RAG-Instruct: Boosting LLMs with Diverse Retrieval-Augmented Instructions

Wanlong Liu[1,2*], **Junying Chen**[2*], **Ke Ji**[2], **Li Zhou**[2], **Wenyu Chen**[1†], **Benyou Wang**[2†]
[1]University of Electronic Science and Technology of China
[2]The Chinese University of Hong Kong, Shenzhen
liuwanlong@std.uestc.edu.cn, cwy@uestc.edu.cn, wangbenyou@cuhk.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) has emerged as a key paradigm for enhancing large language models by incorporating external knowledge. However, current RAG methods exhibit limited capabilities in complex RAG scenarios and suffer from limited task diversity. To address these limitations, we propose **RAG-Instruct**, a general method for synthesizing diverse and high-quality RAG instruction data based on any source corpus. Our approach leverages (1) *five RAG paradigms*, which encompass diverse query-document relationships, and (2) *instruction simulation*, which enhances instruction diversity and quality by utilizing the strengths of existing instruction datasets. Using this method, we construct a 40K instruction dataset from Wikipedia, comprehensively covering diverse RAG scenarios and tasks. Experiments demonstrate that RAG-Instruct effectively enhances LLMs' RAG capabilities, achieving strong zero-shot performance and outperforming various RAG baselines. The code is publicly available at https://github.com/FreedomIntelligence/RAG-Instruct.

1 Introduction

Retrieval-Augmented Generation (RAG) (Guu et al., 2020; Asai et al., 2024b) enhances large language models (LLMs) by integrating external knowledge through document retrieval, effectively reducing hallucinations and improving performance across diverse tasks (Asai et al., 2023; Jin et al., 2024; Lu et al., 2022; Liu et al., 2024a; Zeng et al., 2025; Du et al., 2025a; Wei et al., 2025).

Given the inherent limitations of retrievers (BehnamGhader et al., 2022; Gao et al., 2023), coupled with considerable research showing that noisy retrieval can adversely affect LLM performance (Petroni et al., 2020; Shi et al., 2023; Maekawa et al., 2024; Yang et al., 2025), numerous studies have focused on enhancing the robustness of RAG in handling noisy retrieval contexts. On one hand, some studies involve adaptive retrieval based on query analysis (Asai et al., 2024a; Jeong et al., 2024), or query reformulation (Chan et al., 2024; Ma et al., 2023) to enhance the robustness of LLM-based RAG systems. On the other hand, (Zhang et al., 2024b; Liu et al., 2024b; Yoran et al., 2024) enhance the robustness of models' naive RAG capabilities by training them to adapt to irrelevant and noisy documents.

However, we find existing RAG methods still have limitations: (1) **Limited RAG scenarios**. Real-world RAG scenarios are complex: Given the query, the retrieved information may directly contain the answer, offer partial help, or be helpless. Some answers can be obtained from a single document, while others require multi-hop reasoning across multiple documents. Our preliminary study demonstrates that existing RAG methods exhibit limitations in complex RAG scenarios. (2) **Limited task diversity**. Due to the lack of a general RAG dataset, most current RAG methods (Wei et al., 2024; Zhang et al., 2024b) are fine-tuned on task-specific datasets (e.g., NQ (Kwiatkowski et al., 2019), TrivialQA (Joshi et al., 2017)), which suffer from limited question diversity and data volume.

To address these limitations, we propose **RAG-Instruct**, a general method for synthesizing diverse and high-quality RAG instruction data based on any source corpus. Using this method, we construct a 40K RAG instruction dataset from Wikipedia. Our method emphasizes the **diversity** in two aspects:

1. **Defining diverse RAG paradigms**: we define five RAG query paradigms that encompass various query-document relationships to adapt to different RAG scenarios, considering both document usefulness and the number of useful documents. Based on these modes, we prompt

[*] Equal Contribution.
[†] Corresponding Authors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3865–3888
November 4-9, 2025 ©2025 Association for Computational Linguistics

LLMs to synthesize RAG-specific instructions and responses using external documents.

2. **Enhancing task diversity and data quality**: we incorporate exemplar data from existing instruction datasets, such as SlimOrca (Mitra et al., 2023) and Evol Instruct (Xu et al., 2023a), to guide the generation of RAG instructions. This approach is inspired by recent advancements in synthetic instruction datasets which have two key advantages: (1) high-quality instruction-following responses generated by proprietary LLMs, and (2) diverse instructions that cover a wide range of real-world tasks. We refer to this approach as "*Instruction Simulation*", which leverages the strengths of existing instruction datasets to improve the diversity and quality of the synthesized data.

Our contributions are summarized as follows:

- We introduce **RAG-Instruct**, a general method for synthesizing diverse and high-quality RAG instruction data from any given corpus. Using this method, we construct the RAG-Instruct dataset (based on Wikipedia), the first RAG instruction dataset covering diverse RAG scenarios and tasks.
- We define five *RAG paradigms* to cover diverse query-document relationships and introduce *Instruction Simulation*, a technique that enhances instruction diversity and quality by utilizing the strengths of existing instruction datasets. These techniques ensure the diversity of synthesized datasets across RAG scenarios and tasks.
- Empirical experiments on 11 tasks, including knowledge-intensive QA, multi-step reasoning, and domain-specific benchmarks, demonstrate that RAG-Instruct significantly enhances the model's RAG capabilities. Further experiments demonstrate that the RAG-Instruct outperforms existing RAG datasets and exhibits strong generalization across multiple retrieval sources and retrievers.

2 Preliminary Study

Since retrievers are not perfect, the helpfulness of retrieved documents to the query varies in real-world scenarios. This raises the question: **Can existing RAG methods handle complex and various RAG scenarios?**

Method	TriviaQA (Single-hop)			HotpotQA (Multi-hop)	
	Helpful	Midhelp	Helpless	Helpful	Midhelp
Llama2-7b	71.0	48.0	17.1	51.2	21.2
Llama3-8b	76.4	51.0	20.2	61.4	21.4
Self-RAG (2-7b)	77.3	42.4	14.7	45.1	16.6
RQ-RAG (2-7b)	80.9	52.6	18.7	57.9	24.0
ChatQA-1.5 (3-8b)	83.5	54.9	21.4	65.1	23.9
ChatQA-2.0 (3-8b)	82.4	51.5	20.1	61.4	19.9
RAG-Instruct (3-8b)	**86.9**	**72.6**	**40.5**	**73.1**	**42.2**

Table 1: Preliminary study of limited RAG scenarios. Accuracy (%) is reported. We divided TriviaQA and HotPotQA into multiple subsets. More information for each subset is shown in Appendix C.1

To investigate this, we first define five RAG scenarios based on query-document relationships, which we believe cover the majority of RAG use cases: Single-Doc Answer (helpful), Single-Doc Support (midhelp), Useless Doc (helpless), Multi-Doc Answer (helpful), and Multi-Doc Support (midhelp). Detailed definitions for each scenario are provided in § 3.1.

Next, we evaluate the performance of existing RAG methods across these five scenarios. Using GPT-4o (Achiam et al., 2023), we categorize questions from two question answering (QA) datasets, Single-hop QA (TriviaQA) and Multi-hop QA (HotPotQA (Yang et al., 2018)), into relevant subsets based on the defined RAG scenarios[1]. Detailed prompts for categorization are provided in the Appendix C.1. Then we choose some robust RAG methods, including Self-RAG (Asai et al., 2024a), RQ-RAG (Chan et al., 2024), ChatQA-1.5 and ChatQA-2.0 (Liu et al., 2024b) as baselines to explore their performance across the five RAG scenarios.

As shown in Table 1, existing RAG methods improve primarily in helpful scenarios, while gains in mid-helpful and helpless scenarios are minimal, with some, such as Self-RAG, even underperforming the baseline. **This indicates that existing RAG methods are still unable to handle complex and diverse RAG scenarios effectively.** In comparison, our RAG-Instruct method demonstrates significant improvements across all five scenarios, highlighting its effectiveness and adaptability to complex and diverse RAG scenarios.

Comparision with existing RAG datasets. As shown in Table 2, existing RAG datasets fail to

[1] We choose these datasets for their large number of questions and subsets, which reduces bias.

Dataset	Data Size	RAG Scenarios					Task Diversity	RAG Capability Gains (Δ)			
		r_0	r_1	r_2	r_3	r_4		TQA (acc)	HotpotQA (acc)	ARC (EM)	CFQA (EM)
LongAlpaca (Chen et al., 2023)	12K	✗	✗	✗	✓	✗	✗	1.6 ↑	8.7 ↑	3.9 ↓	1.9 ↓
SQuAD2.0 (Rajpurkar et al., 2018)	130K	✗	✗	✗	✓	✗	✗	1.3 ↑	5.7 ↑	14.1 ↓	5.8 ↓
NarrativeQA (Kočiskỳ et al., 2018)	15K	✗	✗	✗	✓	✗	✗	3.7 ↑	1.0 ↓	5.1 ↓	7.5 ↓
RAG-12000 (Liu et al., 2024b)	12K	✗	✗	✗	✓	✗	✗	5.5 ↑	6.1 ↓	8.7 ↓	1.7 ↓
Self-RAG Data (Asai et al., 2024a)	150K	✓	✗	✗	✓	✗	✗	2.1 ↑	14.2 ↓	6.5 ↑	3.6 ↓
RQ-RAG Data (Chan et al., 2024)	40K	✗	✗	✗	✓	✓	✓	3.2 ↑	4.0 ↑	4.2 ↑	2.0 ↓
RAG-Instruct (Ours)	40K	✓	✓	✓	✓	✓	✓	**6.6** ↑	**12.8** ↑	**9.6** ↑	**4.1** ↑

Table 2: Comparison with three types of non-task-specific RAG datasets: Long-context instruction dataset , reading comprehension datasets , and RAG-specific datasets . r_0 to r_4 represent the five RAG scenario paradigms defined in Table 3. RAG Capability Gains (Δ) refer to the performance difference between models trained on *Llama3.1-8B* using these datasets and *Llama3.1-8B-Instruct*. More details can be found in Table 4 and Table 7.

balance both scenario and task diversity. Long-context instruction datasets and reading comprehension datasets are limited to a narrow range of RAG scenarios, and only show improvements on certain QA tasks, while significantly underperforming on tasks like ARC and CFQA. Additionally, RAG-specific datasets, such as Self-RAG Data and RAG-12000, perform poorly on multi-hop reasoning benchmarks due to the lack of focus on multi-hop scenarios. In contrast, our RAG-Instruct effectively balances both RAG scenario and task diversity, demonstrating superior generalization and robustness.

