

# ALR<sup>2</sup>: A RETRIEVE-THEN-REASON FRAMEWORK FOR LONG-CONTEXT QUESTION ANSWERING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

The context window of large language models (LLMs) has been extended significantly in recent years. However, while the context length that the LLM can process has grown, the capability of the model to accurately reason over that context degrades noticeably. This occurs because modern LLMs often become overwhelmed by the vast amount of information in the context; when answering questions, the model must identify and reason over relevant evidence sparsely distributed throughout the text. To alleviate the challenge of long-context reasoning, we develop a retrieve-then-reason framework, enabling LLMs to reason over relevant evidence collected during an intermediate retrieval step. We find that modern LLMs struggle to accurately retrieve relevant facts and instead, often hallucinate “*retrieved facts*”, resulting in flawed reasoning and the production of incorrect answers. To address these issues, we introduce ALR<sup>2</sup>, a method that augments the long-context reasoning capability of LLMs via an explicit two-stage procedure, i.e., aligning LLMs with the objectives of both retrieval and reasoning. We demonstrate the efficacy of ALR<sup>2</sup> for mitigating performance degradation in long-context reasoning tasks. Through extensive experiments on long-context QA benchmarks, we find our method to outperform competitive baselines by large margins, achieving at least 8.4 and 7.9 EM gains on the long-context versions of HotpotQA and SQuAD datasets, respectively.

## 1 INTRODUCTION

The ability for large language models (LLMs) to reason over long contexts is critical in many downstream tasks, e.g., document analysis (Wang et al., 2024a), multi-hop tool use (Yao et al., 2023), agents with long history (Park et al., 2023), etc. Significant efforts have been made to extend the effective context length of LLMs, ranging from investigating how different modules in Transformer (Vaswani, 2017) architecture impact long-context performance (Su et al., 2024a; Chiang & Cholak, 2022; Xiao et al., 2024) to proposing new architectures with improved efficiency (Dai et al., 2019; Beltagy et al., 2020; Chevalier et al., 2023).

While these developments are promising, in our preliminary study, we show that the long-context performance of LLMs varied significantly across different tasks. In particular, we design a series of experiments to evaluate the performance of LLMs on long-context retrieval<sup>1</sup> and reasoning (Figure 1). We observe that, when tasked to generate answers by *directly* reasoning over the full context, performance degrades as the input context grows. In contrast, when tasked with retrieving the set of evidence relevant to the question, the performance of LLMs is only mildly affected by the growth of the input context.

Those findings motivate us to explicitly leverage an intermediate retrieval step to address the challenge of LLMs in long-context reasoning. In our proposed retrieve-then-reason framework, an LLM functions as both a retriever and a reasoner. When tasked to answer a question, it first collects the relevant evidence sparsely distributed across the context, and then derives the final answer by reasoning over that collected evidence. We also find that modern LLMs struggle to accurately retrieve “relevant facts”, i.e., simply prompting them to retrieve supporting evidence often results in severe

<sup>1</sup>Following the definition in prior work (Chen et al., 2023; Hsieh et al., 2024), i.e., retrieval is performed by generating the relevant information from a given input text.

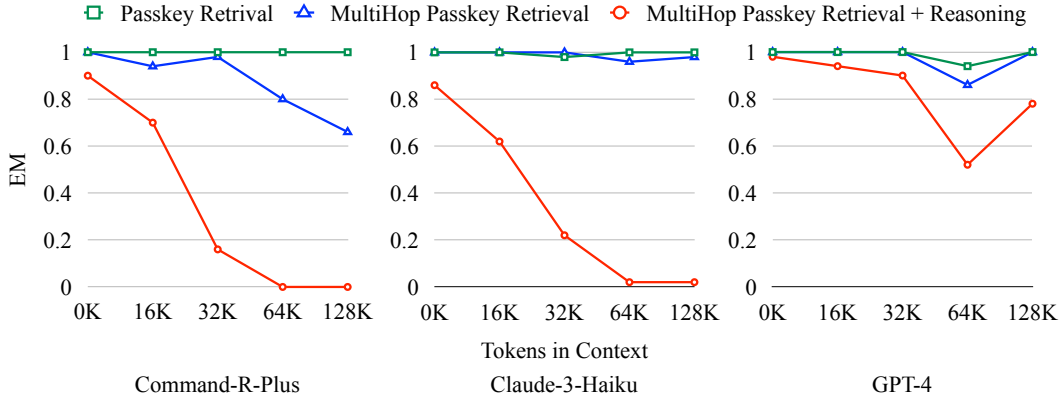


Figure 1: Performance of LLMs on three increasingly challenging long-context tasks. The x-axis represents the number of tokens in the long context, while the y-axis indicates the exact match score.

