebQuestions (Berant et al., 2013), SQuAD 2.0 (Rajpurkar et al., 2018) and SciQ (Welbl et al., 2017), and then retain only the pairs consisting of one correct answer and one incorrect answer, each accompanied by its corresponding explanation. It consists of $3,756$ pairs of *LLM Answer* and *RAG answer* with their respective explanations, which we promise to release to the public to facilitate future research.

With this dataset, we train three different LLMs, including Mistral-7B (Jiang et al., 2023a), LLaMa-2-13B-Chat (Touvron et al., 2023b) and Qwen-2.5-7B- Instruct (Qwen et al., 2025), and evaluate them on three widely used datasets, i.e., Natural Questions (NQ) (Kwiatkowski et al., 2019), TrivialQA (Joshi et al., 2017), and HotpotQA (Yang et al., 2018). It is demonstrated that our **Self-Selection-RGP** method consistently achieves high effectiveness across various retrieval settings and different LLMs, enhancing the robustness and stability of RAG systems. Moreover, additional experiments reveal that our **Self-Selection-RGP** method not only enhances LLMs' ability to distinguish valid answers from noisy ones but also improves their answer generation capabilities. We further validate the rationale of each design in our method through ablation studies.

Major contributions of our paper are three-fold:

- We introduce a novel **Self-Selection** RAG framework that leverages the LLM itself to determine the correct answer by evaluating a pair of responses generated with internal parametric knowledge solely and also with external retrieved knowledge.

- We propose a **Self-Selection-RGP** method that applies Direct Preference Optimization (DPO) to enhance LLMs in both identifying and generating correct answers with a curated Retrieval-Generation Preference (RGP) dataset.

- Extensive experiments with three open-sourced LLMs achieve superior performance on three widely-used datasets, demonstrating the effectiveness of our proposed **Self-Selection** method.

2 Self-Selection

In this section, we elaborate on the proposed **Self-Selection** framework for enhanced Retrieval-Augmented Generation (RAG).

2.1 Preliminaries

For an LLM represented by $\mathcal{M}$, given a prompt $\bar{p}$ and a query q as inputs, it returns a textual answer

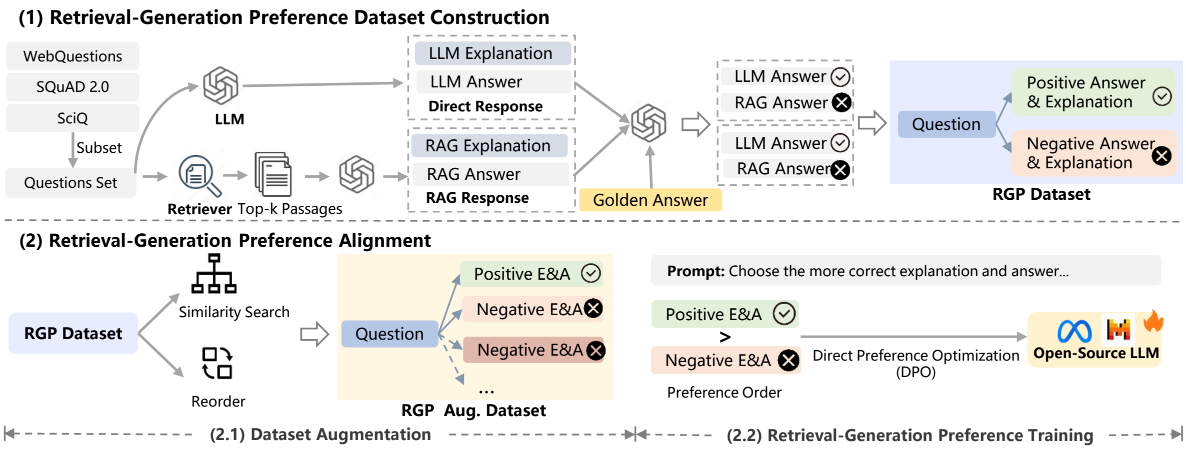

Figure 2: An illustration of the proposed **Self-Selection-RGP** method.

$\bar{a}$ as the output, which is formally expressed as

$$\bar{a} = \mathcal{M}(\bar{p}, q). \tag{1}$$

An Retrieval-Augmented Generation (RAG) system employs a retriever to enhance the capability of the LLM by enabling it to access external knowledge beyond its internal parametric knowledge (Lee et al., 2019; Guu et al., 2020). Given a query q, the retriever $\mathcal{R}$ searches for the relevant knowledge (e.g., passages) C from an external knowledge base or corpus. A common approach for RAG is to include the retrieved passages C in the input to the LLM to improve the response quality. Formally,

$$C = \mathcal{R}(q), \tag{2}$$

$$\hat{a} = \mathcal{M}(\hat{p}, q, C), \tag{3}$$

where $\hat{p}$ represents the prompt used in RAG and $\hat{a}$ denotes the answer predicted by the LLM taking into account the retrieved passages C.

