INSTRUCTRAG: INSTRUCTING RETRIEVAL AUGMENTED GENERATION VIA SELF-SYNTHESIZED RATIONALES

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) has shown promising potential to enhance the accuracy and factuality of language models (LMs). However, imperfect retrievers or noisy corpora can introduce misleading or even erroneous information to the retrieved contents, posing a significant challenge to the generation quality. Existing RAG methods typically address this challenge by directly predicting final answers despite potentially noisy inputs, resulting in an *implicit* denoising process that is difficult to interpret and verify. On the other hand, the acquisition of explicit denoising supervision is often costly, involving significant human efforts. In this work, we propose INSTRUCTRAG, where LMs *explicitly* learn the denoising process through self-synthesized rationales — First, we instruct the LM to explain how the ground-truth answer is derived from retrieved documents. Then, these rationales can be used either as demonstrations for in-context learning of explicit denoising or as supervised fine-tuning data to train the model. Compared to standard RAG approaches, INSTRUCTRAG requires no additional supervision, allows for easier verification of the predicted answers, and effectively improves generation accuracy. Experiments show INSTRUCTRAG consistently outperforms existing RAG methods in both training-free and trainable scenarios, achieving a relative improvement of 8.3% over the best baseline method on average across five knowledge-intensive benchmarks. Extensive analysis indicates that INSTRUCTRAG scales well with increased numbers of retrieved documents and consistently exhibits robust denoising ability even in out-of-domain datasets, demonstrating strong generalizability.[1]

1 INTRODUCTION

While large language models (LMs) have demonstrated remarkable text generation abilities (Brown et al., 2020; Team et al., 2023; Touvron et al., 2023), they may occasionally produce factually incorrect contents (Dhuliawala et al., 2023; Huang et al., 2023a; Ji et al., 2023; Sun et al., 2023; Xu et al., 2024d; Zhang et al., 2023), particularly when the task at hand requires the most current information or out-of-domain knowledge not adequately represented in the pre-training corpus (Jiang et al., 2023b; Shuster et al., 2021; Yu et al., 2023; Zhao et al., 2023). This limitation significantly hinders the reliable deployment of LMs in high-stakes domains where factuality is crucial (Magesh et al., 2024; Singhal et al., 2023; Xiao et al., 2021; Xiong et al., 2024).

In light of this, retrieval-augmented generation (RAG) (Asai et al., 2023b; Guu et al., 2020; Izacard et al., 2023; Khandelwal et al., 2019; Lewis et al., 2020) has been introduced to enhance the generation accuracy of LMs in knowledge-intensive tasks by leveraging the most up-to-date information and specialized knowledge from external sources (Kasai et al., 2024; Vu et al., 2023; Yang et al., 2024; Zhou et al., 2022). However, the retrieved contents are typically mixed with irrelevant or even erroneous information due to the absence of perfect retrieval solutions (Izacard et al., 2021; Karpukhin et al., 2020; Khattab et al., 2022; 2023; Shi et al., 2023; Su et al., 2024) and the presence of noisy data in the retrieval corpus (Izacard & Grave, 2021; Li et al., 2023; Yoran et al., 2024), posing a long-standing challenge to almost all RAG systems. Typically, vanilla RAG approaches

[1]Our code is included in the submitted supplementary material and available at https://anonymous.4open.science/r/InstructRAG-ICLR2025-Submission.

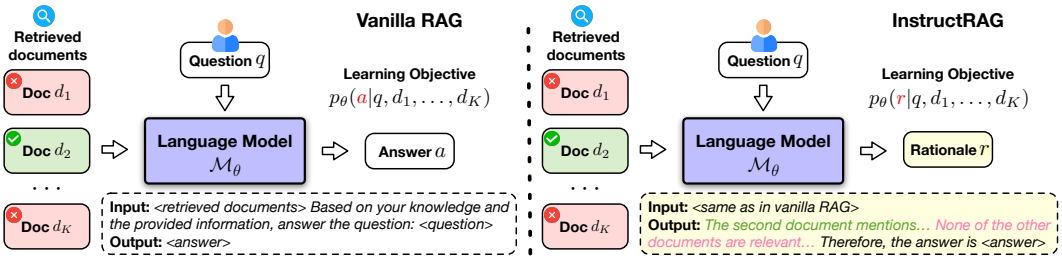


Figure 1: Comparison between vanilla RAG and our INSTRUCTRAG. In vanilla RAG, the model is tasked to directly predict answers given user queries and potentially noisy retrieved documents, without explicit denoising processes or explanations for how the answer is derived. In contrast, our proposed INSTRUCTRAG generates rationales that explicitly denoise the retrieved documents and justify the predicted answers, enhancing both the generation accuracy and trustworthiness.

address this issue *implicitly* by training LMs to directly predict correct answers despite potentially noisy inputs. Such latent processes are not only difficult to interpret and verify but also vulnerable to higher noise ratios, especially when the number of retrieved documents is large (Chen et al., 2024; Cuconasu et al., 2024; Liu et al., 2024; Wu et al., 2024). On the other hand, obtaining high-quality explicit denoising supervision often requires substantial human efforts, which is time-consuming and costly.

In this work, we introduce a new RAG framework, INSTRUCTRAG, which enables the LM to *explicitly* denoise retrieved information and justify its predicted final answers by generating denoising responses (*i.e.*, *rationales*), as illustrated in Figure 1. Compared to vanilla RAG approaches, INSTRUCTRAG does not require any additional supervision, while enjoying improved generation accuracy and trustworthiness. Specifically, our method consists of two steps. First, given a set of question-answer pairs and potentially noisy retrieved documents, we prompt an instruction-tuned LM to synthesize denoising rationales that analyze the documents and articulate how they lead to the ground-truth answers (§ 2.2). Then, these synthetic rationales can be utilized as in-context learning examples or as supervised fine-tuning data, allowing the LM to explicitly learn to denoise retrieved contents (§ 2.3). The effectiveness of INSTRUCTRAG can be attributed to the strong instruction-following ability of LMs (Jiang et al., 2024b; Ouyang et al., 2022; Wei et al., 2021), a significant feature that still remains underexplored in the context of RAG. We show that such self-synthesized rationales not only provide high-quality explicit denoising supervision for in-domain RAG tasks, but also facilitate superior out-of-domain generalization. This finding underscores how instruction-tuned LMs can synthesize generalizable supervision to overcome the inevitable noise in RAG.

The main contributions of this work are as follows: (1) We propose INSTRUCTRAG, a simple yet effective RAG framework that allows LMs to explicitly denoise retrieved contents by generating rationales for better verifiability and trustworthiness. (2) INSTRUCTRAG is a self-synthesis method that does not require additional supervision compared to standard RAG methods, and can be seamlessly applied to both in-context learning and supervised fine-tuning settings. (3) INSTRUCTRAG consistently outperforms state-of-the-art RAG approaches, yielding a relative improvement of 8.3% on average compared to the best baseline method across five knowledge-intensive benchmarks. Extensive analysis and ablation studies further confirm the superiority of self-synthesized denoising rationales, and demonstrate INSTRUCTRAG's robust denoising ability against increased noise ratios and strong task transferability in various training-free and trainable scenarios.