hallucinations, which leads to flawed reasoning by the LLMs caused by the reliance on inaccurate information. To this end, we introduce the ALR<sup>2</sup> approach, which augments the long-context reasoning capability of LLMs via an explicit two-stage procedure, i.e., ALigning LLMs with the objectives of both Retrieval and Reasoning.

Extensive experiments on long-context single-hop and multi-hop question answering tasks demonstrate the effectiveness of our ALR<sup>2</sup> approach. We compare ALR<sup>2</sup> against various advanced prompting techniques, including direct-answering and other popular prompting methods for long-context reasoning (Weston & Sukhbaatar, 2023; Lee et al., 2024). Results show that ALR<sup>2</sup> outperforms these approaches by at least 23.4 and 12.7 EM scores on the long-context versions of HotpotQA (Yang et al., 2018) and SQuAD (Rajpurkar et al., 2016) benchmarks. In addition, we also compare ALR<sup>2</sup> with LLMs fine-tuned with direct-answering prompting and observe ALR<sup>2</sup> excelling at 8.4 and 7.9 EM gains on HotpotQA and SQuAD, respectively. Furthermore, we conduct extensive analysis revealing that ALR<sup>2</sup> displays favourable generalization capability and significantly mitigates retrieval issues, e.g. hallucinations and low accuracy, that are prevalent in prompting-based approaches (§5).

Our contributions are three-fold:

- In our preliminary study, we show that as the input context length grows, LLM reasoning performance degradation is higher than that of retrieval.
- We propose ALR<sup>2</sup>, a retrieve-then-reason approach to alleviate the challenges occurred when LLMs reasoning over long-contexts by aligning LLMs with both retrieval and reasoning objectives. ALR<sup>2</sup> substantially mitigates the primary difficulties of long-context reasoning, such as hallucination and low retrieval accuracy.
- We conduct extensive experiments to systematically compare our approach with baseline methods, including both prompting-based and fine-tuning-based techniques. Our results show that ALR<sup>2</sup> significantly outperforms these baselines on long-context QA tasks, i.e., achieving at least 8.4 and 7.9 EM gains on long-context version of the HotpotQA and SQuAD benchmarks, respectively.

## 2 PRELIMINARY STUDY

In this section, we will present the challenges of long-context reasoning and the motivation for our work, drawn from observations on a series of long-context tasks.

**Setup** We assess the long-context capabilities of LLMs using a series of increasingly challenging tasks, inspired by prior work (Chen et al., 2023; Hsieh et al., 2024):

- **Task 1: Passkey Retrieval:** A sentence, such as "The passkey of Alice is KEY," is inserted into the long context, and the LLM is asked to retrieve Alice's passkey. The "KEY" is a randomly generated string of 10 characters, sampled from the alphabet (A-Z).

- **Task 2: Multi-hop Passkey Retrieval:** In this task, the 10-character passkey is split into two halves, with each half randomly assigned to “Alice” and “Bob.” The LLM is then prompted to retrieve either Alice’s or Bob’s passkey. This task requires an LLM to identify the most relevant one from multiple passkeys, making it more complex than the first.
- **Task 3: Multi-hop Passkey Retrieval + Reasoning:** This task involves both retrieval and reasoning, and Task 1 & 2 are the pre-requisite tasks, i.e., the passkey retrieval is critical for the final reasoning. Given the passkeys of both Alice and Bob, the LLM is asked to concatenate them, e.g., if “Alice has ABC” and “Bob has DEF”, then the expected answer is “ABCDEF.” The order of concatenation is critical, and the final concatenated answer is controlled to match the passkeys used in Task 1. Notably, the reasoning is designed to be simple purposefully, and a model is expected to achieve good performance on this task if it performs well on Task 1 & 2.