3 Method

This section outlines the RAG-Instruct process, focusing on constructing diverse and high-quality synthetic RAG datasets. The detailed architecture is illustrated in Figure 1.

3.1 RAG-Instruct

Synthesizing RAG Instructions. Recent proprietary models like GPT-4o (Achiam et al., 2023) have demonstrated remarkable capabilities, and many works (Zheng et al., 2023b; Xu et al., 2023a) based on synthetic datasets have achieved notable success. Therefore, we use GPT-4o to synthesize RAG instructions by leveraging source documents D^*[2] to create context-rich instructions. Specifically, GPT-4o synthesizes an instruction q^* based on D^*, followed by a response a^* referencing D^*, which can be formalized as:

$$(q^*, a^*) = \mathbf{LLM}(D^*). \tag{1}$$

Inspired by work (Zhang et al., 2024b), we introduce documents D^- unrelated to q^*, which serve as additional noise to enhance the robustness. Then our target RAG instruction is as follows.

$$D^*, D^-, q^* \rightarrow a^*.$$

However, RAG instructions generated this way lack diversity in both RAG scenarios and tasks. To address this, we define five **RAG paradigms** and introduce **Instruction Simulation**.

RAG Paradigms. Real-world RAG scenarios are complex: Given the q^*, D^* may directly contain the answer, offer partial help, or be helpless. Some answers can be obtained from a single document in D^*, while others require multi-hop reasoning across multiple documents. To address this, we define RAG paradigms $\mathbb{R}$, where each $r \in \mathbb{R}$ characterizes the relationship between D^* and q^*. As in Table 3, these RAG paradigms consider both document utility and the count of useful documents.

Instruction Simulation. Generating (q^*, a^*) from D^* faces the challenge of instruction monotony. Although q^* is related to D^*, the task, phrasing, and difficulty of the instructions can become repetitive with a similar synthesis prompt. Previous datasets address this by broadly collecting instructions (Izacard et al., 2023) or using self-instruct (Wang et al., 2023b). In our approach, we leverage diverse, high-quality instructions to diversify q^*, a process we term *Instruction Simulation*.

In this process, we use questions from synthetic datasets including ShareGPT (Wang et al., 2023a), Alpaca (hin Cheung and Lam, 2023), WizardLM-70K (Xu et al., 2023a), Lmsys-chat-1M (Zheng et al., 2023a), and SlimOrca (Mitra et al., 2023) as exemplar data. These datasets cover a wide range of tasks, diverse phrasing styles, and varying levels of instruction difficulty. Since RAG is most effective in knowledge-intensive task scenarios (Maekawa et al., 2024; Shi et al., 2023), we use GPT-4o to filter knowledge-intensive instructions

[2] We will explain how D^* are obtained in the following *Instruction Simulation* section.

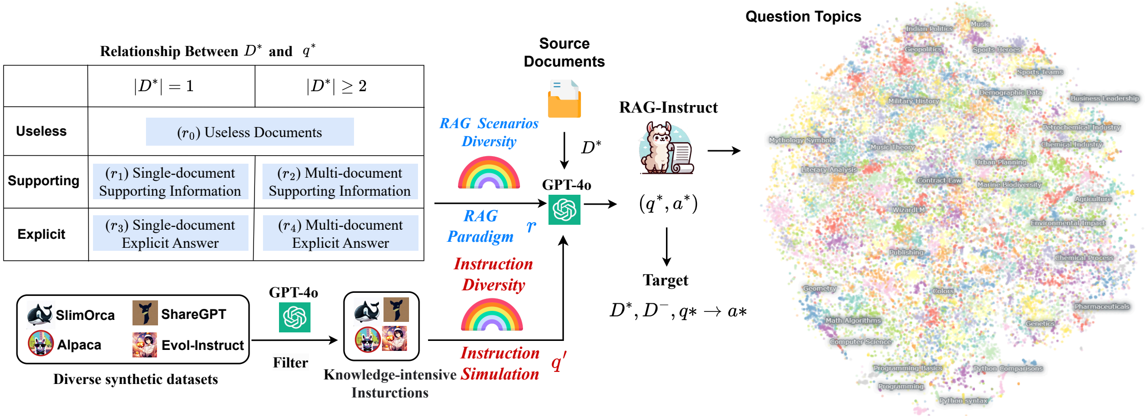


Figure 1: The process of synthesizing data with RAG-Instruct involves ensuring instruction data diversity through five RAG paradigms and Instruction Simulation. The visualization of the question topic is generated using Atlas.

D^*-q^* **Relationship**	**Usefulness of** D^*	$\lvert D^*\rvert$	**Relationship Description**
(r_0) Useless Doc	Useless	1	D^* offers no help in answering q^*, even if related.
(r_1) Single-Doc Support	Supporting	1	One doc ($\lvert D^*\rvert = 1$) aids $q*$, providing supporting info or clues without explicit answers.
(r_2) Multi-Doc Support	Supporting	≥ 2	Multiple documents ($\lvert D^*\rvert \geq 2$) support $q*$ by providing clues or supporting information without explicitly answering it, requiring integration (multi-hop reasoning).
(r_3) Single-Doc Answer	Explicit	1	One doc ($\lvert D^*\rvert = 1$) directly provides the answer a^* to q^*.
(r_4) Multi-Doc Answer	Explicit	≥ 2	Multiple docs ($\lvert D^*\rvert \geq 2$) provide a full answer to q^*, requiring integration (multi-hop reasoning).

Table 3: Detailed descriptions of our defined **five RAG paradigms**. See Appendix C.2 for specific prompts.

from these synthetic datasets (details of the prompt are provided in Appendix A.1).

Then for each synthesis, an instruction $q' \in Q$ is randomly sampled for simulation. Given a corpus D containing multiple documents $d \in D$, the source documents $D^* \subset D$ are retrieved based on q'. Subsequently, (q^*, a^*) can be synthesized as follows:

$$(q^*, a^*) = \mathbf{LLM}(D^*, q', r), \tag{2}$$

where r denotes the sampled RAG paradigm, and the synthesis prompt is illustrated in Figure 3. Here, D^* controls the topic of q^*, while q' shapes its format and task requirements.

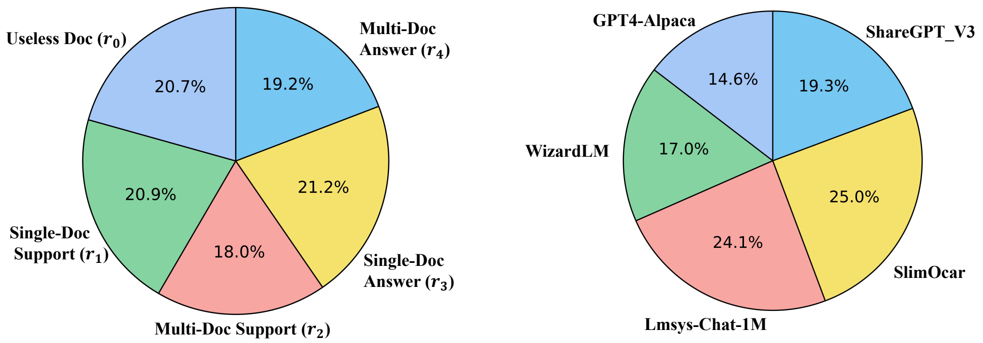
3.2 Dataset Construction

We construct RAG-Instruct using Wikipedia corpus. For each synthesis, we sample an RAG paradigm r, a simulated instruction q', and retrieved source documents D^* to generate (q^*, a^*) using GPT-4o. To incorporate unrelated documents D^-, we randomly sample documents retrieved based on q^* and ranked beyond the top 200 as D^-. Additionally, for cases where $|D^*| \geq 2$, we ensure that the number of source documents is fewer than 5. Subsequently, $D^*, D^-, q^* \to a^*$ is set as the training objective to form RAG-Instruct. In total, we build a dataset of 53K instructions, with the distributions of RAG paradigms and simulated instructions illustrated in Figure 2. More dataset construction details are shown in Appendix A.1.

4 Experiments

4.1 Experimental Settings

Evaluation Tasks. We conduct evaluations of our RAG-Instruct and various baselines across 10 tasks in four major categories: (1) **Open-Ended Tasks**, including WebQA (WQA) (Berant et al., 2013), PopQA (PQA) (Mallen et al., 2023), and TriviaQA-unfiltered (TQA) (Joshi et al., 2017), where models answer open-domain factual questions with accuracy as the metric. (2) **Closed-Set Tasks**, including OpenbookQA (OBQA) (Mihaylov et al., 2018), PubHealth (Pub) (Zhang et al., 2023) and ARC-Challenge (ARC) (Clark et al.,



(a) Distributions of RAG Paradigms

(b) Distributions of Data Sources

Figure 2: The detailed distributions of 5 RAG paradigms and simulated instruction data sources.

<Documents>
[1] **`{<document 1>}`**
[2] **`{<document 2>}`**
[3] ...
</Documents>

Your task is to generate an English question q* and a corresponding response a* based on the provided <Documents>. Please note that the question q* can take various forms, not limited to questions with a question mark, but also including statements, instructions, and other formats. You need to follow the requirements below to generate the q* and a* (RAG Paradigms):
1. The answer to q* can be derived from multiple documents within <Documents>, involving multi-hop reasoning or the integration of information from several documents.
2. a* should leverage the information in <Documents> to provide an accurate answer to q*, ensuring that the response is accurate, detailed, and comprehensive.

Additionally, to ensure diversity, richness, and high quality in the question q* you generate, we will randomly provide a question for you to emulate. In other words, while satisfying the requirements above, make q* similar in task requirement and expression to the <Simulated Instruction> below:
<Simulated Instruction>
`{<Simulated Instruction>}`
</Simulated Instruction>

Please directly generate the question-answer pair (q*, a*) following all the rules above in the format of {"q*": ..., "a*": ...}. Ensure the quality of the generated (q*, a*).