2.2 Task Formulation

In this part we present the formulation of our **Self-Selection** framework. An illustration is provided in Figure 1. Given a query q, we first prompt an LLM $\mathcal{M}$ with $\bar{p}$ denoting the prompt to output the answer $\bar{a}$ with its explanation $\bar{e}$, where we refer to $\bar{a}$ as the *LLM Answer* and $\bar{e}$ as the *LLM Explanation*. Next, we use the retriever $\mathcal{R}$ to gather relevant passages C (Eq. (2)) to the same query q. Then, we prompt the LLM $\mathcal{M}$ with $\hat{p}$ while providing q and C in the input, to generate $\hat{a}$ with its explanation $\hat{e}$, where we refer to $\hat{a}$ and $\hat{e}$ as the *RAG Answer* and the *RAG Explanation*. Finally, we prompt the LLM M with a prompt p by taking the query q, the *LLM Answer* $\bar{a}$ with its *LLM Explanation* $\bar{e}$, the *RAG Answer* $\hat{a}$ with its *RAG Explanation* $\hat{e}$ as inputs to select one as the final answer a and final explanation e. Formally,

$$(\bar{a}, \bar{e}) = \mathcal{M}(\bar{p}, q); \tag{4}$$

$$(\hat{a}, \hat{e}) = \mathcal{M}(\hat{p}, q, C); \tag{5}$$

$$(a, e) = \mathcal{M}(p, q, (\bar{a}, \bar{e}), (\hat{a}, \hat{e})). \tag{6}$$

2.3 Self-Selection-RGP

Motivation. We evaluate the performance of the proposed Self-Selection framework on widely used QA datasets using existing open-source models, without any model parameter updates. We report the experimental results in Table 2 of Section 3. The Self-Selection framework is promising in enhancing LLMs' answer generation by fusing internal knowledge with external knowledge, but directly applying such knowledge fusion does not always bring enhancements. One assumption is that LLMs struggle to reliably discern the correct answer between two candidates generated from different knowledge sources. In essence, this knowledge selection process is consistent with the goal of preference alignment in LLMs, i.e. generating the desired (positive) sample while rejecting the undesired (negative) one from a pair of preference data. To address this challenge, we explore tuning LLMs through preference alignment techniques to enhance their ability to discern and select the correct answer from two candidates generated by different knowledge sources.

Retrieval-Generation Preference Dataset. Here we explain how we build the Retrieval-Generation Preference (RGP) dataset used for fine-tuning LLMs in **Self-Selection-RGP**:

• *Preference Candidate Generation.* We first employ an LLM to produce two sets of responses for each query q: (i) an *LLM Answer* $\bar{a}$ with its *LLM*

Explanation $\bar{e}$, derived from the model's internal parametric knowledge; and (ii) an *RAG Answer* $\hat{a}$ with its *RAG Explanation* $\hat{e}$, relying on the externally retrieved information. Specifically, we randomly select a subset of QA pairs from three existing open-domain QA datasets, including WebQuestions (Berant et al., 2013), SQuAD2.0 (Rajpurkar et al., 2018), and SciQ (Welbl et al., 2017). Let $\mathcal{D}$ denote the obtained set of QA pairs. Formally,

$$\mathcal{D} = \{q^{(i)}, a_g^{(i)}\}_{i=1}^{N} \quad (7)$$

where a_g is the golden answer to the query q, N is the number of QA pairs and i is the i-th QA pair in $\mathcal{D}$. For each query q in $\mathcal{D}$, we utilize a retriever $\mathcal{R}$ to retrieve the top-K passages C from a corpus (Eq. (2)). To ensure the quality of the constructed preference dataset, we employ GPT-3.5 (Ouyang et al., 2022) as the model $\mathcal{M}$ for candidate answer and explanation generation given a query. According to Eq. (4) and Eq. (5), we generate the answers and explanations ($\bar{a}$, $\bar{e}$) and ($\hat{a}$, $\hat{e}$). Finally, we obtain a collection of preference candidates for constructing the RGP datasets. Formally, the D is expanded as

$$\mathcal{D} = \{q^{(i)}, a_g^{(i)}, \bar{a}^{(i)}, \bar{e}^{(i)}, \hat{a}^{(i)}, \hat{e}^{(i)}\}_{i=1}^{N}. \quad (8)$$

• *Preference Data Filtering.* In the RGP dataset, each instance should include both a desired (positive) answer and an undesired (negative) answer. We filter these required instances from the collection $\mathcal{D}$. For each instance in D, we first employ GPT-3.5 to assess whether the *LLM Answer* $\bar{a}$ and the *RAG Answer* $\hat{a}$ are correct by comparing each to the golden answer a_g. After that, we only retain the instances that contain one right answer and one wrong answer, where (i) $\bar{a}$ is correct but $\hat{a}$ is incorrect; or (ii) $\hat{a}$ is correct but $\bar{a}$ is incorrect. Based on this strategy, we gather all appropriate instances in $\mathcal{D}$ to build our RGP dataset $\mathbb{D}$. Formally,

$$\mathbb{D} = \{q^{(j)}, a_g^{(j)}, (a_p^{(j)}, e_p^{(j)}), (a_n^{(j)}, e_n^{(j)})\}_{j=1}^{M} \quad (9)$$

where a_p and e_p represent the positive answer and its explanation, a_n and e_n represent the negative answer and its explanation, M denotes the number of instances and j denotes the j-th instance in $\mathbb{D}$. Finally, we retain $3,756$ preference instances in the RGP dataset. We promise to release it for facilitating future reseach.

Retrieval-Generation Preference Alignment. With the constructed RGP dataset, we train open-source LLMs to enhance their ability to distinguish the positive answer from the negative counterpart.