2 OUR METHOD: INSTRUCTRAG

In this section, we first introduce our problem setting (§ 2.1) and then present the proposed framework INSTRUCTRAG that enables LMs to explicitly denoise retrieved contents. As shown in Figure 2, our method consists of two steps. First, we prompt an instruction-tuned LM (*i.e.*, rationale generator $\mathcal{M}_\phi$) to synthesize rationales that provide denoising supervisions (§ 2.2). These rationales aim to explain how to derive the correct answer from potentially noisy retrieved documents for each training sample. Then, we guide the LM (*i.e.*, rationale learner $\mathcal{M}_\theta$) to learn explicit denoising by leveraging these rationales as either in-context learning demonstrations or as supervised fine-tuning data (§ 2.3). As detailed in Algorithm 1, during the entire process, INSTRUCTRAG does not require

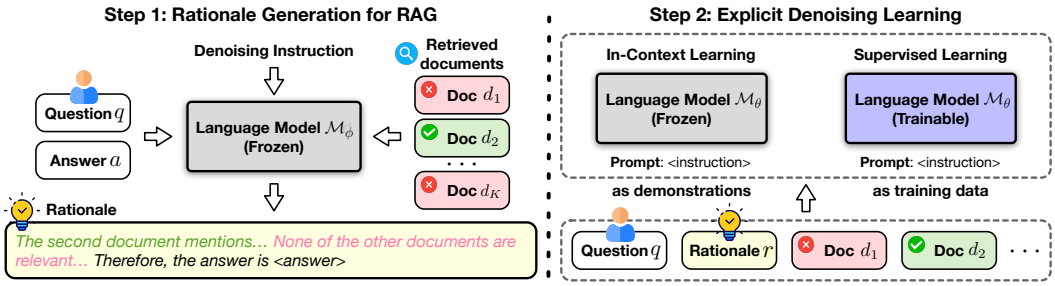


Figure 2: An overview of INSTRUCTRAG. In step one, given the question q, retrieved documents $\{d_1, \cdots, d_K\}$ and ground-truth answer a from the training set, we prompt an instruction-tuned LM (*i.e.*, rationale generator $\mathcal{M}_\phi$) to generate rationale r that explains how the answer can be derived from the potentially noisy input. In step two, we utilize the synthesized rationales from the first step to guide the LM (*i.e.*, rationale learner $\mathcal{M}_\theta$) to explicitly learn denoising of the retrieved documents, either through in-context learning or supervised learning. By default, we use the same model for both $\mathcal{M}_\phi$ and $\mathcal{M}_\theta$, but they can be instantiated with different models as well (see ablation study § 3.3).

Table 1: Rationale generation prompt for the i-th training sample.

Rationale Generation
Input: Read the following documents relevant to the given question: $\{q_i\}$ Document [1] (Title: $\cdots$): {contents of d_i^1} $\cdots$ Please identify documents that are useful to answer the given question: $\{q_i\}$, and explain how the contents lead to the answer: $\{a_i\}$ {task-specific instruction} **Output:** {rationale r_i}

any additional supervisions beyond standard RAG methods. By default, we instantiate both $\mathcal{M}_\phi$ and $\mathcal{M}_\theta$ with the same off-the-shelf instruction-tuned model (*i.e.*, meta-llama/Meta-Llama-3-8B-Instruct), making INSTRUCTRAG a fully *self-synthesis* method. We also experiment with different instantiations of $\mathcal{M}_\phi$ and $\mathcal{M}_\theta$ and conduct ablation study in both training-free and trainable settings (§ 3.3). For simplicity, we use placeholders to represent omitted instructions in the prompts presented in this section, while the full list of complete prompt templates is provided in Appendix D.

2.1 PROBLEM SETTING

We adopt the standard RAG setting where the LM $\mathcal{M}_\theta$ has access to annotated datasets of downstream tasks (*e.g.*, question-answering task $\mathcal{T} = \{\langle q, a \rangle\}$), and an external knowledge base with the off-the-shelf retriever $\mathcal{R}$ for retrieval. Different from previous works (Asai et al., 2023b; Yoran et al., 2024) which leverage additional supervisions from GPT-3 (Brown et al., 2020) or GPT-4 (Achiam et al., 2023), we assume the model has strictly limited access to the above two information sources. Given a question q, the retriever $\mathcal{R}$ returns a set of potentially noisy documents $D = \{d_1, \cdots, d_K\}$ from the external knowledge base. The model is then tasked to predict the correct answer a to the given question q based on D and its own parametric knowledge, denoted as $p_\theta(a|q, D)$.

Our work focuses on investigating the noise robustness of LMs and developing efficient denoising techniques for RAG. Hence, we directly employ off-the-shelf retrievers instead of training our own, and prepend all retrieved documents to the question as input to the model, without any filtering or re-ranking. This setting is orthogonal to existing research efforts centered on optimizing the retriever or performing adaptive retrieval (Asai et al., 2023b; Wang et al., 2024a; Yang et al., 2024).

2.2 RATIONALE GENERATION VIA INSTRUCTION-FOLLOWING

Recent studies (Leike et al., 2018; Meng et al., 2024; Ouyang et al., 2022) have made encouraging progress in aligning LMs with human preferences and intentions, enabling the synthesis of high-

Algorithm 1 INSTRUCTRAG

Require: Retriever $\mathcal{R}$, Rationale generator $\mathcal{M}_\phi$, Rationale learner $\mathcal{M}_\theta$, Training data $\mathcal{T} = \{\langle q, a\rangle\}$

```
/* Training data generation */
```

for each $\langle q, a\rangle \in \mathcal{T}$ **do**
  Retrieve $D = \{d_1, \cdots, d_K\} \leftarrow \mathcal{R}(q)$
  Synthesize denoising rationale $r \leftarrow \mathcal{M}_\phi(q, a, D)$ ▷ Rationale Generation (§ 2.2)
Augment training data $\mathcal{T} \rightarrow \mathcal{T}^+ = \{\langle q, r\rangle\}$

```
/* Two learning modes */
```

if `LearningMode` == `In-Context Learning` **then** ▷ INSTRUCTRAG-ICL
  Sample ICL examples $\mathcal{E} = \{\langle q, r\rangle\} \subseteq \mathcal{T}^+$
  $r \leftarrow \mathcal{M}_\theta(r|q, \mathcal{R}(q), \mathcal{E})$ given inference query q ▷ Detailed in Table 10
else if `LearningMode` == `Fine-Tuning` **then** ▷ INSTRUCTRAG-FT
  Fine-tune $\mathcal{M}_\theta$ on $\mathcal{T}^+$ with retrieved documents $\{\langle q, r, D\rangle\}$
  $r \leftarrow \mathcal{M}_\theta(r|q, \mathcal{R}(q))$ given inference query q ▷ Detailed in Table 11
return r

quality data that closely follows user instructions (Xu et al., 2024c). Inspired by these advances, we propose to leverage the LM's strong instruction-following ability to generate explicit denoising responses (*i.e.*, *rationales*) for RAG. As shown in Table 1, given a QA pair $\langle q_i, a_i\rangle \in \mathcal{T}$ and a set of retrieved documents $\{d_i^1, \cdots, d_i^K\}$, we prompt an off-the-shelf LM $\mathcal{M}_\phi$ (as the rationale generator) with denoising instructions to produce the corresponding rationale r_i that distinguishes useful documents from noisy ones and explains how the contexts lead to the ground-truth answer a_i. To ensure the synthetic rationales are aligned with the ground-truth answers, we use a simple substring match to assess their consistency. The consistency ratio on training samples with at least one relevant document containing the ground-truth answer is 98% on average across five benchmarks, supporting the reliability of synthetic rationales as a sanity check. This allows us to effectively augment the standard dataset $\mathcal{T} = \{\langle q, a\rangle\} \rightarrow \mathcal{T}^+ = \{\langle q, r\rangle\}$ with self-synthesized denoising rationales solely by instructing the LM, without any additional supervision.

We also validate the necessity of using an LM-based generator (*i.e.*, $\mathcal{M}_\phi$) to create the rationales instead of employing simple heuristics — without the generator, rationales can be created in a template-based manner (Table 6), by roughly identifying relevant retrieved documents through simple substring-matching with the ground-truth answer. However, as demonstrated in our ablation study, this approach suffers from semantically inaccurate matching of relevant documents, leading to significant performance degradation. Another advantage of the LM-based generator is that it can produce high-quality rationales even *without referring to the ground-truth answer*, which only results in a minor performance drop. More detailed analyses on rationale generation design can be found in our ablation study (§ 3.3).

2.3 LEARNING DENOISING RATIONALES IN RAG

With the rationale-augmented dataset $\mathcal{T}^+$, it becomes possible to develop a rationale learner $\mathcal{M}_\theta$ that directly learns explicit denoising for RAG tasks with efficient learning strategies. Next, we introduce two simple yet effective learning methods in the *training-free* and *trainable* RAG settings, namely, INSTRUCTRAG-ICL and INSTRUCTRAG-FT.