Figure 3: The prompt of RAG-Instruct. **`<document>`** and **`<Simulated Instruction>`** represent input variables for the document and simulated instruction, respectively. (Blue text) indicates RAG Paradigms, illustrating the prompt for r_4; other paradigms are shown in Appendix C.2. (Red text) represents Instruction Simulation.

2018), involving multiple-choice QA with Extract Match (EM) as the metric. (3) **Multi-Hop Tasks**, including 2WikiMultiHopQA (2WIKI) (Ho et al., 2020), HotpotQA (HotQ) (Yang et al., 2018), and Musique (MSQ) (Trivedi et al., 2022), requiring multi-hop reasoning with accuracy as the metric. (4) **Domain-Specific Tasks**, CFQA (Chen et al., 2022) in the financial domain and PubMedQA (Jin et al., 2019) in the medical domain. We also include the long-form QA evaluation in Appendix B.1. We perform zero-shot evaluations throughout these experiments, providing task instructions without few-shot demonstrations. Reasoning details and prompts are provided in Appendix A.2.

Baselines. We compare our method against a diverse set of baselines, grouped into two main categories: (1) **Closed-Source LLMs without RAG**, including GPT-4o and GPT-4o-mini. We test them using OpenAI's official APIs. (2) **Open-source instruction-turned baselines with RAG**, such as Llama3.1-8b-Instruct (Dubey et al., 2024), Llama3.1-70B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2024) and Qwen2.5-72B-Instruct. (3) **RAG-specific baselines**, including Self-RAG, RQ-RAG, InstructRAG (Wei et al., 2024) and ChatQA. For these methods, we evaluate

	Open-ended			Closed-set			Multi-hop			Domain-specific		
	WQA	PQA	TQA	OBQA	Pub	ARC	2WIKI	HotP	MSQ	CFQA	PubMed	AVG
	(acc)	(acc)	(acc)	(EM)	(EM)	(EM)	(acc)	(acc)	(acc)	(EM)	(EM)	
Closed-Source LLMs without RAG												
GPT-4o	72.5	71.3	84.4	88.6	87.7	88.0	88.0	54.6	31.4	63.0	77.0	73.4
GPT-4o-mini	69.5	69.2	82.2	89.6	87.0	84.1	74.4	54.5	30.8	60.7	73.0	70.4
$\sim$ ***8B Open-Source LLMs with RAG***												
Llama-3.1-8B-Instruct	59.5	60.8	71.4	77.2	56.8	70.3	66.8	45.5	18.7	53.7	73.6	54.5
Qwen2.5-7B-Instruct	64.1	62.0	75.6	74.2	74.2	75.7	66.5	49.5	20.8	58.7	62.6	58.4
RQ-RAG (Llama2-7B)	56.5	57.1	70.2	80.6	71.8	68.3	53.7	43.1	18.2	21.9	55.6	52.6
Self-RAG (Llama2-7B)	49.0	55.8	69.3	78.0	72.4	73.1	48.4	35.8	11.5	21.5	49.8	50.4
InstructRAG (Llama3-8B)	63.2	66.2	78.5	73.1	71.8	66.3	69.2	55.2	30.1	27.9	60.3	60.2
ChatQA-2.0 (Llama3-8B)	50.5	58.3	72.5	72.6	75.8	65.6	59.0	42.3	16.1	51.8	61.3	54.7
Llama-2-7B + **RAG-Instruct**	67.2	62.4	77.4	71.4	75.9	74.8	68.1	53.5	21.8	29.7	71.2	60.3
Qwen2.5-7B + **RAG-Instruct**	66.1	63.7	78.1	78.4	76.4	78.0	74.8	54.6	27.7	**59.7**	72.7	64.5
Llama-3.1-8B + **RAG-Instruct**	**69.7**	**68.4**	**80.0**	**84.8**	**77.2**	**79.9**	**79.3**	**56.4**	**33.7**	57.8	**77.0**	**66.8**
$>$ ***10B Open-Source LLMs with RAG***												
Llama-3.1-70B-Instruct	64.9	63.3	75.4	85.0	75.4	84.7	73.5	47.5	26.6	59.1	77.2	63.9
Qwen2.5-72B-Instruct	68.8	68.7	81.5	83.0	78.0	80.2	81.1	56.6	35.5	66.8	80.0	68.8
Llama-3.1-70B + **RAG-Instruct**	**73.6**	**70.4**	83.8	88.6	**82.8**	**85.1**	83.1	62.9	40.1	62.1	79.7	72.0
Qwen2.5-72B + **RAG-Instruct**	72.4	70.3	**85.0**	**89.3**	78.5	82.1	**88.3**	**63.9**	**42.0**	**69.2**	**82.0**	**73.7**

Table 4: Zero-shot performance of different instruction datasets on RAG Benchmarks. **Bold** and underline indicate the best and second-best experimental results within each section. The datasets were fine-tuned using identical hyperparameters.

using publicly released model weights and prompts provided by their respective works.

Training settings. We train our model using the RAG-Instruct dataset (wikipedia), which features diverse instruction-following input-output pairs. During the dataset construction, we employ the off-the-shelf Contriever-MS MARCO (Izacard et al.) as the retriever. For each data entry, we ensure the use of all source documents D^* and supplement them with enough unrelated documents D^- to total 10 documents. Additional training details are provided in Appendix A.1.

Inference settings. We use VLLM (Kwon et al., 2023) for memory-efficient inference and adopt a greedy decoding strategy for model generation. For evaluation benchmarks, we utilize Wikipedia as the retrieval corpus and use the Contriever retriever for document retrieval. More detailed inference specifications can be found in Appendix A.2.

4.2 RAG Capability Gains

Comparison against closed-source LLMs. As shown in Table 4, compared to powerful proprietary models like GPT-4o and GPT-4o-mini, our RAG-Instruct, trained on base 8B models, matches or even outperforms them on several tasks, including open-ended tasks (PQA and TQA), multi-hop tasks (HotQA and MSQ), and domain-specific tasks (PubMedQA). This demonstrates that our RAG-Instruct significantly enhances the model's RAG capabilities.

Comparison against RAG-specific models. As shown in Table 4, RAG-specific models such as Self-RAG, and RQ-RAG show significant improvements over the base models on open-ended and closed-set tasks. However, they underperform compared to the base models on domain-specific and multi-hop tasks. In contrast, our RAG-Instruct achieves significant improvements across all four categories of tasks compared to the base models and outperforms all previous SOTA RAG-specific models, particularly in multi-hop and domain-specific tasks. This highlights its superior robustness and generalization across a broader range of RAG scenarios and tasks.

Comparison against Open-source instruction-tuned models. We also compare our method with open-source instruction-tuned models, which exhibit strong RAG capabilities. As shown in Table 4, models trained with RAG-Instruct on base models outperform these instruction-tuned models across various tasks, demonstrating that the RAG instruction dataset effectively enhances the model's RAG performance.

4.3 Impact of Instruction Simulation

To investigate the impact of *Instruction Simulation*, we design a comparative experiment. We randomly

	TQA	ARC	HotP
RAG-Instruct$_{20k}$ (Llama3.1-8B)	**77.0**	**79.4**	**53.1**
w.o. Simulation$_{20k}$	75.9	70.4	47.7
Llama3.1-8B-Instruct *w.o. Retrieval*	63.1	64.1	33.9
RAG-Instruct *w.o. Retrieval*	63.2	62.8	33.4

Table 5: Ablation Study (**only 20k data used**) on RAG-Instruct. *w.o. Simulation* indicates the removal of the *Instruction Simulation* process, while *w.o. Retrieval* indicates the performance in non-retrieval scenarios. Complete ablation results are in shown Appendix B.2

Method	**TriviaQA (Single)**			**HotpotQA (Multi)**	
	Helpful	Midhelp	Helpless	Helpful	Midhelp
RAG-Instruct	**86.9**	**72.6**	**40.5**	**73.1**	**42.2**
w.o. r_0	86.4	69.6	36.4^{-}	73.1	39.3
w.o. r_1	86.5	66.5^{-}	40.9	72.4	41.3
w.o. r_2	86.2	71.8	39.7	68.2	29.8^{-}
w.o. r_3	83.5^{-}	70.6	39.6	72.8	42.2
w.o. r_4	85.2	72.1	39.5	65.4^{-}	38.8

Table 6: Ablation study on role of query paradigms. All experiments are conducted based on the *Llama3.1-8B* model using identical hyperparameters. '–' indicates large performance drops for each paradigm.

sample a subset D_s containing 20,000 entries from our RAG-Instruct dataset and create another subset D'_s without using *Instruction Simulation*. To ensure a fair comparison, D_s and D'_s share the same source documents D^* and include all five RAG scenario paradigms. We then train two models on Llama3.1-8B using D_s and D'_s with identical hyperparameters.

As shown in Table 5, removing the *Instruction Simulation* process results in performance declines across all tasks. The drop is smaller for open-ended tasks (TQA) but significantly larger for closed-set (ARC), multi-hop (HotP) tasks. We observe that without *Instruction Simulation*, GPT-4o tends to generate overly simple and uniform questions, resembling open-ended ones, leading to minimal impact on closed-set evaluation. However, the diverse formats of closed-set, multi-hop, and domain-specific tasks, such as multiple-choice and multi-hop reasoning, pose challenges that the model struggles to handle. This highlights the critical role of *Instruction Simulation* in enabling the model to adapt to a wide variety of tasks.

Furthermore, we provide specific cases in Appendix B.5, demonstrating that *Instruction Simulation* generates questions that closely resemble exemplar questions, significantly enhancing diversity compared to those produced without it.

4.4 Role of RAG Paradigms

To evaluate the role of RAG paradigms, we design an ablation experiment to verify the effectiveness of the five RAG scenarios in RAG-Instruct. Specifically, we remove the data corresponding to each paradigm from RAG-Instruct one at a time and train models on Llama3.1-8B using identical training hyperparameters, respectively.