• *Dataset Augmentation.* To improve LLMs' preference alignment, we first augment the RGP dataset through a simple yet effective approach to produce more preference instances. In particular, given a query q in RGP, we search for the top-K similar queries in the RGP datasets and we regard all answers to these K queries as negative answers to the query q. Formally, for each query q in the RGP dataset $\mathbb{D}$, we denote the obtained most similar queries and their responses in RGP as $\mathbb{G}$:

$$\mathbb{G}^{(i)} = \{q^{(j)}, y_w^{(j)}, y_l^{(j)} \mid \underset{\text{top-K}}{\operatorname{argmax}}\, S\left(q^{(i)}, q^{(j)}\right)\} \quad \forall q^{(i)} \in \mathbb{D}, i \neq j \quad (10)$$

where $S(q^{(i)}, q^{(j)})$ represents the similarity score between $q^{(i)}$ and $q^{(j)}$, and $y_w^{(j)}$ and $y_l^{(j)}$ represent the corresponding positive and negative response (i.e., answer with its explanation) for $q^{(j)}$ in RGP. For one query $q^{(i)}$, $y_w^{(i)}$ and $y_l^{(i)}$ are the original positive and negative response in RGP. Then, we regard all $y_w^{(j)}$ and $y_l^{(j)}$ in the obtained set $\mathbb{G}^{(i)}$ as the additional negative responses to $q^{(i)}$. For each query, now we have 1 positive response and $2K + 1$ negative responses, which can be used to form $2K + 1$ pairs of preference instances. Let $\mathbb{D}_{aug}$ denote the augmented RGP dataset. Formally,

$$\mathbb{D}_{aug}^{(i)} = \left\{\left(q^{(i)}, y_w^{(i)}, y_{lj}^{(i)}\right)\right\}_{i=1}^{M}, j = 1, 2, \ldots, 2K + 1, \quad (11)$$

where $y_{lj}^{(i)}$ is the j-th negative response to the query $q^{(i)}$.

• *Retrieval-Generation Preference Training.* With the augmented preference dataset, our goal is to train open-sourced LLMs to enhance their capabilities in distinguishing the correct answers from the incorrect ones. During the preference alignment phase of LLM training, each instance in the augmented dataset $\mathbb{D}_{aug}$ comprises three key elements: an input x, a desired response y_w, and an undesired response y_l, which is denoted as $y_w \succ y_l \mid x$. Specifically, the input x consists of a query q, the desired response y_w, the undesired response y_l, and a prompt p designed to instruct the LLM M to choose between y_w and y_l (see Eq. (6)). The desired response y_w includes the correct answer along with its explanation, while the undesired response y_l contains an incorrect answer and its explanation. To enhance the robustness of the trained model, we randomly alternate the order of y_w and y_l within the input x.

We adopt Direct Preference Optimization

(DPO) (Rafailov et al., 2023) to train LLMs. It enables preference data to be directly associated with the optimal policy, eliminating the need for any additional reward model. DPO formulates a maximum likelihood objective as follows:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{ref}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{ref}(y_l \mid x)}\right)\right], \tag{12}$$

where β represents the deviation of the policy π_θ from the reference model π_{ref}. Our proposed optimization method aims to enhance LLMs' answer selection capability, enabling them to holistically evaluate multiple responses generated from diverse knowledge sources and identify the most accurate one among them. In addition, we also hope this optimization method can further improve the inherent ability of LLMs in answer generation (more analysis is provided in Section 3.3).

3 Experiments

We evaluate the proposed method with extensive experiments. We experiment with two variants based on the proposed **Self-Selection** framework: i) **Self-Selection-Ori**, i.e. the RAG method that applies Self-Selection on vanilla LLMs, and ii) **Self-Selection-RGP**, i.e. the RAG method that applies Self-Selection on the LLMs trained with our augmented RGP dataset. For a comprehensive evaluation, we seek to address the following questions: **RQ1**: How does Self-Selection-RGP perform compared to other compared methods? **RQ2**: To what extent can Self-Selection-RGP affect the LLMs' inherent ability in answer generation? **RQ3**: What is the effect of each design in our proposed Self-Selection-RGP? **RQ4**: Whether Self-Selection-RGP is generalizable across different retrieval settings?

3.1 Experimental Setup

Datasets. We use three open-domain QA datasets, namely, Natural Question (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017) and HotpotQA (Yang et al., 2018), along with the retriever BGE (Xiao et al., 2024). Following prior works (Trivedi et al., 2023; Jeong et al., 2024), we use the same test split for each dataset with the same external corpus to evaluate RAG methods. We provide their statistics in Table 1, and implementation details are deferred to the appendix.

Table 1: Statistics of the test datasets.

Dataset	# Passages	# QA Pairs
Natural Questions (NQ)	21,015,324	500
TriviaQA	21,015,324	500
HotpotQA	5,233,329	500

Baselines. We compare our methods with below baselines:

- **LLM Only**: The response to each query is generated solely by LLMs.
- **Standard RAG**: The response to each query is produced by LLMs after appending the retrieved passages to the input.
- **Self-RAG** (Asai et al., 2024): Specialized reflection tokens are utilized to enable LLMs to control retrieval and evaluate the relevance of the retrieved content during reasoning.
- **SURE** (Kim et al., 2024): LLMs first generate summaries of the retrieved passages for each candidate answer, and then identify the most plausible answer by evaluating each summary's validity and ranking.

Evaluation Metrics. We use Exact Match (EM), F1 score and Accuracy (Acc) as evaluation metrics (Mallen et al., 2023).

3.2 Main Results (RQ1)

We verify the effectiveness of the proposed **Self-Selection** framework by comparing its performance with baseline methods. Table 2 shows our main results, from which we make below key observations: 1) With Mistral-7B and Qwen-2.5-7B-Instruct as the base LLMs, our Self-Selection-RGP model consistently outperforms all compared methods on three datasets. On the TriviaQA dataset, Self-Selection-RGP scores 67.0 and 70.2 in accuracy, making significant improvements of 5.4 and 8.0 points, respectively, compared to the best baselines, which score 61.6 and 62.2. On the NQ and HotpotQA datasets, the performance gains of Self-Selection-RGP in accuracy are relatively smaller compared to those observed on the TriviaQA dataset. These performance improvements highlight the effectiveness of our Self-Selection-RGP method. 2) With LLama2-13B-Chat as the base LLM, our Self-Selection-RGP model consistently delivers strong performance. In particular, Self-Selection-RGP exhibits superior performance on the TriviaQA and HotpotQA datasets, achieving accuracies of 66.0 and 36.8, respectively, i.e., 2.2 and 3.8 points higher than the best baseline SURE.