INSTRUCTRAG-ICL is a training-free instantiation of INSTRUCTRAG where the model learns denoising rationales via in-context learning (ICL). As shown in Table 10, given a test question q and a set of retrieved documents $D = \{d_1, \cdots, d_K\}$, we first randomly sample N demonstrations $\langle q_i, r_i\rangle \in \mathcal{T}^+$ from the rationale-augmented training dataset, and then prompt the model to follow the exemplars and generate rationale r. To save memory and enhance inference efficiency, we only show exemplary questions and their corresponding rationales in such ICL demonstrations.

INSTRUCTRAG-FT is a trainable instantiation of INSTRUCTRAG that learns denoising rationales via supervised fine-tuning (FT) with standard language modeling objective. As defined in Eq. 1, it maximizes the likelihood of rationale r conditioned on question q and retrieved documents D.

$$\max_\theta \mathbb{E}_{(q,r)\sim\mathcal{T}^+} \log p_\theta(r|q, D). \tag{1}$$

where θ represents the model parameters. Both the training and inference of INSTRUCTRAG-FT share the same data format. As depicted in Table 11, it takes as input the retrieved documents followed by the question, and outputs the denoising rationale r.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETTING

Table 2: Dataset statistics and retrieval setting.

Dataset	Train	Test	Retriever	Top-K	Recall@K
PopQA	12,868	1,399	Contriever	5	68.7
TriviaQA	78,785	11,313	Contriever	5	73.5
Natural Questions	79,168	3,610	DPR	5	68.8
ASQA	4,353	948	GTR	5	82.2
2WikiMultiHopQA	167,454	12,576	BM25	10	40.7

RAG tasks and evaluation metrics. We extensively validate the effectiveness of INSTRUCTRAG on five knowledge-intensive benchmarks, including PopQA (Mallen et al., 2023), TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), ASQA (Stelmakh et al., 2022), and 2WikiMultiHopQA (Ho et al., 2020). We use Wikipedia corpus as the retrieval source, and test our method with both sparse and dense off-the-shelf retrievers, including BM25 (Robertson & Walker, 1994), DPR (Karpukhin et al., 2020), GTR (Ni et al., 2022) and Contriver (Izacard et al., 2021). The retrieval quality is measured by Recall@K, indicating whether the retrieved K documents contain the correct answer. Table 2 shows the detailed dataset statistics. Following standard evaluation settings (Asai et al., 2023b), we adopt the official metric of correctness (*str-em*), citation precision (*pre*) and recall (*rec*) for ASQA (Gao et al., 2023a), and use *accuracy* for the other tasks, which measures whether the ground-truth answers are included in the model generations (Mallen et al., 2023; Schick et al., 2023). Additionally, we also adopt LLM-as-a-judge for further evaluation (§ 3.4), as the above standard metrics are subject to the limitations of pattern-matching, which cannot accurately handle semantic equivalence.

Baselines. We compare our method with a wide range of RAG baselines under both training-free and trainable settings. Given that state-of-the-art LMs have incorporated a large amount of world-knowledge during the pre-training stage, we also report the performance of a non-retrieval baseline (namely, **vanilla zero-shot prompting**) for reference. Specifically, the training-free RAG baselines includes: (1) **in-context retrieval-augmented language modeling (RALM)** (Ram et al., 2023), a prompting method that extends the non-retrieval baseline by presenting the model with retrieved documents; (2) **few-shot demonstration with instruction**, an ICL method using ground-truth question-answer pairs sampled from the training set as demonstration exemplars.

The trainable RAG baselines include: (1) **vanilla supervised fine-tuning (SFT)**, a supervised method with the training objective of maximizing the data likelihood of ground-truth answer given potentially noisy input; (2) **RetRobust** (Yoran et al., 2024), which fine-tunes the RAG model on a mixture of relevant and irrelevant contexts to make it robust to irrelevant contexts; (3) **Self-RAG** (Asai et al., 2023b), a strong trainable baseline, focusing on adaptive retrieval controlled by special reflection tokens. Both RetRobust and Self-RAG were originally built on Llama-2 (Touvron et al., 2023) with additional supervisions. For example, RetRobust augments the training data for multi-hop reasoning tasks (*e.g.*, 2WikiMultiHopQA) by prompting GPT-3 to decompose the original query and generate intermediate subqueries, and Self-RAG requires GPT-4 to generate additional reflective tokens to augment training samples.

For a fair comparison, we re-implement the two methods on Llama-2_{7B} and/or Llama-2_{13B} with augmented training data released by their authors, and report their performance as the higher one between the original scores and our reproduced results. As our method adopts instruction-tuned Llama-3 as the backbone model, we also train RetRobust and Self-RAG with Llama-3-Instruct$_{8B}$ and optimize their performance through extensive hyper-parameters search. More details on implementation, including training, inference, and prompt design are available in Appendix B and Appendix D. We also present some case studies in Appendix C.

Table 3: Overall results of INSTRUCTRAG and baselines on five knowledge-intensive benchmarks in training-free and trainable RAG settings. We re-implement baselines and report their performance as the higher one between the original scores and our reproduced results. * indicates the results copied from Asai et al. (2023b) for reference. "–" indicates the results are not reported in the original paper or not applicable (*e.g.*, some methods cannot produce citations). The best performance is highlighted in ***bold.***

Method	PopQA (acc)	TriviaQA (acc)	NQ (acc)	MultiHopQA (acc)	ASQA (em)	ASQA (pre)	ASQA (rec)
Baselines w/o Retrieval							
Vanilla Zero-shot Prompting							
ChatGPT*	29.3	74.3	–	–	35.3	–	–
Llama-3-Instruct$_{8B}$	22.8	69.4	46.6	45.6	30.6	–	–
Llama-3-Instruct$_{70B}$	28.9	80.6	57.9	57.5	39.1	–	–
RAG w/o Training							
In-Context RALM (Ram et al., 2023)							
ChatGPT*	50.8	65.7	–	–	40.7	65.1	76.6
Llama-3-Instruct$_{8B}$	62.3	71.4	56.8	43.4	40.0	62.1	66.4
Llama-3-Instruct$_{70B}$	63.8	76.3	60.2	51.2	43.1	62.9	67.6
Few-Shot Demo. w/ Instruction							
Llama-3-Instruct$_{8B}$	63.1	74.2	60.1	45.3	42.6	55.0	64.4
Llama-3-Instruct$_{70B}$	63.9	79.1	62.9	53.9	45.4	49.3	57.1
INSTRUCTRAG-ICL							
Llama-3-Instruct$_{8B}$	64.2	76.8	62.1	50.4	44.7	**70.9**	**74.1**
Llama-3-Instruct$_{70B}$	**65.5**	**81.2**	**66.5**	**57.3**	**47.8**	69.1	71.2
RAG w/ Training							
Vanilla Supervised Fine-tuning							
Llama-3-Instruct$_{8B}$	61.0	73.9	56.6	56.1	43.8	–	–
Self-RAG (Asai et al., 2023b)							
Llama-2$_{7B}$	55.8	68.9	42.4	35.9	30.0	66.9	67.8
Llama-2$_{13B}$	56.3	70.4	46.4	36.0	31.4	**70.3**	**71.3**
Llama-3-Instruct$_{8B}$	55.8	71.4	42.8	32.9	36.9	69.7	69.7
RetRobust (Yoran et al., 2024)							
Llama-2$_{13B}$	–	–	39.6	51.5	–	–	–
Llama-3-Instruct$_{8B}$	56.5	71.5	54.2	54.7	40.5	–	–
INSTRUCTRAG-FT							
Llama-3-Instruct$_{8B}$	**66.2**	**78.5**	**65.7**	**57.2**	**47.6**	65.7	70.5

3.2 MAIN RESULT

Table 3 shows the overall experimental results, providing a comprehensive comparison between our INSTRUCTRAG and baseline methods in both training-free and trainable RAG settings.