As shown in Table 6, when a single RAG paradigm (e.g. r_0) is removed from RAG-Instruct, we observe a noticeable performance drop in evaluation benchmarks corresponding to that specific RAG scenario. This indicates that each RAG paradigm plays a critical role in enhancing the model's RAG capabilities.

5 Further Analysis

5.1 What advantages does RAG-Instruct have over existing instruction datasets?

To explore whether existing instruction datasets are sufficient for RAG scenarios, we evaluate models fine-tuned on four common instruction datasets and three context-enhenced datasets using LLaMA-3.1-8B. Results are shown in Table 7 and our findings are as follows:

Take-away 1. *Rich context datasets (e.g., long-context instruction dataset LongAlpaca and reading comprehension dataset SQuAD2.0) improve RAG capabilities more effectively than those with shorter context lengths (e.g., Wizardlm and Aplaca).*

Take-away 2. *Traditional instruction datasets fail to effectively enhance models' RAG capabilities, significantly lagging behind the official instruction-tuned models, while RAG-Instruct can significantly improve RAG performance.*

5.2 Does fine-tuning with RAG-Instruct affect model's general capabilities?

To explore whether fine-tuning with RAG-Instruct affects model's general capabilities, we evaluate the fine-tuned model (on Llama3.1-8B) in non-RAG scenarios. As shown in Table 5, RAG-Instruct$_{w.o.\ Retrieval}$ performs on bar with Llama-3.1-8B-Instruct in non-RAG scenarios, without significant performance degradation. This demonstrates that RAG-Instruct enhances the model's RAG capabilities while also improving its general instruction-following abilities. We assume that RAG-Instruct are inherently based on general instruction datasets, which inherit the advantages

	Open-ended			Closed-set		Multi-hop			Domain-specific		
	WQA	PQA	TQA	OBQA	ARC	2WIKI	HotP	MSQ	CFQA	PubMed	AVG
Proprietary instruction-tuned LLaMA											
Llama-3.1-8B-Instruct	59.5	60.8	73.4	77.2	70.3	66.8	45.5	18.7	53.7	73.9	60.0
Llama-3.1-8B (Base)											
Fine-tuning with Traditional Instruction Datasets											
+ Evol-Instruct (70K)	54.6	54.2^{-}	71.5	73.4^{-}	63.1^{-}	50.9^{-}	41.1	14.7	38.7^{-}	53.5^{-}	51.6^{-}
+ ShareGPT (94K)	60.9	54.9^{-}	72.8	67.2^{-}	52.9^{-}	59.0^{-}	43.9	14.3	40.3^{-}	67.2^{-}	52.4^{-}
+ Alpaca (52K)	53.1^{-}	56.4	72.3	65.6^{-}	60.6^{-}	57.7^{-}	41.3	13.4^{-}	34.8^{-}	36.5^{-}	49.2^{-}
+ SlimOrca (518K)	55.3	60.0	69.1	82.4	62.7^{-}	54.7^{-}	40.2^{-}	15.5	33.1^{-}	66.9^{-}	54.0^{-}
Fine-tuning with Context-Enhanced Datasets											
+ LongAlpaca (12K)	63.9	56.0	75.0	75.2	66.4	72.9^{+}	54.2^{+}	27.7^{+}	51.8	65.7^{-}	60.9
+ SQuAD2.0 (130K)	61.5	57.2	72.1	59.8^{-}	56.2^{-}	65.7	51.2^{+}	23.7^{+}	47.9^{-}	51.6^{-}	54.7^{-}
+ NarrativeQA (12K)	61.2	57.0	77.1	67.8^{-}	65.2^{-}	52.0^{-}	44.5	17.2	46.2^{-}	68.7^{-}	55.6
Fine-tuning with RAG Instructions											
+ RAG-Instruct (40K)	69.7^{+}	68.4^{+}	80.0^{+}	84.8^{+}	79.9^{+}	79.3^{+}	56.4^{+}	33.7^{+}	57.8	77.0	68.6^{+}

Table 7: Zero-shot performance of different instruction datasets using RAG. Using *Llama-3.1-8B-Instruct* as the pivot, '+' indicates a >5-point improvement, while '–' indicates a >5-point drop. All datasets were fine-tuned with identical hyperparameters. See Section 4.1 for evaluation details.

of these datasets without compromising general capabilities. Additionally, we evaluate our model on MMLU and MMLU-Pro (in Appendix B.6), which further demonstrates that RAG-Instruct does not impair the model's general capabilities.

Take-away 3. *RAG-Instruct dataset enhances RAG capabilities without compromising the model's general capabilities.*

5.3 How does RAG-Instruct perform with other retrieval sources and retrievers?

To further explore the generalization of our method, we investigate the impact of using different retrieval sources. Specifically, we further evaluate our method on four single-hop QA tasks, including ARC, PQA, TQA and OBQA, utilizing DuckDuckGo, and Bing Search as retrieval sources during inference. The results (detailed in Appendix B.3.) suggest that all retrieval sources effectively improve task performance, with minimal variation in performance across different sources. Additionally, we also explore the performance on the BM25 retriever (Robertson et al., 2009). The detailed results can be found in Appendix B.4. These results demonstrate the robustness of RAG-Instruct across different retrieval sources and retrievers.

5.4 Does the performance improvement stem from enhanced RAG capabilities rather than knowledge injection?

Since our RAG-Instruct is built on the Wikipedia corpus, the performance improvements on evaluation benchmarks may stem from knowledge injection during the supervised fine-tuning stage. To investigate whether our approach genuinely enhances the model's RAG capabilities, we compare the performance in both retrieval and non-retrieval scenarios (based on the Llama3.1-8B model trained on RAG-Instruct). As shown in Table 5, performance in non-retrieval scenarios is significantly lower across all benchmarks compared to retrieval scenarios. This demonstrates that RAG-Instruct indeed effectively enhances the model's capabilities in RAG scenarios rather than knowledge injection.

6 Related Work

6.1 Retrieval-Augmented Generation

Large language models (LLMs) are advancing at a rapid pace across numerous fields and applications (Zhang et al., 2025d,b; Tan et al., 2025; Feng and Peng, 2014; Zhang et al., 2025f; Wu et al., 2025; Guo et al., 2025; Leong and Wu, 2024; Zhang et al., 2025a). Retrieval-augmented generation (RAG) is a widely adopted approach for supplementing the parametric knowledge of LLMs with external information sources (Fang et al., 2025; Liu et al., 2025a; Lu et al., 2025; Liu et al., 2025b). Due to the imperfections of retrievers, the retrieved information often fails to align well with the LLM's needs, which can negatively impact LLM performance (Petroni et al., 2020; Shi et al., 2023; Maekawa et al., 2024).

To enhance LLM-based RAG capabilities, some studies focus on aligning retrievers with LLM needs (Shi et al., 2024; Lin et al., 2023) through multi-step retrieval processes (Trivedi et al., 2023; Jiang et al., 2023; Jeong et al., 2024; Shao et al., 2023; Yu et al., 2023; Asai et al., 2024a; Du et al., 2025b,b) and query reformulation (Ma et al., 2023; Jeong et al., 2024). On the other hand, several studies focus on enhancing the RAG capabilities of LLMs by improving their robustness in noisy retrieval contexts. Research such as (Chan et al., 2024; Zhang et al., 2024b; Liu et al., 2024b; Yoran et al., 2024) trains models with additional irrelevant or noisy documents to better handle such scenarios. However, these approaches consider only a limited range of RAG scenarios. Furthermore, the lack of a general RAG dataset forces many works, such as RAFT (Zhang et al., 2024b), to fine-tune models on task-specific datasets, leading to poor task generalization. This highlights the need for a dataset that covers diverse RAG scenarios and tasks.

6.2 Instruction Data

With the development of LLMs (Zhao et al., 2023; Zhang et al., 2024a; Leong et al., 2024; Zhang et al., 2025e,c), the creation of instruction datasets has been instrumental in enhancing the instruction-following and generalization capabilities of these models. Early initiatives, such as (Mishra et al., 2022), introduced task-specific instructions to guide model behavior. Subsequent efforts, including Super-NaturalInstructions (Wang et al., 2022) and Unnatural Instructions (Honovich et al., 2022), expanded the diversity and complexity of these instructions. These datasets enabled LLMs like Alpaca (Taori et al., 2023) and Dolly (Conover et al., 2023) to better align with human intent through fine-tuning on structured instruction-output pairs, fostering adaptability to unseen tasks through varied instruction formulations. Recent studies, such as WizardLM (Xu et al., 2023b) and ShareGPT (OpenAI, 2023), have further enhanced the generalization and richness of instruction datasets, significantly contributing to the robust generalization capabilities of LLMs. Therefore, RAG-Instruct inherits multiple high-quality and rich instruction datasets, leveraging their advantages.

7 Conclusion

This work introduces RAG-Instruct, a method for synthesizing diverse and high-quality RAG instruction data from any source corpus. It incorporates five RAG paradigms to capture diverse query-document relationships and uses instruction simulation to enhance data quality and diversity by leveraging existing datasets. Using this approach, we construct a 40K instruction dataset from Wikipedia, covering diverse RAG scenarios and tasks. For future work, we plan to expand the instructions in RAG-Instruct to incorporate chain-of-thought (CoT) characteristics, enabling models to perform planned retrieval based on the query.

Limitations

Granularity of RAG Paradigms While RAG-Instruct introduces five distinct RAG query paradigms to handle various query-document relationships, this relationship is of a coarse granularity. Specifically, the current set of paradigms focuses on broad categories but does not explore more granular or specialized paradigms that could better capture nuanced retrieval tasks. For instance, for multi-hop queries, the number of hops could be specified, and relevance might have more granular options. Expanding the range of RAG paradigms to cover finer distinctions could enhance the model's ability to handle complex, diverse, and edge-case retrieval situations, thereby improving its robustness and performance.

Reliance on Synthetic Data Our approach relies on synthetic data generation, which inherently carries the risk of introducing errors or biases, even when using powerful large language models like GPT-4. While the use of large-scale instruction datasets such as SlimOrca and Evol Instruct improves the diversity and quality of the generated data, it is still possible for GPT-4 to produce flawed or inconsistent RAG instructions that may negatively impact downstream tasks. As synthetic data generation becomes more prevalent, ensuring the accuracy and reliability of such data remains an ongoing challenge, especially in high-stakes domains where the correctness of information is critical.