Table 2: Main results of our proposed methods and all baseline methods.

LLM	Method	NQ EM	NQ F1	NQ acc	TriviaQA EM	TriviaQA F1	TriviaQA acc	HotpotQA EM	HotpotQA F1	HotpotQA acc
Mistral(7B)	LLM Only	21.8	35.5	34.0	41.2	53.4	52.6	20.4	28.0	24.0
	Standard RAG	35.8	51.2	51.0	45.8	58.1	59.8	**30.8**	41.0	36.8
	Self-RAG	-	-	-	29.0	43.2	60.6	14.0	24.6	35.4
	SURE	**39.0**	52.4	47.6	48.6	59.7	61.6	22.4	34.1	28.8
	Self-Selection-Ori	34.6	50.1	50.2	48.4	61.2	62.8	30.6	41.7	36.8
	Self-Selection-RGP	37.8	**52.5**	**53.6**	**54.4**	**66.2**	**67.0**	30.0	**42.4**	**37.2**
Qwen2.5(7B)	LLM Only	22.6	32.6	28.8	49.4	57.8	54.6	19.0	25.6	20.8
	Standard RAG	43.2	**56.3**	52.8	55.6	63.8	62.2	38.4	47.2	40.8
	SURE	35.0	48.3	39.4	51.2	61.0	58.6	30.0	39.3	32.2
	Self-Selection-Ori	42.0	55.1	52.0	58.8	66.9	65.2	38.8	47.5	40.6
	Self-Selection-RGP	**44.0**	**56.3**	**53.2**	**61.8**	**72.2**	**70.2**	**40.6**	**49.9**	**43.4**
Llama2(13B)	LLM Only	21.2	31.9	28.2	43.2	50.1	48.0	18.8	26.5	22.4
	Standard RAG	24.6	37.0	45.2	35.2	46.1	55.0	28.6	39.9	35.0
	Self-RAG	-	-	-	17.2	36.6	63.4	5.8	18.2	31.8
	SURE	**39.4**	**52.3**	**52.0**	50.4	63.0	63.8	24.8	37.4	33.0
	Self-Selection-Ori	31.6	43.8	45.2	43.0	53.5	56.6	27.4	39.8	34.2
	Self-Selection-RGP	36.6	49.2	46.2	**56.6**	**66.3**	**66.0**	**29.0**	**41.4**	**36.8**

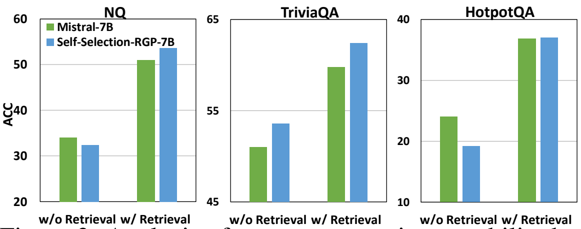

Figure 3: Analysis of answer generation capability before and after preference alignment training. "Mistral-7B" denotes the vanilla LLM before preference alignment training; "Self-Selection-RGP-7B" refers to the LLM obtained after training Mistral-7B with the preference alignment dataset.

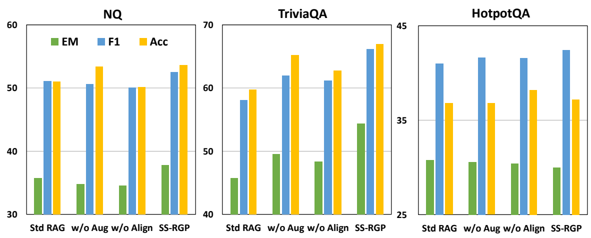

Figure 4: Ablation study. "Std RAG" denotes Standard RAG; "w/o Aug", "w/o Align" denote the method without Dataset Augmentation or Preference Alignment; "SS-RGP" is our Self-Selection RAG method.

In comparison, on the NQ dataset, it underperforms SURE. This discrepancy likely stems from the inherent difficulty of NQ questions for the model, which hampers our method's ability to distinguish correct answers effectively.

3.3 Analysis of Answer Generation Capability of LLMs (RQ2)

The main results have shown that LLMs trained with the preference dataset acquire remarkable improvements in distinguishing correct responses from incorrect ones. We then address **RQ2** by comparing LLMs' answer generation performance before and after preference alignment training.

We adopt *Mistral-7B* as the base LLM and compare it with *Self-Selection-RGP-7B* model trained on the augmented RGP dataset. We evaluate their answer generation capabilities on the three datasets. See results in Figure 3. We make the following observations: 1) When not using external knowledge, Self-Selection-RGP-7B exhibits improved performance on TriviaQA and slightly worse performance than Mistral-7B on NQ and HotpotQA. 2) Under RAG setting, Self-Selection-RGP-7B consistently outperforms Mistral-7B on the three datasets, showing enhanced answer generation abilities.

These empirical findings reveal that the Self-Selection-RGP-7B model exhibits enhanced capabilities not only in answer selection, but also in answer generation when using external knowledge. This indicates that training LLMs on the augmented RGP dataset can enhance their ability to generate high-quality answers.