Baselines without retrieval. As shown in the first block, the basic instruction-tuned models (Llama-3-Instruct$_{8B}$ and Llama-3-Instruct$_{70B}$) already achieve notable performance across all five benchmarks, with the 70B model exhibiting a surprisingly competitive performance of 80.6% on the TriviaQA. This observation suggests that the required knowledge for these tasks mostly falls within the LM's parametric knowledge, probably due to what is known as *data contamination* (*i.e.*, the presence of test data of downstream tasks in the pre-training data of LMs) (Golchin & Surdeanu, 2023; Jacovi et al., 2023; Magar & Schwartz, 2022).

RAG without training. The second block shows the comparison among training-free RAG methods. In-context RALM and few-shot demonstration with instruction methods generally achieve higher performance than the non-retrieval baseline, highlighting the importance of retrieval for knowledge-intensive tasks. Encouragingly, our INSTRUCTRAG-ICL consistently outperforms all training-free baselines across various metrics, confirming the effectiveness of self-synthesized denoising rationales. Moreover, the boost from 8B to 70B model indicates that INSTRUCTRAG-ICL scales effectively with larger backbone models, validating the generalizability of our method.

RAG with training. As present in the bottom block of Table 3, our INSTRUCTRAG-FT not only surpasses all non-retrieval and training-free baselines across all five benchmarks, but also significantly outperforms trainable RAG baselines on almost every metric. The only exception is in the ASQA

Table 4: Ablation study on the impact of ground-truth answer, retrieved documents, and model size on rationale generation, and the use of demonstrations during model inference. The results of our default setting in INSTRUCTRAG are underlined.

Method	Trainable RAG Setting PopQA	Trainable RAG Setting ASQA	Training-free RAG Setting PopQA	Training-free RAG Setting ASQA
Rationale Generation Design				
with both	66.2	47.6	64.2	44.7
w/o ground-truth answer	65.2 (↓ 1.5%)	46.4 (↓ 2.5%)	64.0 (↓ 0.3%)	44.5 (↓ 0.4%)
w/o retrieved documents	64.5 (↓ 2.6%)	45.2 (↓ 5.0%)	64.1 (↓ 0.2%)	44.3 (↓ 0.9%)
Model Size of Rationale Generator				
rationale template (no generator)	59.6 (↓ 10.0%)	46.3 (↓ 2.7%)	60.0 (↓ 6.5%)	41.4 (↓ 7.4%)
Llama-3-Instruct (8B)	66.2	47.6	64.2	44.7
Llama-3-Instruct (70B)	67.0 (↑ 1.2%)	49.1 (↑ 3.2%)	64.8 (↑ 0.9%)	47.9 (↑ 7.1%)
Inference Strategy Comparison				
w/o demonstration	66.2	47.6	63.0 (↓ 1.9%)	43.1 (↓ 3.6%)
w/ demonstration	66.1 (↓ 0.2%)	44.7 (↓ 6.1%)	64.2	44.7

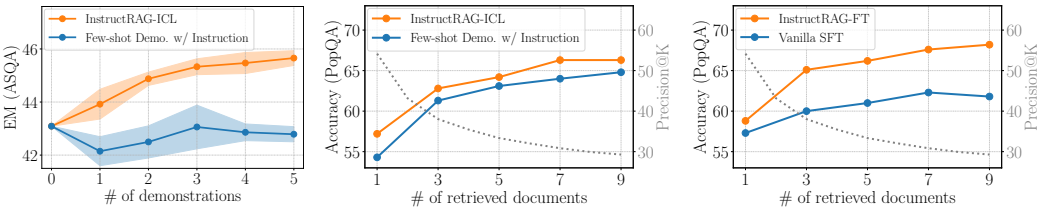
task, where our method slightly underperforms Self-RAG in terms of citation (*i.e.*, *pre* and *rec*). This is because our work primarily focuses on explicit denoising for RAG to improve the correctness of generations, which is measured by *em*. Despite not being explicitly optimized for citation metrics, our method still achieves competitive citation performance, significantly enhancing both generation accuracy and trustworthiness. Note that RetRobust achieves competitive performance on 2WikiMultiHopQA, which involves multi-hop reasoning. We attribute this to the additional training supervision provided by GPT-3, which enables the model to explicitly generate intermediate sub-queries and sub-answers. Another interesting finding is that Self-RAG consistently exhibits inferior performance compared to vanilla SFT, and even underperforms the training-free in-context RALM baseline across all benchmarks. We speculate the reason might be that these RAG tasks favor more domain-specific knowledge than general knowledge. However, it is challenging for Self-RAG to directly leverage in-domain features from existing training data as it requires GPT-4 to generate reflection tokens on these benchmarks, which is not available in our problem setting (§ 2.1).

3.3 ABLATION STUDY

Providing ground-truth answers and retrieved documents is important for rationale generation. As depicted in the first block of Table 4, we ablate the rationale generation design from two aspects: (1) *w/o ground-truth answer*, where the model has no access to the ground-truth answer during rational generation and must predict the answer and explain how it is derived solely based on retrieved documents; (2) *w/o retrieved documents*, where the model is not provided with any retrieved documents during rational generation, and in this case, it has to explain the given answer based on its own knowledge. Although it is not surprising that our default design consistently outperforms the two ablations, it is encouraging to find that our method still works well even without access to the retrieved documents or ground-truth answers. This finding suggests the great potential of our INSTRUCTRAG to operate in a fully unsupervised manner, which we believe is an exciting direction for future work.

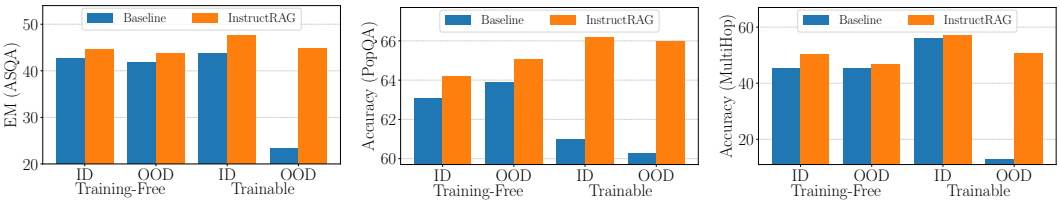
Larger rationale generator leads to better results. The middle block shows how different sizes of rationale generators impact the performance of our method. It is evident that the template-based rationale generation method significantly underperforms our method, highlighting the necessity of rationale generator. This is because the template-based method relies on pattern matching to identify relevant documents containing the ground-truth answer, which only considers lexical similarity while ignoring semantic meaning. The neglect of semantics inevitably introduces noise in template-generated rationales, making them less effective compared to rationales generated by LMs. Moreover, we also compare two variants of INSTRUCTRAG using Llama-3-Instruct$_{8B}$ and Llama-3-Instruct$_{70B}$ as rationale generators. The results show that the one with a 70B generator consistently outperforms its 8B counterpart in both training-free and trainable settings, indicating that the self-synthesized denoising rationales can provide better supervision when generated by stronger models.

Inference with demonstrations should only be applied to INSTRUCTRAG-ICL. In the bottom block, we study the use of demonstrations during the model inference. While demonstrations play



(a) Training-free RAG setting. (b) Training-free RAG setting. (c) Trainable RAG setting.

Figure 3: Impact of different number of demonstrations and retrieved documents. (a) Demonstration sensitivity study of INSTRUCTRAG-ICL. (b) Noise robustness study of INSTRUCTRAG-ICL. (c) Noise robustness study of INSTRUCTRAG-FT.



(a) Short-form to long-form QA. (b) Long-form to short-form QA. (c) Single-hop to multi-hop QA.