Acknowledgment

This work was supported by Major Frontier Exploration Program (Grant No. C10120250085) from the Shenzhen Medical Academy of Research and Translation (SMART), the Shenzhen Science and Technology Program (JCYJ20220818103001002), NSFC grant 72495131, Shenzhen Doctoral Startup Funding (RCBS20221008093330065), Tianyuan Fund for Mathematics of National

Natural Science Foundation of China (NSFC) (12326608), Shenzhen Science and Technology Program (Shenzhen Key Laboratory Grant No. ZDSYS20230626091302006), Shenzhen Stability Science Program 2023 and the National Natural Science Foundation of China under grant U22B2061.

References

Achiam et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

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

Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen-tau Yih. 2024b. Reliable, adaptable, and attributable language models with retrieval. *arXiv preprint arXiv:2403.03187*.

Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2022. Can retriever-augmented language models reason? the blame game between the retriever and the language model. *arXiv preprint arXiv:2212.09146*.

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.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

Yukang Chen, Shaozuo Yu, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Long alpaca: Long-context instruction-following models. `https://github.com/dvlab-research/LongLoRA`.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292.

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

Jacob Conover et al. 2023. Dolly: A 12b-parameter model for instruction following. ArXiv preprint arXiv:2303.11366.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Guanting Dong, Xiaoshuai Song, Yutao Zhu, Runqi Qiao, Zhicheng Dou, and Ji-Rong Wen. 2024. Toward general instruction-following alignment for retrieval-augmented generation. *arXiv preprint arXiv:2410.09584*.

Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. 2025a. Graphmaster: Automated graph synthesis via llm agents in data-limited environments. ArXiv preprint arXiv:2504.00711v2.

Enjun Du, Siyi Liu, and Yongqi Zhang. 2025b. Mixture of length and pruning experts for knowledge graphs reasoning. ArXiv preprint arXiv:2507.20498.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. *arXiv preprint arXiv:2503.10720*.

Pengbin Feng and Xuhui Peng. 2014. A note on monge–kantorovich problem. *Statistics & Probability Letters*, 84:204–211.

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

Jiani Guo, Zuchao Li, Jie Wu, Qianren Wang, Yun Li, Lefei Zhang, hai zhao, and Yujiu Yang. 2025. Tom: Leveraging tree-oriented mapreduce for long-context reasoning in large language models. In *The 2025 Conference on Empirical Methods in Natural Language Processing*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Tsun hin Cheung and Kin Man Lam. 2023. Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking. *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 846–853.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Or Honovich et al. 2022. Unnatural instructions: Tuning language models with synthetic data. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5021–5035. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C 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 7029–7043.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *arXiv preprint arXiv:2405.13576*.

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

Mandar Joshi, Eunsol Choi, Daniel S 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.

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)*, pages 6769–6781.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

H.Y. Leong, Y. Gao, S. Ji, Y. Zhang, and U. Pamuksuz. 2024. Efficient fine-tuning of large language models for automated medical documentation. In *2024 4th IEEE International Conference on Digital Society and Intelligent Systems (DSInS)*, Sydney, Australia.

H.Y. Leong and Y. Wu. 2024. Why should next-gen llm multi-agent systems move beyond fixed architectures to dynamic, input-driven graphs? *SSRN Electronic Journal*.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*.

Aofan Liu, SONG Shiyuan, Cehao Yang, and Yiyan Qi. 2025a. Beyond function-level search: Repository-aware dual-encoder code retrieval with adversarial verification. In *Knowledgeable Foundation Models at ACL 2025*.

Aofan Liu, Yuguo Yin, Hongjian Xing, Zhen Li, and Yiyan Qi. 2025b. Md3r: Minimizing data distribution discrepancies to tackle inconsistencies in multilingual query-code retrieval. In *Knowledgeable Foundation Models at ACL 2025*.

Wanlong Liu, Enqi Zhang, Li Zhou, Dingyi Zeng, Shaohuan Cheng, Chen Zhang, Malu Zhang, and Wenyu Chen. 2024a. A compressive memory-based retrieval approach for event argument extraction. *arXiv preprint arXiv:2409.09322*.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Chatqa:

Building gpt-4 level conversational qa models. *arXiv preprint arXiv:2401.10225*.

Shuai Lu, Nan Duan, Hojae Han, Daya Guo, Seung-won Hwang, and Alexey Svyatkovskiy. 2022. Reacc: A retrieval-augmented code completion framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6227–6240.

Xuan Lu, Sifan Liu, Bochao Yin, Yongqi Li, Xinghao Chen, Hui Su, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2025. Multiconir: Towards multi-condition information retrieval. *Preprint*, arXiv:2503.08046.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Seiji Maekawa, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. 2024. Retrieval helps or hurts? a deeper dive into the efficacy of retrieval augmentation to 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 5506–5521.

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.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Swaroop Mishra et al. 2022. Natural instructions: Benchmarking generalization in instruction following. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5021–5035. Association for Computational Linguistics.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

OpenAI. 2023. Sharegpt: A large-scale instruction-following dataset. Https://openai.com/research/sharegpt.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

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)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

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 8364–8377.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.

Wenting Tan, Dongxiao Chen, Jieting Xue, Zihao Wang, and Taijie Chen. 2025. Teaching-inspired integrated prompting framework: A novel approach for enhancing reasoning in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 827–839.

Rishi Taori et al. 2023. Alpaca: A 175b-parameter model for instruction following. ArXiv preprint arXiv:2303.11366.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition.

Transactions of the Association for Computational Linguistics, 10:539–554.

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.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023a. How far can camels go? exploring the state of instruction tuning on open resources. *ArXiv*, abs/2306.04751.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Yizhong Wang et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109. Association for Computational Linguistics.

Jiyao Wei, Saiping Guan, Da Li, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025. A survey of link prediction in n-ary knowledge graphs. *arXiv preprint arXiv:2506.08970*.

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

Jie Wu, Haoling Li, Xin Zhang, Jianwen Luo, Yangyu Huang, Ruihang Chu, Yujiu Yang, and Scarlett Li. 2025. Teaching your models to understand code via focal preference alignment. *Preprint*, arXiv:2503.02783.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *ArXiv*, abs/2304.12244.

Yizhou Xu et al. 2023b. Wizardlm: Empowering large language models to follow complex instructions. ArXiv preprint arXiv:2304.12244.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Wang, and Liangming Pan. 2025. How is llm reasoning distracted by irrelevant context? an analysis using a controlled benchmark. *Preprint*, arXiv:2505.18761.

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.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

W. Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *ArXiv*, abs/2311.09210.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. 2025. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*.

H. Zhang, B. Huang, Z. Li, X. Xiao, H.Y. Leong, Z. Zhang, X. Long, T. Wang, and H. Xu. 2025a. Sensitivity-lora: Low-load sensitivity-based fine-tuning for large language models. In *Findings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jusheng Zhang, Kaitong Cai, Yijia Fan, Jian Wang, and Keze Wang. 2025b. Cf-vlm:counterfactual vision-language fine-tuning. *Preprint*, arXiv:2506.17267.

Jusheng Zhang, Yijia Fan, Wenjun Lin, Ruiqi Chen, Haoyi Jiang, Wenhao Chai, Jian Wang, and Keze Wang. 2025c. Gam-agent: Game-theoretic and uncertainty-aware collaboration for complex visual reasoning. *arXiv preprint arXiv:2505.23399*.

Jusheng Zhang, Zimeng Huang, Yijia Fan, Ningyuan Liu, Mingyan Li, Zhuojie Yang, Jiawei Yao, Jian Wang, and Keze Wang. 2025d. KABB: Knowledge-aware bayesian bandits for dynamic expert coordination in multi-agent systems. In *Forty-second International Conference on Machine Learning*.

Le Zhang, Bo Wang, Xipeng Qiu, Siva Reddy, and Aishwarya Agrawal. 2025e. Rearank: Reasoning re-ranking agent via reinforcement learning. *arXiv preprint arXiv:2505.20046*.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024a. Exploring the best practices of query expansion with large language models. *arXiv preprint arXiv:2401.06311*.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Interpretable unified language checking. *arXiv preprint arXiv:2304.03728*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024b. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Zhilin Zhang, Naveed Ahmed Saleem Janvekar, Pengbin Feng, and Nitika BHASKAR. 2025f. Graph-based detection of abusive computational nodes. US Patent 12,223,056.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, et al. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*.

Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. 2023b. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865.

A Experimental Details

A.1 More Details of RAG-Instruct Dataset

Dataset Construction. Our RAG-Instruct corpus is built using Wikipedia. Following the approach (Karpukhin et al., 2020), each document is a disjoint text block of up to 100 words extracted from a Wikipedia article. Following work (Shi et al., 2023), we generate Wikipedia document embeddings.

For exemplar data, we select datasets such as ShareGPT (Wang et al., 2023a), Alpaca (hin Cheung and Lam, 2023), WizardLM-70K (Xu et al., 2023a), Lmsys-chat-1M (Zheng et al., 2023a), and SlimOrca (Mitra et al., 2023). First, we remove overly short, overly long, and low-quality data from these datasets. Then, we randomly sample 120K questions from the filtered data. Since RAG is most effective in knowledge-intensive task scenarios (Maekawa et al., 2024; Shi et al., 2023), we use GPT-4o to further filter for knowledge-intensive instructions from these synthetic datasets. The specific prompt used is shown in Figure 5.

Detailed Statistics of RAG-Instruct Dataset. We have included detailed statistics for the RAG-Instruct dataset, including the number of questions, average question lengths, average answer length, average number of source documents, data source distribution, and RAG scenario distribution. These are presented in the Table 8.

Additionally, we report the API cost in constructing RAG-Instruct, including the GPU hours used for training and evaluation in Table 9.