3.4 Ablation Study (RQ3)

To verify the effect of each design in our proposed method, we conduct an ablation study with Mistral-7B on NQ, TriviaQA and HotpotQA datasets. We utilize three methods for ablation in addition to our **Self-Selection-RGP**: 1) **Standard RAG**, which appends the retrieved passages to the input of the vanilla LLM; 2) **w/o Dataset Augmentation**, which removes the step of dataset augmentation

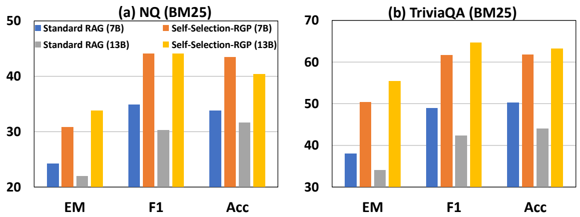

Figure 5: Effects of using a different retriever. Our **Self-Selection-RGP** and Standard RAG both use BM25 as the new retriever. We adopt Mistral-7B as the base LLM for Standard RAG (7B) and Self-Selection-RGP (7B), and Llama2-13B-Chat for Standard RAG (13B) and Self-Selection-RGP (13B).

and trains Mistral-7B with the original RGP dataset only; 3) **w/o Preference Alignment**, which removes the step of preference alignment for the LLM and applies the **Self-Selection** on the vanilla LLM. For fair comparisons, we use the same prompt as input for generating candidate answers in both the Standard RAG model and other models.

We present the results in Figure 4, from which we observe: 1) Removing either the step of dataset augmentation or preference alignment results in a performance drop in F1 and Accuracy on all three datasets, highlighting the rationale of the design in our proposed **Self-Selection-RGP** method. 2) Comparably, the removal of preference alignment results in a more substantial decrease in performance, revealing the importance of teaching LLMs to choose the correct answer from multiple candidates. 3) Compared to the Standard RAG method, our method consistently yields superior performance on all three datasets, demonstrating great effectiveness in enhancing RAG systems.

3.5 Impact of Retrieval Settings (RQ4)

Effects of A Different Retriever. We test the compatibility of the proposed Self-Selection framework with different retrieval methods. Beyond the BGE retriever considered in Table 2, we also use BM25 (Robertson et al., 1994) as the retriever, and compare our Self-Selection-RAG method with Standard RAG to see if our method still maintains superior performance.

We use Mistral-7B and Llama2-13B-Chat as the base models and present their performance on NQ and TriviaQA in Figure 5, from which we make following obersevations. With BM25 as the retriever, our proposed Self-Selection-RGP method consistently outperforms the Standard RAG method in each setting as illustrated in Figure 5, revealing the compatibility and generalizability of our method regarding new retrieval techniques like BM25. To be more specific, as shown in Figure 5 (a) and (c), our Self-Selection-RGP (7B) and Self-Selection-RGP (13B) outperform the Standard RAG (7B) and Standard RAG (13B) models by substantial margins on all three metrics over both NQ and TriviaQA datasets. This underscores the high effectiveness and strong compatibility of our Self-Selection-RGP method across different retrievers.

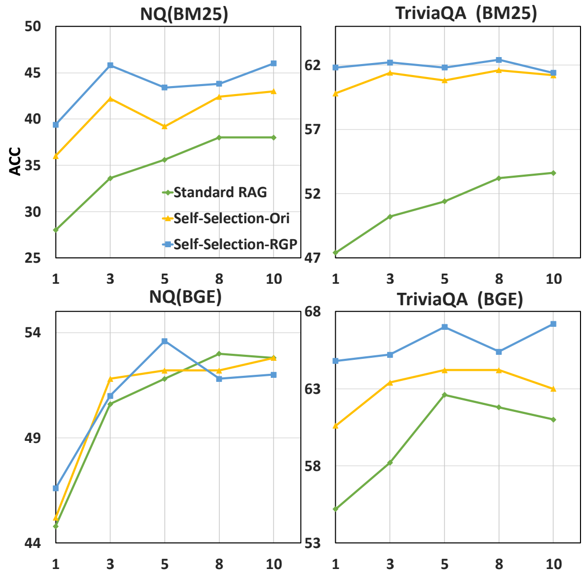

Figure 6: Effects of retrieved passage counts with Mistral-7B as the base LLM.

Effects of Retrieved Passage Count. Next, we investigate the effects of the number of retrieved passages on the performance. Specifically, we adopt BGE and BM25 as the retriever and Mistral-7B as the base LLM. We compare performance across three RAG methods, i.e. Standard RAG, Self-Selection-Ori, and Self-Selection-RGP, on both NQ and TriviaQA datasets. The experimental results in Figure 6 lead to following observations: 1) Increasing the number of retrieved passages improves accuracy in all three RAG methods initially. However, beyond a certain threshold, adding more passages yields marginal or negative effects, likely due to information overload within the LLM's constrained context window. 2) On the TriviaQA dataset, Self-Selection-RGP consistently achieves the highest performance across different retrieved passages and retrievers, followed by Self-Selection in second, and Standard RAG in last. 3) On the NQ dataset, Self-Selection-RGP consistently ranks the first using BM25. When using BGE, all methods perform competitively across varying numbers of retrieved passages. Notably, Self-Selection-RGP performs best with 5 retrieved passages. Standard RAG outperforms the LLM-only approach by integrating retrieved passages, establishing a strong baseline, as shown in Table 2. Our method achieves comparable performance, demonstrating great effectiveness across varying retrieved passage counts.