For all tasks, the LLMs are prompted to generate the correct passkey from the long context. We generate 100 random passkeys and use the Wikitext-103 dataset (Grave et al., 2017) to augment the input context. The length of the Wikitext-103 context is varied between 0 and 128K tokens, with the passkeys randomly inserted at 30% and 60% positions within the context. Exact match (EM) is used to evaluate the LLMs’ performance. We test three modern LLMs in this experiment: COMMAND-R-PLUS<sup>2</sup>, GPT-4 (Achiam et al., 2023), and CLAUDE-3-HAIKU (Anthropic).

**Results** From Figure 1, we make several key observations. First, most LLMs perform well on the first two passkey retrieval tasks (Tasks 1 & 2), achieving high EM scores across context lengths ranging from 0K to 128K. Second, while the simple reasoning required by Task 3 is not an issue for most LLMs within short context, their performance drops significantly as the context length increases. This decline is more pronounced in Task 3 compared to the retrieval-only tasks in Task 1 and 2. Results of Task 3 reveal the challenge facing by modern LLMs in directly reasoning over the long-context input<sup>3</sup>. It further drives our motivation in leveraging an intermediate retrieval step as a stepping stone for the subsequent reasoning step to address the challenge.

### 3 METHODS

#### 3.1 OVERVIEW

To address the aforementioned challenges, we introduce a retrieve-then-reason approach which augments the long-context reasoning capability of LLMs via a two-stage breakdown, i.e., (i) first explicitly retrieving relevant facts pertaining to the task; and (ii) then reasoning over the collected facts to derive the final answer. Moreover, as manifested in Figure 2, modern LLMs struggled to retrieve with high accuracy in realistic scenarios and frequently hallucinate “*retrieved facts*”, leading to faulty reasoning and the generation of incorrect answers. To this end, we introduce a simple and effective method, namely ALR<sup>2</sup>, to augment the long-context reasoning capability of LLMs by an explicit two-stage procedure, i.e., ALigning LLMs with the objectives of both Retrieval and Reasoning. Our approach is inspired by the formulation of Retrieval-Augmented Generation (RAG) approach (Lewis et al., 2020), which was originally designed for open-domain QA.

#### 3.2 FORMULATION OF RAG

The standard open-domain QA problem can be formalized as:

$$\mathbf{y}^* = \arg \max_{\mathbf{y}} p_{\theta}(\mathbf{y}|\mathbf{q}),$$

where  $\mathbf{q}$  is the user question,  $\mathbf{y}^*$  is the answer with the highest probability, and  $\theta$  denotes parameters of the whole system. RAG introduces a latent variable  $\mathbf{z}$  and reformulates the conditional probability  $p_{\theta}(\mathbf{y}|\mathbf{q})$  as follows:

$$p_{\theta}(\mathbf{y}|\mathbf{q}) = \sum_{\mathbf{z}} p_{\phi}(\mathbf{z}|\mathbf{q}) p_{\mu}(\mathbf{y}|\mathbf{z}, \mathbf{q}), \quad (1)$$

<sup>2</sup><https://cohere.com/blog/command-r-plus-microsoft-azure>

<sup>3</sup>Similar trends are observed in more complex, realistic question-answering tasks, such as HotpotQA and SQuAD (see §4 for more details).

**Question:**

Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

**Golden Supporting Facts:**

[1] The Apple Remote is a remote control device released in or after October 2005 by Apple Inc. for use with a number of its products which use infrared capabilities.

[2] The device was originally designed to interact with the Front Row media program on the iSight iMac G5 and is compatible with some later desktop and portable Macintosh computers.

[3] *The software relies on iTunes and iPhone and is controlled by an Apple Remote or the keyboard function keys.*

**Golden Answer:** keyboard function keys

**Retrieved Supporting Facts:**

[1] The Apple Remote was originally designed to interact with the Front Row media program.

[2] Additionally, the *Siri Remote*, released in tandem with the fourth generation Apple TV, **is also compatible with the program.**

**Model Answer:** Siri Remote

Figure 2: Error case of the retrieve-then-reason prompting approach. The Command-R model and prompt in Figure 3 are used, and more details are in §4.1. The text with under-wave is the information matched between the golden facts and retrieved facts. We also use the text with underline to represent information that is not in golden facts but in the long context. The information hallucinated by LLM are marked by **red text**. The *italic text* in supporting facts shows the appearance of the answer.

where  $z$  represents the retrieved passage, while  $p_\phi(z|q)$  and  $p_\mu(y|z, q)$  are the probability distributions of the retriever and generator, respectively. The retriever, parameterized by  $\phi$ , builds an external retrieval index and retrieves the top- $K$  most relevant passages for each question from the index (Karpukhin et al., 2020; Izacard et al., 2022; Ni et al., 2022). The generator, typically a language model (Vaswani, 2017) parameterized by  $\mu$ , then takes the retrieved passages and the question  $q$  as input to generate the final answer.