A.2 More Details of Training and Inference

Training Details. We train our models using 8 Nvidia A800 GPUs, each with 80GB of memory. All models are trained for 3 epochs with a total batch size of 128, a peak learning rate of 5e-6, 3% warmup steps, and linear weight decay. The maximum token length is set to 4096 for all models. We leverage DeepSpeed Stage 3 (Rajbhandari et al., 2020) for multi-GPU distributed training with BFloat16 precision enabled. FlashAttention (Dao et al., 2022) is employed to improve efficiency during long-context training.

Inference Details. We conduct evaluations of our RAG-Instruct and various baselines across a wide range of downstream tasks, covering 11 tasks in four major categories. Throughout these experiments, we perform zero-shot evaluations, providing task instructions without few-shot demonstrations. For RAG-specific models, we follow the original papers' weights and prompts for inference. For our model and other baselines, reasoning details and prompts are provided in Table 18.

Open-Ended Tasks include three open-domain question-answering datasets, WebQA (WQA) (Berant et al., 2013), PopQA (PQA) (Mallen et al., 2023), and TriviaQA-unfiltered (TQA) (Joshi et al., 2017), where models are required to answer arbitrary questions based on factual knowledge. We retrieve the top 10 most relevant documents from the corpus as candidate documents. Following (Asai et al., 2024a), we evaluate the performance based on accuracy, assessing whether gold answers are included in the model output.

Closed-Set Tasks include two multiple-choice question-answering datasets: OpenbookQA (OBQA) (Mihaylov et al., 2018), PubHealth (Pub) (Zhang et al., 2023) and ARC-Challenge (ARC) (Clark et al., 2018). We retrieve the top 5 most relevant documents from the corpus as candidate documents. Extract Match (EM) is used as the evaluation metric, and results are reported on the test set for both datasets.

Multi-Hop Tasks include three multi-hop question-answering datasets: 2WikiMultiHopQA (2WIKI), HotpotQA (HotQ), and Musique (MSQ). Following (Chan et al., 2024), we adopt a reading comprehension setup for these datasets, using candidate documents from their original sources. Each question is linked to 10 passages, with only a few (2 for HotQ and 2 or 4 for 2WIKI) being relevant. MSQ is more challenging, requiring 2, 3, or 4 reasoning hops to answer. We use accuracy as the evaluation metric.

Domain-Specific Tasks include two datasets: CFQA (Chen et al., 2022) in the financial domain and PubMedQA (Jin et al., 2019) in the medical domain. For both, we adopt a reading comprehension setup, utilizing the provided context as candidate documents. Exact Match (EM) is used as the evaluation metric.

B Additionally Experiments

B.1 More Evaluation Datasets

Long-form QA Evaluation To explore the performance of RAG-Instruct in more general scenarios, we conducted evaluations on the ASQA dataset (Stelmakh et al., 2022). The results are shown in Table 10. The metrics used for ASQA are

Statistic	Q Num	Avg. Q Len.	Avg. A Len.	Avg. D_s Num.	Retrieved Docs Num.	RAG Scenarios
RAG-Instruct	40000	22.1 (words)	81.2 (words)	2.65	10	5

Table 8: More detailed statistics about RAG-Instruct dataset.

Statistic	API Cost ($)	A800 GPU Hours (Training)	A800 GPU Hours (Evaluation)
RAG-Instruct Construction	620	-	-
Llama3.1-8B + RAG-Instruct	-	26.4	5.3
Qwen2.5-7B + RAG-Instruct	-	24.7	5.3
Llama3.1-70B + RAG-Instruct	-	288	24.8
Qwen2.5-72B + RAG-Instruct	-	294	25

Table 9: Model and Cost Statistics. We report the API cost in constructing RAG-Instruct, including the GPU hours used for training and evaluation.

correctness (str-em), citation precision (pre), and recall (rec), following the settings of Self-RAG (Asai et al., 2024a). The results demonstrate that our RAG-Instruct exhibits strong generalization and performs well in more general scenarios.

Additionally, Work (Dong et al., 2024) also attempts to align RAG with instruction fine-tuning. Compared to their approach, we argue that our **RAG-Instruct** framework provides stronger advantages in multi-hop RAG scenarios. As their model is not publicly available, we evaluate our method on the FollowRAG benchmark they proposed for comparison. The results are presented in Table 11. *IF* (Instruction Following) measures how well the model adheres to atomic instructions, based on the pass rate across samples. *RAG* evaluates the correctness of the model's outputs compared to gold answers, using GPT-4o for scoring. As shown in the table, our model consistently outperforms theirs, particularly in multi-hop tasks such as HotpotQA.

Method	ASQA (em)	ASQA (pre)	ASQA (rec)
Llama-3-Instruct-8B	43.8	62.9	66.4
Self-RAG (3-8b)	36.9	69.7	69.7
InstructRAG (3-8b)	47.6	65.7	70.5
RAG-Instruct (3-8b)	49.1	70.5	72.8

Table 10: Evaluation results on the ASQA dataset to explore the generalization of RAG-Instruct in broader scenarios. Metrics include correctness (str-em), citation precision (pre), and recall (rec), following the settings of Self-RAG.

Method	TriviaQA			HotpotQA		
	IF	RAG	AVG	IF	RAG	AVG
Llama3-8B-SFT-VIF-RAG	42.7	78.0	60.4	39.6	46.0	42.8
RAG-Instruct (3-8b)	45.3	80.5	62.9	42.4	52.9	47.7

Table 11: Comparison of RAG-Instruct against Llama3-8B-SFT-VIF-RAG on the FollowRAG benchmark. The IF metric measures the pass rate of atomic instruction following, and the RAG metric evaluates output correctness against gold answers using GPT-4o scoring. RAG-Instruct outperforms the baseline, particularly in multi-hop tasks like HotpotQA.

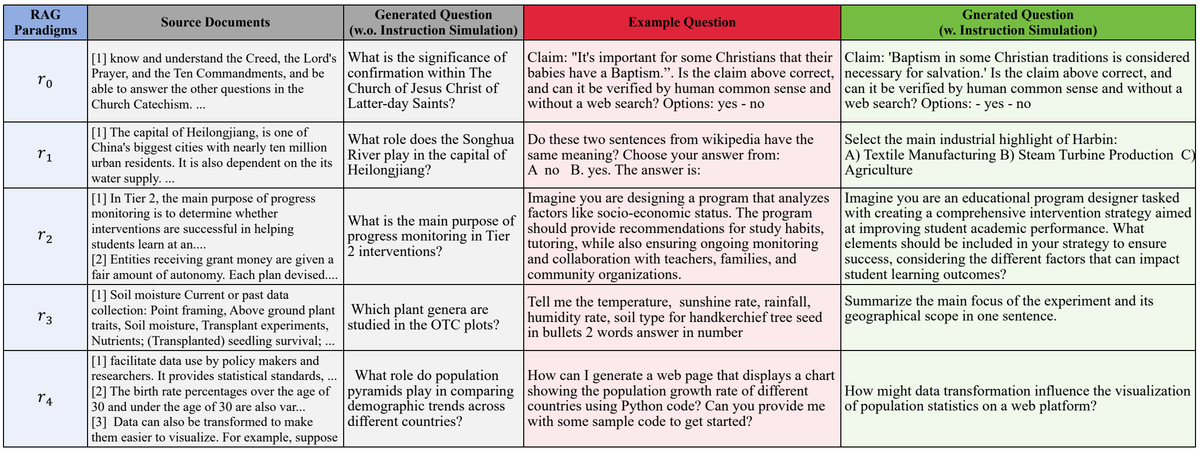
B.2 Complete Ablation Study Results.

As shown in Table 12, removing the *Instruction Simulation* process results in performance declines across all tasks. The drop is smaller for open-ended tasks (TQA) but significantly larger for closed-set (ARC), multi-hop (HotP) tasks. We observe that without *Instruction Simulation*, GPT-4o tends to generate overly simple and uniform questions, resembling open-ended ones, leading to minimal impact on closed-set evaluation. However, the diverse formats of closed-set, multi-hop, and domain-specific tasks, such as multiple-choice and multi-hop reasoning, pose challenges that the model struggles to handle. This highlights the critical role of *Instruction Simulation* in enabling the model to adapt to a wide variety of tasks.

Furthermore, we provide specific cases in Appendix B.5, demonstrating that *Instruction Simulation* generates questions that closely resemble exemplar questions, significantly enhancing diversity compared to those produced without it. Given the high quality and diversity of the synthesized dataset, *Instruction Simulation* ensures both attributes effectively.

B.3 Experiments on Different Retrieval Source

To further explore the generalization of our method, we investigate the impact of using different retrieval sources. Specifically, we further evaluate our method on four single-hop QA tasks, including ARC, PQA, TQA, and OBQA, utilizing DuckDuckGo, Wikipedia, and Bing Search as retrieval



RAG Paradigms	Source Documents	Generated Question (w.o. Instruction Simulation)	Example Question	Gnerated Question (w. Instruction Simulation)
r_0	[1] know and understand the Creed, the Lord's Prayer, and the Ten Commandments, and be able to answer the other questions in the Church Catechism. ...	What is the significance of confirmation within The Church of Jesus Christ of Latter-day Saints?	Claim: "It's important for some Christians that their babies have a Baptism.”. Is the claim above correct, and can it be verified by human common sense and without a web search? Options: yes - no	Claim: 'Baptism in some Christian traditions is considered necessary for salvation.' Is the claim above correct, and can it be verified by human common sense and without a web search? Options: - yes - no
r_1	[1] The capital of Heilongjiang, is one of China's biggest cities with nearly ten million urban residents. It is also dependent on the its water supply. ...	What role does the Songhua River play in the capital of Heilongjiang?	Do these two sentences from wikipedia have the same meaning? Choose your answer from: A no B. yes. The answer is:	Select the main industrial highlight of Harbin: A) Textile Manufacturing B) Steam Turbine Production C) Agriculture
r_2	[1] In Tier 2, the main purpose of progress monitoring is to determine whether interventions are successful in helping students learn at an.... [2] Entities receiving grant money are given a fair amount of autonomy. Each plan devised....	What is the main purpose of progress monitoring in Tier 2 interventions?	Imagine you are designing a program that analyzes factors like socio-economic status. The program should provide recommendations for study habits, tutoring, while also ensuring ongoing monitoring and collaboration with teachers, families, and community organizations.	Imagine you are an educational program designer tasked with creating a comprehensive intervention strategy aimed at improving student academic performance. What elements should be included in your strategy to ensure success, considering the different factors that can impact student learning outcomes?
r_3	[1] Soil moisture Current or past data collection: Point framing, Above ground plant traits, Soil moisture, Transplant experiments, Nutrients; (Transplanted) seedling survival; ...	Which plant genera are studied in the OTC plots?	Tell me the temperature, sunshine rate, rainfall, humidity rate, soil type for handkerchief tree seed in bullets 2 words answer in number	Summarize the main focus of the experiment and its geographical scope in one sentence.
r_4	[1] facilitate data use by policy makers and researchers. It provides statistical standards, ... [2] The birth rate percentages over the age of 30 and under the age of 30 are also var... [3] Data can also be transformed to make them easier to visualize. For example, suppose	What role do population pyramids play in comparing demographic trends across different countries?	How can I generate a web page that displays a chart showing the population growth rate of different countries using Python code? Can you provide me with some sample code to get started?	How might data transformation influence the visualization of population statistics on a web platform?