4 Related Work

4.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020) has been widely used for improving LLMs' performance across various tasks by incorporating an Information Retriever (IR) module to leverage external knowledge. Most RAG systems (Lewis et al., 2020; Ram et al., 2023; Izacard et al., 2023) integrate retrieved knowledge directly into the input; some utilize Chain of Thought (CoT) (Wei et al., 2022; Trivedi et al., 2023) or task decomposition (Xu et al., 2024; Wang et al., 2024; Kim et al., 2024) to integrate external knowledge in intermediate reasoning steps or subtasks. Though effective, the indiscriminate use of external knowledge may introduce noise, degrading quality of generated responses. Conditional use of external knowledge in RAG has also been investigated. Adaptive retrieval methods decide whether to retrieve and utilize external knowledge based on query characteristics (Mallen et al., 2023; Wang et al., 2023; Jeong et al., 2024) or next token generation probability (Asai et al., 2024; Jiang et al., 2023b; Su et al., 2024; Wang et al., 2024), while relevance-based methods (Zhang et al., 2023; Xu et al., 2024; Liu et al., 2024) employ a relevance verification module to filter retrieved passages. The former often rely solely on the input query or generated tokens, with limited effectiveness as they may only acquire incomplete information; the latter largely rely on an additional verification module, leading to increased complexity of RAG systems and final response's quality highly sensitive to its verification accuracy. Compared with previous methods, our approach empowers the LLM itself to holistically evaluate and reconcile responses from only its internal parametric knowledge and also from externally retrieved information, aiming to produce more accurate responses.

4.2 Preference Alignment for LLMs

Preference alignment is aimed at improving the reliability of LLMs (Ouyang et al., 2022) by enabling them to evolve from their generated responses and environmental feedback. Among existing techniques, Reinforcement Learning from Human Feedback (RLHF) leverages human-provided feedback to train reward models, ensuring that LLMs produce responses aligned well with human preferences (Christiano et al., 2017; Ziegler et al., 2019), which however tends to suffer limited scalability and high training complexity. However, RLHF requires extensive human annotation to train the reward model and involves a complex three-stage process, resulting in limited scalability and high training complexity. To improve the scalability, Reinforcement Learning from AI Feedback (RLAIF) (Bai et al., 2022; Lee et al., 2024) utilizes the feedback from the LLM itself to train a reward model to optimize LLM performance through reinforcement learning. Direct Preference Optimization (DPO) (Rafailov et al., 2023) defines preference loss directly via a change of variables, treating the LLM itself as its reward model, which substantially reduces the training complexity by eliminating the need for an additional reward model. In RAG systems, some works use the signals generated by LLMs to optimize the retriever (Bonifacio et al., 2022; Shi et al., 2024) to retrieve LLM-preferred data, and some align LLMs with specific domain knowledge and specific tasks through reinforcement learning (Zhang et al., 2024; Yang et al., 2024; Salemi et al., 2024; Dong et al., 2024; Song et al., 2024). In this work, we construct a retrieval-generation preference dataset automatically and utilize it to strengthen LLMs' answer selection and generation capabilities in RAG systems via DPO.

5 Conclusion

In this work, we propose a novel **Self-Selection** framework to improve the accuracy and reliability of responses generated by LLMs in RAG systems. Our method allows the LLM to select the more accurate one from a pair of responses generated based on internal parametric knowledge solely and by integrating external retrieved knowledge, to achieve enhanced performance. To strengthen the capabilities of the LLM in generating and selecting correct answers, we develop a **Self-Selection-RGP** method that trains the LLM with Direct Preference Optimization over a newly built Retrieval-Generation Preference (RGP) dataset. We conduct extensive experiments and analyses, which well validate the effectiveness of the proposed method. We hope this work paves the way for the development of more robust and reliable LLMs in RAG.

Limitations

Although we have made some discoveries and improvements, we must acknowledge some limitations in our work:

First, the limitation of computing resources restricts our experiments to open source LLM models of limited and medium scale, such as Mistral 7B, and Llama2-13B-Chat. We will explore applying our method to larger open source models in future work. The indicators used in our experiments, such as F1 and Accuracy, may overestimate the correctness of the response. These indicators only verify the degree of overlap of the answer or whether it exists in the response.

Second, the knowledge base and the retriever have an important impact on the quality of the retrieved data. We only use Wikipedia paragraph pairs to verify the effectiveness of our method on BGE and BM25 respectively. The application of RAG in real situations usually involves multi-source retrieval. We will explore applying our method to the combination of internal knowledge and multi-source knowledge in future work.

Ethical Statements

All datasets used are publicly available and comply with the terms set by the original authors. The QA pairs are from WebQuestions (Berant et al., 2013), SQuAD2.0 (Rajpurkar et al., 2018), and SciQ (Welbl et al., 2017), with proper citations. The external knowledge base is the 2018 English Wikipedia, used in accordance with fair use policies (Karpukhin et al., 2020; Jeong et al., 2024).

We use open-source models, including the BGE retriever (Xiao et al., 2024), GPT-3.5 for answer generation, and a Sentence transformer (Reimers and Gurevych, 2020) for dataset augmentation. All models (Mistral 7B (Jiang et al., 2023a), Qwen-2.5-7B-Instruct (Qwen et al., 2025), Llama2-13B-Chat (Touvron et al., 2023b)) are based on open-source frameworks to ensure transparency and reproducibility.

We prioritize privacy and ensure that all data used is anonymized and does not contain personally identifiable information, conducting research responsibly and ethically.

References

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46. Association for Computational Linguistics.

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

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544. Association for Computational Linguistics.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2387–2392. Association for Computing Machinery.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 4302–4310. Curran Associates Inc.