Figure 4: Generalizing INSTRUCTRAG from source domain task to target domain task, where ID and OOD denote in-domain and out-of-domain settings. (a) PopQA (short-form QA task) as source domain and ASQA (long-form QA task) as target domain. (b) ASQA as source domain and PopQA as target domain. (c) PopQA (single-hop QA task) as source domain and 2WikiMultiHopQA (multi-hop QA task) as target domain. We adopt *few-shot demonstration with instruction* and *vanilla supervised fine-tuning* as the training-free and trainable baselines.

an important role for INSTRUCTRAG-ICL, we find that they actually hurt the performance of INSTRUCTRAG-FT. We attribute this to the fact that INSTRUCTRAG-FT is optimized to directly generate denoising rationales given potentially noisy input, without referring to any demonstrations. Therefore, providing in-context demonstrations for INSTRUCTRAG-FT is redundant and may compromise its capability due to the discrepancy between training and inference.

3.4 ANALYSIS

INSTRUCTRAG-ICL consistently benefits from more demonstrations. Figure 3a shows the demonstration sensitivity of INSTRUCTRAG-ICL and the few-shot demonstration with instruction baseline. It is interesting to find that the baseline method achieves its best performance with only one demonstration, and presenting more demonstrations actually harms its performance. In contrast, our method consistently improves with the increasing number of demonstrations, confirming the superiority of self-synthesized rationales over plain answers in terms of denoising.

INSTRUCTRAG-ICL and INSTRUCTRAG-FT are robust to increased noise ratios. Figure 3b and Figure 3c show the generation accuracy of INSTRUCTRAG-ICL and INSTRUCTRAG-FT and the corresponding retrieval precision under an increasing number of retrieved documents. While retrieving more documents provides richer external knowledge to the RAG model, it also introduces more noise and lowers the retrieval precision. As a result, both the training-free and trainable baselines show diminishing improvements or even degrade as the number of documents increases, reflecting their vulnerability to high noisy ratios. In contrast, our INSTRUCTRAG-ICL and INSTRUCTRAG-FT are not negatively affected by this increased noise ratio but rather gain further improvement, demonstrating their robust denoising ability.

INSTRUCTRAG-ICL and INSTRUCTRAG-FT generalize well to unseen tasks. Figure 4 demonstrates the generalization ability of our method in both training-free and trainable settings. For the in-domain (ID) method, it directly utilizes target domain demonstrations (in training-free settings) or is trained on the target domain task (in trainable settings). In contrast, the out-of-domain (OOD) method can only learn from demonstrations or training data in the source domain, and have no prior knowledge of the target domain. In this case, the model must leverage the knowledge learned from the source domain task to solve the unseen target domain task. The results show that our

Table 5: (a) Transfer from the source QA task (PopQA) to the target code generation task (HumanEval). Our method INSTRUCTRAG-FT is fine-tuned only on the source task and is evaluated on the unseen target task. We compare it with off-the-shelf LLaMA-3-8B-Instruct using the standard metrics pass@k (Chen et al., 2021), in both non-retrieval and retrieval-augmented generation settings. (b) Evaluation with GPT-4o as the judge. Compared to pattern-matching based metrics, it allows the judge to consider semantic equivalence and is expected to yield a more fair evaluation.

Method	pass@1	pass@10
Without Retrieval		
Llama-3-8B-Instruct	58.5	64.6
INSTRUCTRAG-FT	60.4	65.2
With Retrieval		
Llama-3-8B-Instruct	59.8	69.5
INSTRUCTRAG-FT	64.6	71.3

(a) Transfer from QA task to code generation task.

Method	Pattern-based	LLM-based
RAG w/o Training		
In-Context RALM	56.8	64.5
INSTRUCTRAG-ICL	62.1	67.6
RAG w/ Training		
Vanilla SFT	56.6	65.1
INSTRUCTRAG-FT	65.7	69.7

(b) Evaluation with GPT-4o as the judge.

method consistently outperforms the baselines across various scenarios in both in-domain and out-of-domain settings, demonstrating strong task generalizability. One counter-intuitive finding is that in the scenario of generalizing from long-form to short-form QA task (Figure 4b), the training-free OOD method substantially outperforms its in-domain counterpart. We speculate that the training-free OOD method achieves better performance because it benefits from the demonstrations with long answers from the source domain (ASQA). The reason is that the questions in ASQA are ambiguous and can have multiple interpretations, and ground-truth long answers often address the questions from various perspectives, which can be regarded as a form of chain-of-thought demonstration.

Furthermore, we also study the generalizability of INSTRUCTRAG to a non-QA knowledge-intensive task such as code generation. As presented in Table 5a, we directly apply INSTRUCTRAG-FT trained on the QA task (PopQA) to solve the unseen code generation task (HumanEval (Chen et al., 2021)), following the CodeRAG-Bench setup (Wang et al., 2024c). We evaluate the code generation performance using the standard pass@k metric and compare our method with the off-the-shelf Llama-3-8B-Instruct as the baseline. It can be observed that our method consistently achieves better generalization performance in the unseen code generation task in both non-retrieval and RAG settings. This finding aligns with our observation that INSTRUCTRAG trained on QA tasks tends to generate more text-based comments that articulate the design of coding solutions compared to the off-the-shelf Llama-3-8B-Instruct, thereby leading to more accurate code generation.

Evaluation with LLM-as-a-judge. Despite being standard evaluation metrics for question-answering, accuracy or exact match are known to be imperfect (Cuconasu et al., 2024) as they mainly rely on pattern-matching to judge whether the predicted answer aligns with the ground-truth answer. Such metrics cannot handle cases where the predicted answer and ground-truth answer are synonyms (*e.g.*, "Donald Trump" vs "Donald J. Trump" cannot be correctly recognized as a match), leading to biased evaluation results. Therefore, we adopt the LLM-as-a-judge (Bubeck et al., 2023; Zheng et al., 2024b) approach to evaluate whether the model response aligns with the ground-truth answer using GPT-4o (OpenAI, 2024), which allows the judge to consider semantic equivalence and is expected to yield a more fair evaluation. As shown in Table 5b, we evaluate our method and baseline models on the open-domain Natural Questions benchmark in both training-free and trainable RAG settings. Compared to pattern-matching based metrics, LLM-as-a-judge generally leads to higher evaluation results, mostly due to its capability to accurately match semantically equivalent phrasings. Notably, our method consistently outperforms baselines under both pattern-matching based and LLM-based evaluation metrics, further validating the effectiveness of INSTRUCTRAG.

4 RELATED WORK

4.1 RETRIEVAL-AUGMENTED GENERATION

Retrieval-augmented generation (RAG) is a widely adopted approach to enhance large language models (LLMs) with external knowledge (Asai et al., 2023a; 2024; Borgeaud et al., 2022; Gao et al., 2023b; Guu et al., 2020; Khandelwal et al., 2019; Lewis et al., 2020; Ram et al., 2023; Shi et al., 2023), demonstrating promising potential to reduce hallucinations and enhance the generation accuracy of LLMs across various real-world applications (Chase, 2022; Jin et al., 2024b; Liu, 2022; Lu

et al., 2022; Siriwardhana et al., 2023; Tan et al., 2024; Zhou et al., 2022). Recently, a growing research effort has been devoted to enhancing RAG from various aspects, such as improving decoding efficiency (Merth et al., 2024; Jin et al., 2024a; Liu et al., 2023; Wang et al., 2024b), exploring long-context retrieval (Xu et al., 2024a; Yen et al., 2024), compressing prompts (Jiang et al., 2023a; Xu et al., 2023; Cheng et al., 2024), and addressing practical concerns such as adversarial retrieval (Xiang et al., 2024; Zhong et al., 2023; Zou et al., 2024) and privacy leakage (Huang et al., 2023b; Zeng et al., 2024). Despite their advantages, these RAG systems inevitably suffer from irrelevant information introduced by imperfect retrievers or noisy retrieval corpora. However, most existing works typically address this issue by improving the retrieval quality and reducing noise exposure to the model (Gupta et al., 2024; Jiang et al., 2024a; Sarthi et al., 2024; Wang et al., 2024a; Yan et al., 2024; Yang et al., 2024; Zhang et al., 2024a;b). Notable methods include adaptive retrieval (Asai et al., 2023b; Jiang et al., 2023b; Yao et al., 2022) and query rewriting (Chan et al., 2024; Mao et al., 2024). In contrast, our work focuses on an orthogonal direction of developing explicit denoising methods for RAG, thereby enhancing the model's noise robustness and generation accuracy, even in highly noisy contexts.