Figure 4: Some cases of RAG-Instruct for each RAG scenario. We compare the generated questions with and without using Instruction Simulation.

	Open-ended			Closed-set			Multi-hop			Domain-specific		
	WQA (acc)	PQA (acc)	TQA (acc)	OBQA (EM)	Pub (EM)	ARC (EM)	2WIKI (acc)	HotP (acc)	MSQ (acc)	CFQA (EM)	PubMed (EM)	AVG
RAG-Instruct$_{20k}$ (Llama3.1-8B)	64.6	64.8	77.0	80.2	76.0	79.4	73.0	53.1	29.7	55.4	77.2	**66.4**
w.o. Simulation$_{20k}$	63.4	63.1	75.9	74.2	71.4	70.4	62.5	47.7	25.0	47.4	70.4	61.1
Llama3.1-8B-Instruct *w.o. Retrieval*	59.3	28.3	63.1	60.2	62.0	64.1	49.6	33.9	10.6	-	-	**47.9**
RAG-Instruct *w.o. Retrieval*	57.6	28.4	63.2	61.2	60.6	62.8	47.7	33.4	10.1	-	-	47.3

Table 12: Ablation Study on RAG-Instruct. *w.o. Simulation* indicates the removal of the *Instruction Simulation* process, while *w.o. Retrieval* indicates the performance in non-retrieval scenarios.

Method	ARC	PQA	OBQA	WQA	AVG.(↑)	VAR.(↓)
Self-RAG (Llama2-7B)						
+ DuckDuckGo	72.1	56.7	76.4	48.1	62.9	1.9
+ WIKI	73.1	55.8	78.0	49.0		
+ BingSearch	68.6	53.2	76.8	46.4		
RQ-RAG (Llama2-7B)						
+ DuckDuckGo	69.0	58.3	79.8	52.4	65.2	1.6
+ WIKI	68.3	57.1	80.6	56.5		
+ BingSearch	68.9	55.6	78.8	57.4		
RAG-Instruct (Llama2-7B)						
+ DuckDuckGo	75.1	63.0	74.4	68.1	**69.7**	**0.7**
+ WIKI	74.8	62.4	71.4	67.2		
+ BingSearch	75.5	63.8	72.0	69.0		

Table 13: Performance comparison of different retrieval sources. AVG. represents the mean, and VAR. represents the variance.

sources during inference. As shown in Table 13, our RAG-Instruct method demonstrates strong resilience to changes in retrieval sources compared to Self-RAG and RQ-RAG. We use the official API to obtain retrieval results.

While Self-RAG, primarily curated using Wikipedia, shows notable performance drops (3-5%) when switching to Bing Search (with a variance of 1.9), and RQ-RAG similarly experiences performance inconsistencies (variance of 1.6), our RAG-Instruct method exhibits minimal performance fluctuations across different data sources. Specifically, the average performance of RAG-Instruct remains consistently high (69.7) with a variance of only 0.7, even when employing DuckDuckGo, Wikipedia, or Bing Search for retrieval.

This demonstrates that RAG-Instruct not only achieves higher overall performance but also maintains exceptional robustness and stability across diverse retrieval sources, highlighting its superior generalization capabilities compared to existing methods.

B.4 Experiments on Different Retrievers

To further explore the generalization of RAG-Instruct across different retrievers, we also conduct experiments with the BM25 retriever (Robertson et al., 2009), and the results are shown in Table 14. The results indicate that our RAG-Instruct demonstrates excellent generalization across various retrievers.

B.5 Synthetic Data Cases.

We provide specific synthetic data cases, as shown in Figure 4. For each RAG scenario, our synthetic data closely aligns with the particular requirements of that scenario. Additionally, we demonstrate that *Instruction Simulation* generates questions that closely resemble exemplar questions, significantly

	Open-ended			Closed-set			Multi-hop			Domain-specific	
	WQA (acc)	PQA (acc)	TQA (acc)	OBQA (EM)	Pub (EM)	ARC (EM)	2WIKI (acc)	HotP (acc)	MSQ (acc)	CFQA (EM)	PubMed (EM)
Llama-3.1-8B	53.7	52.4	58.8	64.1	56.2	61.6	55.0	45.1	28.3	55.3	68.0
Llama-3.1-8B +RAG-Instruct	**62.7**	**58.4**	**65.2**	**70.2**	**71.2**	**79.6**	**60.3**	**52.4**	**30.7**	**56.5**	**72.0**

Table 14: Performance on BM25 retriever. **Bold** indicates the best experimental results. The datasets were fine-tuned using identical hyperparameters.

enhancing diversity compared to those produced without it. Given the high quality and diversity of the synthesized dataset, *Instruction Simulation* effectively ensures both attributes.

B.6 The effect of RAG-Instruct on Model's General Capabilities.

To evaluate the impact of fine-tuning on the **RAG-Instruct** dataset on the model's general capabilities, we conducted systematic evaluations on two representative and challenging general benchmarks: **MMLU** and **MMLU-Pro**. Specifically, we fine-tuned the *Llama3.1-8B* model, and the detailed experimental results are presented in Table 15. As shown in the table, our RAG-Instruct enhances the capabilities of RAG without compromising the model's general capabilities.

Model	Accuracy	
	MMLU-Pro	MMLU
LLaMA3.1-8B-Instruct	45.7	70.2
Llama-3.1-70B-Instruct	67.6	82.8
Llama3-8B + RAG-Instruct	44.2	72.5
Llama3-70B + RAG-Instruct	65.6	83.4

Table 15: Model Performance for RAG-Instruct trained with *Llama3.1-8B* on MMLU and MMLU-Pro.

B.7 Integration with General Instruction Datasets

As RAG-Instruct serves as an instruction-tuning dataset, its integration with other general instruction-tuning datasets is essential. To validate this, we conducted experiments by mixing RAG-Instruct with general instruction datasets during the training of Llama3.1-8B-base. Specifically, we sampled 5k data points from Evol-Instruct, ShareGPT, SlimOrca, and Alpaca, combining them with RAG-Instruct, resulting in a total of 60k data points for fine-tuning. We then evaluated the model in both RAG and non-RAG scenarios. As shown in Table 16, our results demonstrate that: (1) RAG-Instruct effectively enhances the model's RAG capabilities, even when mixed with other instruction datasets. (2) Mixing RAG-Instruct with general instruction data slightly improves the model's general instruction-following abilities, but it also slightly diminishes its RAG capabilities.

We plan to explore in future work the integration of RAG-Instruct with other types of instruction data, including more detailed investigations into the optimal mixing ratios and other related factors.

C Detailed Prompts in our Experiments

C.1 Prompts for dividing the datasets into five RAG scenarios.

To explore the performance of RAG methods across five different scenarios, we use GPT-4o to categorize questions from two QA datasets: Single-hop QA (TriviaQA) and Multi-hop QA (HotPotQA), into relevant subsets based on the defined RAG scenarios. The prompts used for categorization are shown in Figure 6 (Single-hop QA) and Figure 8 (Multi-hop QA). The final data volume for each subset is shown in Table 17.

C.2 Prompts for synthesizing data for five RAG scenarios.

We construct five RAG paradigms as described in Figure 9-13. To generate data for each RAG paradigm, we simply provide the randomly selected source documents `<Documents>` and the simulated instruction `<Simulated Instruction>`.

	Open-ended			Closed-set			Multi-hop			Domain-specific		
	WQA	PQA	TQA	OBQA	Pub	ARC	2WIKI	HotP	MSQ	CFQA	PubMed	AVG
	(acc)	(acc)	(acc)	(EM)	(EM)	(EM)	(acc)	(acc)	(acc)	(EM)	(EM)	
RAG-Instruct *with Retrieval*	69.7	68.4	80.0	84.4	77.2	79.9	79.3	56.4	33.7	57.8	77.0	**69.5**
Mixed-data *with Retrieval*	68.8	68.3	79.1	84.7	77.5	79.1	76.8	57.4	33.8	56.8	76.2	68.9
Llama3.1-8B-Instruct *w.o. Retrieval*	59.3	28.3	63.1	60.2	62.0	64.1	49.6	33.9	10.6	-	-	**47.9**
Mixed-data *w.o. Retrieval*	58.9	29.9	64.1	60.2	61.2	63.0	48.5	33.2	10.5	-	-	47.7
RAG-Instruct *w.o. Retrieval*	57.6	28.4	63.2	61.2	60.6	62.8	47.7	33.4	10.1	-	-	47.3

Table 16: The effect of mixing RAG-Instruct with general instruction data. "Mix-data" refers to the combination of 20K general instruction data with RAG-Instruct. All experiments are based on training the Llama3.1-8B model.

	TriviaQA(Single-hop QA)			**HotpotQA (Multi-hop QA)**	
	Helpful	Midhelpful	Helpless	Helpful	Midhelpful
Mumber of Data	5628	894	791	4015	3390

Table 17: Detailed information on dataset subsets categorized into five RAG scenarios.

Knowledge-Intensive Data Selection Prompt

`{Question}`
Please determine if retrieving external information would help answer the above question. If it helps, answer "True", otherwise answer "False".