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. *Preprint*, arXiv:2406.18676.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

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*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b.

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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

Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime qa: What's the answer right now? In *Advances in Neural Information Processing Systems*, pages 49025–49043. Curran Associates, Inc.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain QA of LLMs. In *The Twelfth International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096. Association for Computational Linguistics.

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

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. RA-ISF: Learning to answer and understand from retrieval augmentation via iterative self-feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4730–4749. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 752–762. Association for Computing Machinery.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384. Association for Computational Linguistics.

Maojia Song, Shang Hong Sim, Rishabh Bhardwaj, Hai Leong Chieu, Navonil Majumder, and Soujanya Poria. 2024. Measuring and enhancing trustworthiness of llms in rag through grounded attributions and learning to refuse. *Preprint*, arXiv:2409.11242.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Hongru Wang, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Guanhua Chen, Huimin Wang, and Kam fai Wong. 2024. Self-dc: When to retrieve and when to generate? self divide-and-conquer for compositional unknown questions.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

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. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1362–1373. Association for Computing Machinery.

Diji Yang, Jinmeng Rao, Kezhen Chen, Xiaoyuan Guo, Yawen Zhang, Jie Yang, and Yi Zhang. 2024. Im-rag: Multi-round retrieval-augmented generation through learning inner monologues. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 730–740. Association for Computing Machinery.

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. Association for Computational Linguistics.

Yichi Zhang, Zhuo Chen, Yin Fang, Yanxi Lu, Li Fangming, Wen Zhang, and Huajun Chen. 2024. Knowledgeable preference alignment for LLMs in domain-specific question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 891–904. Association for Computational Linguistics.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain QA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4710–4728. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Appendix

A.1 Implementation Details

A.1.1 Dataset Construction

In total, we sample 11,756 QA pairs from WebQuestions (Berant et al., 2013), SQuAD2.0 (Rajpurkar et al., 2018), and SciQ (Welbl et al., 2017). We use the official 2018 English Wikipedia as the external knowledge base, similar to prior works (Karpukhin et al., 2020; Jeong et al., 2024; Asai et al., 2024). We use pre-trained BGE (i.e., bge-large-en-v1.5) (Xiao et al., 2024) as the retriever to obtain the relevant passages for each question. For each query, we retrieve a variable number of passages, ranging from 1 to 5.

We adopt GPT-3.5 as our LLM for generating answers and explanations for each question, as well as evaluating the consistency between two given answers. We control the ratio of correct label retrieval data to non-retrieval data in the dataset to be 1:1, and construct a paired-answer dataset. The training set consists of 3,756 pairs of LLM answers and RAG answers with respective explanations, while the test set consists of 426 pairs.

We utilize a Sentence transformer model (i.e., all-mpnet-base-v2) (Reimers and Gurevych, 2020) to identify the top-K similar questions for dataset augmentation.To control the label balance in the training data, we also consider randomly mixing the order of candidate answers. After augmentation, we retain 21,928 preference instances for model training.

Specifically, we construct preference-based training data (RGP dataset) from paired labeled QA data. The prompt used for this process is shown in Table Table 3.

A.1.2 Model Training Setup

For model training, we adopt the widely-used Mistral 7B (i.e., Mistral-7B-Instruct-v0.2) (Jiang et al., 2023a), Qwen-2.5-7B- Instruct (Qwen et al., 2025) and Llama2-13B-Chat (Touvron et al., 2023b) as the base LLMs. We apply DPO with Low-Rank Adapters (LoRA) (Hu et al., 2022) to train LLMs. We conduct all our experiments on a GPU machine with 4 A800 NVIDIA RTX GPUs. We explore learning rates within the range of 5×10^{-6} to 7×10^{-5} and train each model for 3 epochs. The checkpoint with the highest accuracy on the validation set is selected for further use. The training process takes approximately 2-4 hours for each model.

Retrieval-Generation Preference Alignment Prompt

Task
Based on the given question, choose the more correct and reasonable answer.
1. For each question, provide two alternative answers and their corresponding explanations, and select one based on the explanations.
2. Read the answers and explanations, and choose the more accurate answer based on the content of the explanations.
3. Directly output the final answer and explanation without any irrelevant content."
Output Format
Explanation: Explanation
Answer: Answer
Question
In what century were the first modern greenhouses constructed?
Explanations and Answers
1.
Explanation:The first modern greenhouses were constructed in the 19th century as advancements in glass production and architecture allowed for larger structures with better temperature regulation and ventilation. This century saw significant innovations in horticulture and agriculture, leading to the development of greenhouse technology as we know it today. Additionally, during this time, there was a growing interest in exotic plants and botanical collections, further fueling the need for controlled environments provided by greenhouses.
Answer:19th century
2.
Explanation:The first modern greenhouses were built in Europe in the 16th century to house exotic plants brought back from explorations abroad. This information is contained in the third passage which states that the "first modern greenhouses were built in Europe in the 16th century to keep exotic plants brought back from explorations abroad." Therefore, based on the information provided in the passages, the correct answer to the question is the 16th century.
Answer:16th century",

Table 3: The prompt for Retrieval-Generation Preference.

A.1.3 Inference Settings

For model inference, we use the same retriever BGE (i.e., bge-large-en-v1.5) (Xiao et al., 2024) across all compared methods for a fair comparison. We retrieve the top 5 passages for each query from the external knowledge base.For the zero-shot setting, we follow the official settings of prior work (Asai et al., 2024; Kim et al., 2024). Since the training data of Self-RAG includes the NQ dataset, we do not consider the results of Self-RAG on the NQ dataset in our experiments.In all experiments, we use greedy decoding with a temperature setting of 0.