4.2 Eliciting Reasoning in Large Language Models

Recent studies have extensively explored the reasoning capability of LMs, but typically not in the context of RAG where potentially noisy retrieved contents may mislead the reasoning if not properly addressed. Chain-of-thought (CoT) prompting (Wei et al., 2022) is an effective method to elicit step-by-step reasoning from LMs by showing exemplars with detailed explanations (*i.e.*, rationales (Feng et al., 2024; Lampinen et al., 2022; Rajani et al., 2019; Zelikman et al., 2024)) that lead to the final answer. However, such works often requires manually crafted demonstrations (Wang et al., 2022; Xu et al., 2024b), which is costly and requires extensive efforts and domain knowledge (Zheng et al., 2024a). To mitigate this limitation, automatic chain-of-thought prompting (Auto-CoT) (Zhang et al., 2022) is introduced to automatically select instances from the corpus coupled with zero-shot CoT (Kojima et al., 2022), where the rationales are generated by LMs. Furthermore, it has been shown that CoT reasoning can be elicited even without explicit prompting, particularly for instruction-tuned LMs (Wang & Zhou, 2024). Another related work shows rationales generated by small models can help large models reason better (Lee et al., 2024). Although rationalization has been extensively investigated in many NLP tasks (Chen et al., 2022; Ghoshal et al., 2022; Paranjape et al., 2020; Wiegreffe et al., 2021), none of them are designed for RAG, and how to leverage the instruction-following abilities of LMs for explicit denoising in the context of RAG still remains underexplored.

5 Conclusion

In this work, we presented InstructRAG, a simple retrieval-augmented generation (RAG) approach that explicitly denoises retrieved contents and produces accurate generations. By leveraging the strong instruction-following abilities of large language models, InstructRAG generates detailed rationales that articulate how the ground-truth answers can be derived from the retrieved documents. These synthetic rationales can serve as either in-context learning examples or supervised fine-tuning data, enabling the model to learn an explicit denoising process. Experiments on five knowledge-intensive benchmarks show InstructRAG consistently outperforms state-of-the-art RAG approaches with significant improvements in both training-free and trainable settings. Compared to the best baseline method, InstructRAG achieves an average improvement of 8.3% across all benchmarks, demonstrating its effectiveness in enhancing the noise robustness of retrieval-augmented generation. Limitations and future work are discussed in Appendix A.

Reproducibility Statement

To ensure the highest level of reproducibility for our reported results, we have provided:

- Complete source code, accessible via an anonymous link: `https://anonymous.4open.science/r/InstructRAG-ICLR2025-Submission`;
- Comprehensive implementation details in Appendix B;
- All prompt templates used in our experiments in Appendix D.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023b.

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

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

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

Harrison Chase. LangChain, 2022. URL https://github.com/langchain-ai/langchain.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3792–3805, 2022.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17754–17762, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. *arXiv preprint arXiv:2405.13792*, 2024.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 719–729, 2024.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

Yunlong Feng, Yang Xu, Libo Qin, Yasheng Wang, and Wanxiang Che. Improving language model reasoning with self-motivated learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 8840–8852, 2024.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, pp. 1–79, 2024.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023a.

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

Asish Ghoshal, Srinivasan Iyer, Bhargavi Paranjape, Kushal Lakhotia, Scott Wen-tau Yih, and Yashar Mehdad. Quaser: Question answering with scalable extractive rationalization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1208–1218, 2022.

Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Aman Gupta, Anup Shirgaonkar, Angels de Luis Balaguer, Bruno Silva, Daniel Holstein, Dawei Li, Jennifer Marsman, Leonardo O Nunes, Mahsa Rouzbahman, Morris Sharp, et al. RAG vs Fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv preprint arXiv:2401.08406*, 2024.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023a.

Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. Privacy implications of retrieval-based language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14887–14902, 2023b.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

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

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023a.

Wenqi Jiang, Shuai Zhang, Boran Han, Jie Wang, Bernie Wang, and Tim Kraska. PipeRAG: Fast retrieval-augmented generation via algorithm-system co-design. *arXiv preprint arXiv:2403.05676*, 2024a.

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

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. Instruction-tuned language models are better knowledge learners. *arXiv preprint arXiv:2402.12847*, 2024b.

Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Xin Liu, Xuanzhe Liu, and Xin Jin. RAGCache: Efficient knowledge caching for retrieval-augmented generation. *arXiv preprint arXiv:2404.12457*, 2024a.

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, 2017.

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

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. RealTime QA: What's the answer right now? *Advances in Neural Information Processing Systems*, 36, 2024.

Noriaki Kawamae. Friendly conditional text generator. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 420–428, 2023.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*, 2022.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

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. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276.

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

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 537–563, 2022.

Jooyoung Lee, Fan Yang, Thanh Tran, Qian Hu, Emre Barut, and Kai-Wei Chang. Can small language models help large language models reason better?: LM-guided chain-of-thought. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2835–2843, 2024.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1774–1793, 2023.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: An easy-to-use Python toolkit to support replicable IR research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*, 2021.

Jerry Liu. LlamaIndex, 2022. URL https://github.com/jerryjliu/llama_index.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12, 2024.

Yuhan Liu, Hanchen Li, Kuntai Du, Jiayi Yao, Yihua Cheng, Yuyang Huang, Shan Lu, Michael Maire, Henry Hoffmann, Ari Holtzman, et al. CacheGen: Fast context loading for language model applications. *arXiv preprint arXiv:2310.07240*, 2023.

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

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 157–165, 2022.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. Hallucination-free? Assessing the reliability of leading AI legal research tools. *arXiv preprint arXiv:2405.20362*, 2024.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 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)*, pp. 9802–9822, 2023.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. RaFe: Ranking feedback improves query rewriting for RAG. *arXiv preprint arXiv:2405.14431*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

Thomas Merth, Qichen Fu, Mohammad Rastegari, and Mahyar Najibi. Improving and accelerating retrieval-augmented generation with superposition prompting. In *Forty-first International Conference on Machine Learning*, 2024.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022.

OpenAI. Hello GPT-4o. 2024. URL https://openai.com/index/hello-gpt-4o.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1938–1952, 2020.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4932–4942, 2019.

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

Stephen E Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pp. 232–241. Springer, 1994.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3784–3803, 2021.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17, 2023.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288, 2022.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*, 2024.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (LLM)? A.K.A will LLMs replace knowledge graphs? *arXiv preprint arXiv:2308.10168*, 2023.

Hanzhuo Tan, Qi Luo, Ling Jiang, Zizheng Zhan, Jing Li, Haotian Zhang, and Yuqun Zhang. Prompt-based code completion via multi-retrieval augmented generation. *arXiv preprint arXiv:2405.07530*, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. FreshLLMs: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*, 2023.

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. *arXiv preprint arXiv:2402.10200*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. RAT: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*, 2024a.

Zilong Wang, Zifeng Wang, Long Le, Huaixiu Steven Zheng, Swaroop Mishra, Vincent Perot, Yuwei Zhang, Anush Mattapalli, Ankur Taly, Jingbo Shang, et al. Speculative rag: Enhancing retrieval augmented generation through drafting. *arXiv preprint arXiv:2407.08223*, 2024b.

Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. Coderag-bench: Can retrieval augment code generation? *arXiv preprint arXiv:2406.14497*, 2024c.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Sarah Wiegreffe, Ana Marasović, and Noah A Smith. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10266–10284, 2021.

Kevin Wu, Eric Wu, and James Zou. How faithful are RAG models? quantifying the tug-of-war between RAG and LLMs' internal prior. *arXiv preprint arXiv:2404.10198*, 2024.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. Certifiably robust RAG against retrieval corruption. *arXiv preprint arXiv:2405.15556*, 2024.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. Lawformer: A pre-trained language model for Chinese legal long documents. *AI Open*, 2:79–84, 2021.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*, 2024.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*, 2023.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*, 2024a.