Figure 5: The prompt of filtering knowledge-intensive instructions from synthetic datasets

Dividing Prompt for Single-hop Question.

Documents:
`{Doucments}`

Question:
`{Question}`

Answer:
`{Answer}`

Based on the question and its answer, along with the provided documents, carefully review the documents to assess their overall usefulness in answering the question. Avoid evaluating each document individually; instead, consider the documents as a whole. Choose the most accurate option based on how much the documents contribute to the answer: 1. Very helpful: The answer is directly provided in the documents. 2. Partially helpful: The documents offer supporting information or clues but do not provide an explicit answer. 3. Not helpful: The documents do not contribute to answering the question. Please directly respond with only the chosen option (1, 2, or 3).

Figure 6: The prompt for dividing the single-hop question answering datasets into five RAG scenarios.

Task	**Template**

Open-ended	### Instruction:
	Reference Document:
	`{RETRIEVED DOCUMENTS}`
	Please refer to the documents above and answer the following question:
	`{QUESTION}`
	### Response:

Domain-specific	

OBQA & ARC	### Instruction:
	Reference Document:
	`{RETRIEVED DOCUMENTS}`
	Given four answer candidates, A, B, C and D, choose the best answer choice for the question.
	Please refer to the documents above and answer the following question:
	`{QUESTION (Including Options) }`
	### Response:

Pub (FEVER)	### Instruction:
	Reference Document:
	`{RETRIEVED DOCUMENTS}`
	Is the following statement correct or not? Say true if it's correct; otherwise, say false.
	Please refer to the documents above and answer the following question:
	`{QUESTION}`
	### Response:

Multi-hop	### Instruction:
	Reference Document:
	`{RETRIEVED DOCUMENTS}`
	Please refer to the documents above and answer the following question:
	`{QUESTION}`
	### Response:

Domain-specific	

CFQA	### Instruction:
	Reference Document:
	`{RETRIEVED DOCUMENTS}`
	Please refer to the documents above and answer the following question:
	`{PREVIOUS QUESTIONS ANSWERS}`
	`{QUESTION}`
	### Response:

PubMed	### Instruction:
	Reference Document:
	`{RETRIEVED DOCUMENTS}`
	Please refer to the documents above and answer the following question:
	Answer the question with "yes" or "no" or "maybe".
	`{QUESTION}`
	### Response:

Table 18: **Prompt templates** in our Evaluation. For *Open-ended* and *Close-set datasets*, `RETRIEVED DOCUMENTS` are sourced from the retrieval corpus (e.g., Wikipedia). For *Multi-hop* and *Domain-specific* datasets, `RETRIEVED DOCUMENTS` come from the context provided in datasets.

Prompt for QA Pair Quality Check

Document:
`{Document}`

Question:
`{Question}`

Answer:
`{Answer}`

RAG Scenario:
`{RAG Scenario}`

Simulated Task Question:
`{Simulated Task Question}`

You are an expert AI assistant tasked with evaluating the quality of the above question and answer, given the retrieved document, the specified RAG scenario, and the simulated task question. Please examine whether the question answer pair exhibits any of the following issues:
Issue 1: The question is vague, incomplete, or logically incoherent.
Issue 2: The answer does not respond to the question.
Issue 3: Check strictly whether the question and answer align with the given RAG scenario. If they do not, please identify the inconsistency.
Issue 4: The question significantly deviates from the expected format or purpose of the provided simulated task.
Issue 5: The question or answer contains content that is ethically inappropriate, harmful, or poses safety risks.

If the question and answer pair has none of the five issues above, return `true`; otherwise, return `false`. Please format your output as follows:

```
 {
"is_passed": true/false,
"explanation": "Brief explanation if any issues are present; otherwise, leave empty or use ’None’."
}
```

Figure 7: The prompt used to identify five types of quality issues in QA pairs for RAG-Instruct.

Dividing Prompt for Multi-hop Question.

Documents:
`{Doucments}`

Question:
`{Question}`

Answer:
`{Answer}`

Based on the question and answer provided, carefully review the given documents and assess their overall usefulness in addressing the question. Avoid evaluating each document individually; instead, consider the documents as a whole. Choose the most accurate option based on how much the documents contribute to the answer: 1. Very helpful: The answer can be directly derived from multiple documents. 2. Partially helpful: The documents offer supporting information or clues but do not provide an explicit answer. It needs further reasoning or more knowledge. Please directly respond with only the chosen option (1, or 2).

Figure 8: The prompt for dividing the multi-hop question answering datasets into five RAG scenarios.

Useless Doc (r_0)

<Documents>
[1] {<Document 1>}
</Documents>

Your task is to generate an English question q* and a corresponding response a* based on the provided <Documents>. Please note that the question q* can take various forms, not limited to questions with a question mark, but also including statements, instructions, and other formats. You need to follow the requirements below to generate the q* and a* (RAG Paradigms):
1. q* should be related to the <Documents>, but the <Documents> can not provide any useful information for answering q*.
2. a* should be able to answer q*, ensuring that the response a* is accurate, detailed, and comprehensive.

Additionally, to ensure diversity, richness, and high quality in the question q* you generate, we will randomly provide a question for you to emulate. In other words, while satisfying the requirements above, make q* similar in task requirement and expression to the <Simulated Instruction> below:
<Simulated Instruction>
{<Simulated Instruction>}

</Simulated Instruction>

Please directly generate the question-answer pair (q*, a*) following all the rules above in the format of {"q*": ..., "a*": ...}. Ensure the quality of the generated (q*, a*).

Figure 9: The prompt for synthesizing Useless Doc (r_0) data.

Single-Doc Support (r_1)

<Documents>
[1] {<Document 1>}
</Documents>

Your task is to generate an English question q* and a corresponding response a* based on the provided <Documents>. Please note that the question q* can take various forms, not limited to questions with a question mark, but also including statements, instructions, and other formats. You need to follow the requirements below to generate the q* and a* (RAG Paradigms):
1. <Documents> can support q* by providing useful information or hints, but they do not contain explicit answers.
2. a* should use useful information from <Documents> to aid in answering q*, ensuring that the response is accurate, detailed, and comprehensive.

Additionally, to ensure diversity, richness, and high quality in the question q* you generate, we will randomly provide a question for you to emulate. In other words, while satisfying the requirements above, make q* similar in task requirement and expression to the <Simulated Instruction> below:

<Simulated Instruction>
{<Simulated Instruction>}
</Simulated Instruction>

Please directly generate the question-answer pair (q*, a*) following all the rules above in the format of {"q*": ..., "a*": ...}. Ensure the quality of the generated (q*, a*).

Figure 10: The prompt for synthesizing Single-Doc Support (r_1) data.

Multi-Doc Support (r_2)

<Documents>
[1] {<Document 1>}
[2] {<Document 2>}
[3] ...
</Documents>

Your task is to generate an English question q* and a corresponding response a* based on the provided <Documents>. Please note that the question q* can take various forms, not limited to questions with a question mark, but also including statements, instructions, and other formats. You need to follow the requirements below to generate the q* and a* (RAG Paradigms):
1. Multiple documents within <Documents> can support q* by providing useful information or hints, but they do not contain explicit answers.
2. a* should use useful information from <Documents> to aid in answering q*, ensuring that the response is accurate, detailed, and comprehensive.

Additionally, to ensure diversity, richness, and high quality in the question q* you generate, we will randomly provide a question for you to emulate. In other words, while satisfying the requirements above, make q* similar in task requirement and expression to the <Simulated Instruction> below:
<Simulated Instruction>
{<Simulated Instruction>}
</Simulated Instruction>

Please directly generate the question-answer pair (q*, a*) following all the rules above in the format of {"q*": ..., "a*": ...}. Ensure the quality of the generated (q*, a*).

Figure 11: The prompt for synthesizing Multi-Doc Support (r_2) data.

Single-Doc Answer (r_3)

<Documents>
[1] {<Document 1>}
</Documents>

Your task is to generate an English question q* and a corresponding response a* based on the provided <Documents>. Please note that the question q* can take various forms, not limited to questions with a question mark, but also including statements, instructions, and other formats. You need to follow the requirements below to generate the q* and a* (RAG Paradigms):
1. Ensure that q* can be answered directly using the content of <Documents>, meaning its answer can be fully derived from <Documents>.
2. a* should use the information from <Documents> to answer q* accurately, ensuring that the response is accurate, detailed, and comprehensive.

Additionally, to ensure diversity, richness, and high quality in the question q* you generate, we will randomly provide a question for you to emulate. In other words, while satisfying the requirements above, make q* similar in task requirement and expression to the <Simulated Instruction> below:

<Simulated Instruction>
{<Simulated Instruction>}
</Simulated Instruction>

Please directly generate the question-answer pair (q*, a*) following all the rules above in the format of {"q*": ..., "a*": ...}. Ensure the quality of the generated (q*, a*).

Figure 12: The prompt for synthesizing Single-Doc Answer (r_3) data.

Multi-Doc Answer (r_4)

<Documents>
[1] {<Document 1>}
[2] {<Document 2>}
[3] ...
</Documents>

Your task is to generate an English question q* and a corresponding response a* based on the provided <Documents>. Please note that the question q* can take various forms, not limited to questions with a question mark, but also including statements, instructions, and other formats. You need to follow the requirements below to generate the q* and a* (RAG Paradigms):
1. The answer to q* can be derived from multiple documents within <Documents>, involving multi-hop reasoning or the integration of information from several documents.
2. a* should leverage the information in <Documents> to provide an accurate answer to q*, ensuring that the response is accurate, detailed, and comprehensive.

Additionally, to ensure diversity, richness, and high quality in the question q* you generate, we will randomly provide a question for you to emulate. In other words, while satisfying the requirements above, make q* similar in task requirement and expression to the <Simulated Instruction> below:
<Simulated Instruction>
{<Simulated Instruction>}
</Simulated Instruction>

Please directly generate the question-answer pair (q*, a*) following all the rules above in the format of {"q*": ..., "a*": ...}. Ensure the quality of the generated (q*, a*).

Figure 13: The prompt for synthesizing Multi-Doc Answer (r_4) data.