A.2 Error Analysis

We conduct an error analysis to investigate the limitations of our **Self-Selection-RGP** method. With the Mistral-7B as the base LLM, we sample 100 from the errors made by our **Self-Selection-RGP** method on the TriviaQA dataset to conduct our analysis. We categorize the errors into five groups, as shown in Table 6, each with an example: (1) Lack of Evidence (51%): The LLM itself does not have sufficient internal knowledge to answer the question, and the retrieved passages also fail to provide enough information; (2) Partial Matching (20%): The final prediction captures part of the correct answer only; (3) Reasoning Error (14%): The generated explanation(s) contain the answer or the relevant information to infer the answer, but the LLM fails to predict the answer; (4) Selection Error(12%): One of the pairwise predictions is correct, but the model fails to identify the correct one; (5) Formatting Error(3%): The correct answer is

Answer Generation Prompt

Task
Provide a detailed explanation and your best answer to the following question.
Requirements
1. The explanation must provide specific factual evidence and reasoning steps. It must be truthful and cannot include fabricated information. If you don't know the answer or lack information, state 'Answer: unknown'.
2. Strictly follow the output format. Avoid any irrelevant content.
3. The answer must be concise, typically one or a few words . Do not repeat the question or add unnecessary context.
Output Format
Explanation: [Your detailed explanation]
Answer: [Concise answer]
Question
Question

Table 4: The prompt for Answer Generation without retrieval passages.

Answer Generation Prompt

Task
Read the following question and passages, provide detailed explanation and your best answer for the question.
Requirements
1. The explanation must be detailed and include specific factual evidence, references to the provided passages, and clear reasoning steps. It must be truthful and cannot include fabricated information. If you don't know the answer or lack information, state 'Answer: unknown'.
2. Strictly follow the output format. Avoid any irrelevant content.
3. The answer must be concise, typically one or a few words . Do not repeat the question or add unnecessary context.
Output Format
Explanation: [Your detailed explanation]
Answer: [Concise answer]
Question
Question
Passage
Passage
Output
Give your explanation and concise answer for the question:

Table 5: The prompt for Answer Generation with retrieval passages.

included in the prediction but the output format does not follow the instruction, leading to a failed interpretation.

From the table, we make the following observations:

- Over half of the errors are caused by the failure to obtain the knowledge that is necessary for predicting the answer, i.e., Lack of Evidence. This reflects the importance of developing advanced techniques to fetch the relevant information given a question from external knowledge bases to complement LLMs' internal knowledge for producing more accurate results.

- Approximately 39% of errors, including Partial Matching, Reasoning Errors, and Formatting Errors, originate from LLMs' inadequacies in accurately interpreting the human instruction or the relevant knowledge required to infer the precise answer in the correct format. To address this issue, LLMs with enhanced reasoning capabilities are required.

- 12% errors arise from the LLMs' inability to effectively distinguish the correct answer from plausible ones, underscoring the demand for LLMs with enhanced reasoning capabilities.

Table 6: Examples of errors and corresponding percentages. Q, G, and P denote the question, golden answer, and prediction, respectively. C1 and C2 represent two candidate responses.

Lack of Evidence (51%)	Q: The Express, The Telegraph, The Economist, The Times, The Star, and The Independent were all what in 1830's London? (Not Newspapers) G: Stagecoaches C1: Explanation: The Express, The Telegraph...were not entities in London during the 1830s... Answer: Unknown C2: Explanation: ...the passages provided only discuss their histories and developments starting from the mid-1800s... Answer: Unknown P: Explanation: The Express, The Telegraph...were not entities in London during the 1830s... Answer: Unknown
Partial Matching (20%)	Q: In which Parliamentary constituency did Michael Portillo replace Alan Clark as MP in 1999? G: Kensington and Chelsea (borough) C1: Explanation: ...Portillo is most famously known for defeating Clark...in the constituency of Wellingborough. Answer: Wellingborough. C2: Explanation: Michael Portillo replaced Alan Clark as MP in Kensington and Chelsea... Answer: Kensington and Chelsea(borough) P: Explanation: Michael Portillo replaced Alan Clark as MP in Kensington and Chelsea... Answer: Kensington and Chelsea(borough)
Reasoning Error (14%)	Q: Richard Nixon was Vice President to which US state? G: Dwight D Eisenhower C1: Explanation: Richard Nixon served as the Vice President under Dwight D. Eisenhower from 1953 to 1961... However, it's important to note that Nixon didn't directly represent Kansas as Vice President; instead, he assisted President Eisenhower. Answer: Kansas (through Eisenhower) C2: Explanation: Richard Nixon was the Vice President of the United States for the state of California. Answer: California. P: Explanation: Richard Nixon was the Vice President of the United States for the state of California. Answer: California.
Selection Error (12%)	Q: The Greek word praxis refers to what, contrasting with theory? G: Practice C1: Explanation: In ancient Greek philosophy, the term "praxis" refers to the application or practice of... Answer: practice C2: Explanation: The Greek word praxis refers to the process of enacting... Answer: The enactment or realization of a theory P: Explanation: The Greek word praxis refers to the process of enacting... Answer: The enactment or realization of a theory
Formatting Error (3%)	Q: What do 'Barb' charts show? G: Television Ratings C1: Explanation: Bar charts are a type of chart that display... Answer: Bar charts show categorical data... C2: Explanation: ...and other research findings related to television audience measurement and ratings in the United Kingdom. Answer: Barb charts show comparisons... In the context of television ratings... P: Research ... television audience measurement and ratings in the United Kingdom. In the context of television ratings...