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

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing, 2024c.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024d.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Ke Yang, Charles Yu, Yi R Fung, Manling Li, and Heng Ji. Adept: A debiasing prompt framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10780–10788, 2023.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. CRAG – comprehensive RAG benchmark. *arXiv preprint arXiv:2406.04744*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. *arXiv preprint arXiv:2402.16617*, 2024.

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

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*, 2023.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, et al. The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG). *arXiv preprint arXiv:2402.16893*, 2024.

Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. RATT: A thought structure for coherent and correct LLM reasoning. *arXiv preprint arXiv:2406.02746*, 2024a.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023.

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

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5823–5840, 2023.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*, 2024a.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024b.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13764–13775, 2023.

Shuyan Zhou, Uri Alon, Frank F Xu, Zhengbao Jiang, and Graham Neubig. DocPrompting: Generating code by retrieving the docs. In *The Eleventh International Conference on Learning Representations*, 2022.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. PoisonedRAG: Knowledge poisoning attacks to retrieval-augmented generation of large language models. *arXiv preprint arXiv:2402.07867*, 2024.

A LIMITATIONS AND FUTURE WORKS

Limitations. In this work, we mainly conduct experiments on question answering-type tasks, and it remains unclear how our method may generalize to other scenarios (*e.g.*, open-ended generation). Moreover, despite being the standard evaluation metrics, both accuracy and exact match are biased and cannot perfectly reflect the quality of the model's generations. For instance, such metrics heavily rely on string matching, which assesses correctness at the lexical level rather than the semantic level, thereby failing to recognize different phrasings that convey identical meanings. The evaluation results also suffer from length bias, as longer generations tend to achieve higher accuracy. Exploring more advanced metrics like using LLMs as judges would better evaluate RAG model generations (Yang et al., 2024). Another potential limitation is that our model might be subject to sample bias in the training data. Incorporating bias mitigation methods (Gallegos et al., 2024; Kawamae, 2023; Yang et al., 2023) would be helpful for further improving our work.

Future Work. Future research directions include exploring more advanced techniques for generating high-quality rationales, such as incorporating domain-specific knowledge or leveraging multi-task learning to enable better generalization across various tasks. For instance, although the consistency ratio between synthetic rationales and ground-truth answers on training samples with at least one relevant document achieves 98%, the overall consistency ratio on all training samples is only 89%. This is because for some samples, none of the retrieved documents is relevant to the question, which significantly compromises the quality of the generated rationales. Therefore, it will be interesting to fully explore the potential of our method by incorporating additional designs such as a filtering mechanism, which we leave as future work. It will also be interesting to evaluate the model performance under long-context settings with a dynamic or extremely large number of retrieved documents. Finally, integrating our method with other advanced retrieval techniques (Su et al., 2024), such as active retrieval, could potentially lead to even better performance on knowledge-intensive tasks.

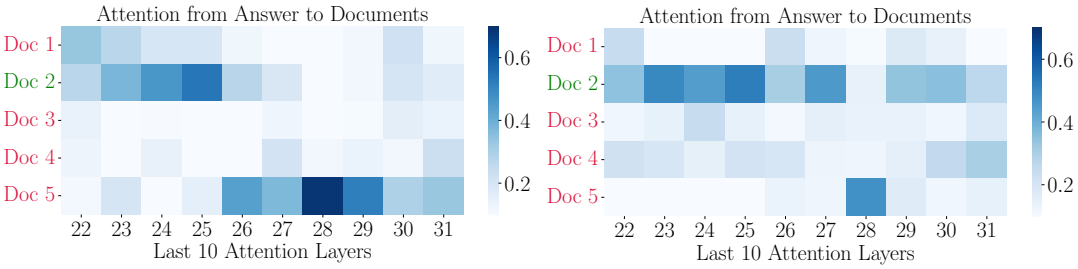
B IMPLEMENTATION DETAILS

Retrieval setup. Following (Asai et al., 2023b; Ram et al., 2023), we use the Wikipedia dump from (Karpukhin et al., 2020) as the external retrieval corpus for all five benchmarks studied in this work, where each document is a disjoint text block of up to 100 words extracted from a Wikipedia article. We compared all RAG methods under a diverse retrieval environment with various sparse and dense retrievers and number of retrieved documents. Specifically, we use Contriever-MS MARCO as the retriever for PopQA and TriviaQA, DPR for Natural Questions, GTR for ASQA, and BM25 for 2WikiMultiHopQA. By default, we retrieve the top 5 documents from the retrieval corpus for each query in all tasks except 2WikiMultiHopQA, where the top 10 documents are retrieved. We use the official weights for all dense retrievers and the implementation from Pyserini (Lin et al., 2021) for the sparse retriever BM25.

Training details. Our models are trained on 4 Nvidia H100 GPUs with 80GB memory via full-parameter fine-tuning. We use fully sharded data parallelism (FSDP) for distributed training, along with FlashAttention (Dao, 2023) and bf16 mixed precision training enabled for computation efficiency. By default, all models are trained using the Adam optimizer (Kingma & Ba, 2014) for 2 epochs, with a batch size of 128, a learning rate of 2.5e-5, and a cosine learning rate schedule with 3% warmup steps. For the trainable baseline vanilla SFT, we use a slightly different learning rate of 2e-5 based on our hyper-parameter search results. To fairly compare with Self-RAG and RetRobust, we re-implement them using Llama-3-Instruct-8B. We also optimize their performance through an extensive hyper-parameter search with learning rates in [8e-6, 1e-5, 2e-5] and training epochs in [1, 2, 3]. For Self-RAG, we use a learning rate of 1e-5 with a single training epoch. For RetRobust, we use a learning rate of 2e-5 with two training epochs. The only exception is the training for RetRobust on 2WikiMultiHopQA, where we train the model for 5 epochs on the augmented training set released by the original authors. The maximum token length for all models is fixed at 4096.

Inference details. By default, the number of demonstrations used in INSTRUCTRAG-ICL and the baseline method few-shot demonstration with instruction is set to be 2. We use vLLM (Kwon et al., 2023) to load models for memory-efficient inference and adopt the greedy decoding strategy for model generation.

C CASE STUDY



(a) Vanilla SFT. (b) INSTRUCTRAG-FT.

Figure 5: Visualization of model attention from answer to retrieved documents on a random sample from the ASQA task, where Doc 2 is the only relevant document that contains the correct answer.

Question: Who played the admiral on Star Trek: Discovery?
Answer: Katrina Cornwell, portrayed by **Jayne Brook**, is a Starfleet admiral.
Contexts: Document 1 (Title: Star Trek: Insurrection):…Document 2 (Title: Star Trek: Discovery (season 1)):…Document 3 (Title: Star Trek):… Document 4 (Title: Brian Thompson):…Document 5 (Title: Star Trek: Enterprise):…

In-Context RALM

According to Document 2, there is no mention of an Admiral character in Star Trek: Discovery. However, according to Document 3, Captain Gabriel Lorca (played by Jason Isaacs) is the commander of the USS Discovery.

❌ **Wrong Answer**

Vanilla SFT

Star Trek: Discovery is an American television series created by Bryan Fuller and Alex Kurtzman for the streaming service CBS All Access. The series is set about a decade before the events of the original Star Trek series and follows the crew of the USS Discovery as they attempt to stop a massive Klingon armada while uncovering the truth about the Red Angel and Section 31. The character of Admiral Cornwell is played by Jayne Brook. The character of Admiral Cornwell is a Starfleet officer who serves as the first officer of the USS Enterprise NCC-1701-D. The character of Admiral Cornwell is played by Elizabeth Maxwell.

❌ **Confusing Answer**

InstructRAG-FT

The documents that are useful to answer the question "Who played the admiral on Star Trek Discovery?" are: Document 1: Star Trek: Insurrection, Document 3: Star Trek, Document 5: Star Trek: Enterprise. These documents provide information about the casting of actors in Star Trek productions, which is relevant to the question. From Document 1, we learn that Anthony Zerbe played the role of Admiral Matthew Dougherty in Star Trek: Insurrection. From Document 3, Star Trek: Discovery is a prequel to the Star Trek franchise and premiered in 2017. From Document 5, we learn that Vaughn Armstrong played Admiral Maxwell Forest in Star Trek: Enterprise. However, none of these documents mention the specific admiral on Star Trek Discovery, which is Admiral Cornwell, played by Jayne Brook. The correct answer is not found in the provided documents, but it can be answered based on external knowledge. Jayne Brook played the role of Admiral Cornwell in Star Trek: Discovery.

✅ **Correct Answer**

Figure 6: A case study of INSTRUCTRAG-FT compared with in-context RALM and vanilla SFT. The red texts denote irrelevant or inaccurate model generations, while the green texts denote contents relevant to the question. This study shows that our model can effectively identify relevant information from noisy input and leverage its own knowledge to correctly answer questions when required.

Attention visualization. To intuitively understand the denoising process of our INSTRUCTRAG, we visualize its attention from the answer to retrieved documents. As pointed out by a recent work (Yu et al., 2023), only attention distributions from deep layers can accurately reflect the LM's retrieval behavior and focus on key information, while attention from shallow layers usually do not imply meaningful patterns. Therefore, we only plot the attention weights of the last 10 layers (Layer 22 to Layer 31). As presented in Figure 5, our model accurately identifies the only benign document from noisy input, showing a strong denoising signal compared to vanilla SFT.

Generation comparison. Figure 6 compares the generated responses of in-context RALM, vanilla SFT, and our INSTRUCTRAG-FT for an actual question from the ASQA task. Among them, only our method can correctly answer this question while providing comprehensive denoising details. Specifically, it first identifies potentially relevant documents from noisy inputs, and then lays out the candidate information. More encouragingly, we find that INSTRUCTRAG-FT is able to refer to its own parametric knowledge when no relevant document is present in the context after denoising, demonstrating its superiority over existing RAG approaches.

D PROMPT TEMPLATES

In this work, we instantiate the proposed INSTRUCTRAG with off-the-shelf instruction-tuned LMs (*i.e.*, meta-llama/Meta-Llama-3-8B-Instruct and meta-llama/Meta-Llama-3-70B-Instruct), and apply the official Meta-Llama-3-Instruct chat template (marked in gray) in all prompts.

Rationale generation. Below are the prompt templates for rationale generation used for all five benchmarks in our work. Table 6 shows the rationale template used in the ablation study (§ 3.3). For simplicity, we use the same prompt structure (Table 7) for all tasks with minor differences in task-specific instructions (Table 8).

Table 6: Rationale template used in ablation study.

Rationale Template
Positive Template: After reviewing the provided document, I found that only documents {documents} contain relevant information to answer the question. Based on my knowledge and the provided contents, the answer is: {answer}. **Negative Template:** After reviewing the provided document, I found that none of them contain relevant information to answer the question. Based on my knowledge and the provided contents, the answer is: {answer}.

Table 7: Rationale generation prompt template.

Rationale Generation
Input: <|begin_of_text|><|start_header_id|>user<|end_header_id|> Read the following documents relevant to the given question: {question} Document [1] (Title: $\cdots$): {contents} $\cdots$ Please identify documents that are useful to answer the given question: "{question}", and explain how the contents lead to the answer: {answer}. If none of the documents is aligned with the answer, in that case, you have to explain the answer only based on your own knowledge, without referring to the provided information. **{task-specific instruction}**<|eot_id|><|start_header_id|>assistant<|end_header_id|> **Output:** {rationale}

Table 8: Task-specific instruction used in rationale generation prompt.

Task-specific Instruction for Rationale Generation
ASQA: Note that the question may be ambiguous and have multiple correct answers. Make sure your response includes all correct answers and provides clear reasoning details followed by a concise conclusion. **PopQA:** Note that the question mainly asks about the object entity that holds a certain relationship with the given subject entity. There may be multiple correct answers. Make sure your response includes all correct answers and provides clear reasoning details followed by a concise conclusion. **TriviaQA / Natural Questions / 2WikiMultiHopQA:** Note that the question may be compositional and require intermediate analysis to deduce the final answer. Make sure your response is grounded and provides clear reasoning details followed by a concise conclusion.

Inference prompts. Below we present the inference prompts for both training-free and trainable RAG methods used in this work, including in-context RALM (Table 9), few-shot demonstrations with instruction (Table 10), and vanilla supervised fine-tuning (Table 11). Note that for a fair comparison, the inference prompt for our INSTRUCTRAG-FT is exactly the same as vanilla SFT. Similarly, the inference prompt for INSTRUCTRAG-ICL shares the same inference instruction as the few-shot demonstrations with instruction. The only difference between the prompts of these two methods lies in the demonstrations where INSTRUCTRAG-ICL employs denoising question-rationale $\langle q, r\rangle$ pairs, while few-shot demonstrations with instruction uses plain question-answer $\langle q, a\rangle$ pairs.

Table 9: Inference prompt for In-Context RALM.

In-Context RALM

Input: <|begin_of_text|><|start_header_id|>user<|end_header_id|>
Document [1] (Title: $\cdots$): {contents}
$\cdots$
Based on your knowledge and the provided information, answer the question: {question}
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Output: {answer}

Table 10: Inference prompt for INSTRUCTRAG-ICL and few-shot demonstrations with instruction.

INSTRUCTRAG-ICL / Few-shot Demonstrations with instruction

Input: <|begin_of_text|><|start_header_id|>user<|end_header_id|>
Your task is to analyze the provided documents and answer the given question. Please generate a brief explanation of how the contents of these documents lead to your answer. If the provided information is not helpful to answer the question, you only need to respond based on your own knowledge, without referring to the documents.

Below are some examples of how to answer the question:
{example question q_1}
{example answer a_1 / rationale r_1}
$\cdots$
Document [1] (Title: $\cdots$): {contents}
$\cdots$
Based on your knowledge and the provided information, answer the question: {question}
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Output: {answer}

Table 11: Inference prompt for INSTRUCTRAG-FT and vanilla supervised fine-tuning.

INSTRUCTRAG-FT / Vanilla Supervised Fine-tuning

Input: <|begin_of_text|><|start_header_id|>user<|end_header_id|>
Document [1] (Title: $\cdots$): {contents}
$\cdots$
Based on your knowledge and the provided information, answer the question: {question}
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Output: {answer}

E EXAMPLE OF LLM-AS-A-JUDGE

To measure the quality of model-generated rationales, we employ the LLM-as-a-judge approach for a more comprehensive evaluation of the model's outputs. As illustrated in Table 12, if the rationale is inaccurate despite the final answer being correct (probably due to the use of the LLM's parametric knowledge), the LLM judge will detect this inconsistency.

Table 12: Evaluate rationale with LLM-as-a-judge.

Evaluation with LLM-as-a-judge

Input: <|begin_of_text|><|start_header_id|>user<|end_header_id|>
You are a question-answering evaluator. Given the question, please judge whether the rationale aligns with the true answer.

{"question": "In what city was Catherine Tishem born?",
"true_answer": "Norwich, England",
"rationale": "The documents that are useful to answer the question are: Documents 4 and 5. Document 4 states that Catherine Tishem was from Breda, and her husband Wouter Gruter co-signed the Compromise of Nobles in 1566, and they moved to the Dutch Calvinist exile community of Norwich. This implies that Catherine Tishem was not born in Norwich. Document 5 provides more information about Catherine Tishem's life, stating that she was an erudite woman from Antwerp who educated her son, Jan Gruter, while in exile in England. This implies that Catherine Tishem was born in Antwerp. The other documents do not provide any information about Catherine Tishem's birthplace, so they are not useful for answering this question. Based on the above information, the answer is Norwich."}
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Output: The predicted answer concludes that Catherine Tishem was born in Norwich, which contradicts the conclusion drawn from the reasoning in the documents, where Antwerp is suggested as her birthplace based on the information in Document 5. Thus, while the predicted conclusion is Norwich, the rationale suggests Antwerp as the more likely birthplace based on the information in the documents. Therefore, the predicted answer does not align with the true answer.