### 3.3 ADAPTING RAG FORMULATION TO LONG-CONTEXT REASONING

Given that LLMs perform well in explicit long-context retrieval, a natural solution is to adapt the RAG formulation (Eq. 2) to re-organize the sparsely distributed relevant facts in intermediate steps, and then reason over those retrieved facts. We define our approach for long-context reasoning as follows:

$$p_\theta(y|q, c) = \sum_z p_\theta(z|q, c) p_\theta(y|z, q, c), \quad (2)$$

where  $c$  represents the long context, and both the retriever and the generator are parameterized by  $\theta$ . Since the key information  $z$  is essential for answering the question, the term  $p_\theta(z|q, c)$  models the process of explicitly retrieving  $z$  from the long context  $c$ . Once the key information is retrieved, the generator  $p_\theta(y|z, q, c)$  can utilize it to answer the question. By leveraging an intermediate retrieval step, our approach simplifies the long-context reasoning to its short-context counterpart, i.e., the reasoning procedure operates on the condensed evidence collected from the retrieval step.<sup>4</sup>

### 3.4 ALIGNING GENERATION TO RETRIEVAL & REASONING

Though Differentiable Search Index (DSI) (Tay et al., 2022; Bevilacqua et al., 2022) demonstrates that replacing the retrieval process with generation is feasible, pre-trained LLMs are generally not

<sup>4</sup>In Figure 3, we provide a concrete prompt to implement our approach. As shown in Appendix A, it can already significantly alleviate the performance drop on the reasoning task introduced in preliminary study (§2).

well-aligned with the retrieval objective. As illustrated in Figure 2, simply prompting the LLM to retrieve key information often results in low retrieval accuracy and severe hallucination<sup>5</sup>, which in turn leads to flawed reasoning and the production of incorrect answers.

To address these challenges, we align the LLM with both retrieval and reasoning objectives as defined in Eq. 2. To build the training data, we collect datasets that provide the necessary supporting facts (a list of statements crucial for answering the question) and use these facts as the target  $z^*$ . We then jointly optimize the LLM for both long-context retrieval and answering with the following objective:

$$\mathcal{L}_{align} = \frac{1}{N} \sum_{i=1}^N (\log p_{\theta}(z_i^* | q_i, c_i) + \log p_{\theta}(y^* | z_i^*, q_i, c_i)), \quad (3)$$

where  $N$  is the number of training examples, and  $z_i^*$  and  $y_i^*$  are the golden supporting facts and answers for the  $i$ -th example, respectively. Notably, for questions with multiple supporting facts, such as in multi-hop QA datasets (Yang et al., 2018), the LLM is trained to retrieve all supporting facts by formatting  $z^*$  as a concatenated string of bulleted facts. We adopt the prompting template in Figure 3 for our model training.

### 3.5 DISCUSSION

Our retrieval approach differs from others in two key aspects. First, most dense retrieval methods (Karpukhin et al., 2020), which use collections of dense vectors as the index, or Differentiable Search Index (DSI), which leverages model parameters (Tay et al., 2022) for indexing. In contrast, we treat the long input context itself as the retrieval index which supports the update of index on-the-fly. Second, most prior methods do not account for the relationships between retrieved statements and the correct number of statements to retrieve, often relying on large top- $K$  values to ensure all necessary information are retrieved. Differently, our retriever is explicitly trained to retrieve a coherent set of facts that are relevant for answering the question, which is particularly valuable in tasks like multi-hop QA. A better retrieval quality is not only important for the subsequent reasoning, but also provides a more trustworthy rationale, helping users better understand the decision-making process of LLMs.

## 4 EXPERIMENTS

### 4.1 SETUP

We follow the RULER benchmark (Hsieh et al., 2024) to evaluate the long-context performance of our approach and the baselines. Our focus is on question-answering (QA) tasks, but we also assess performance on retrieval tasks, such as variants of the Needle-in-a-Haystack (NIAH) task:

- **QA:** RULER utilizes two QA datasets, i.e., SQuAD (Rajpurkar et al., 2016) and HotpotQA (Yang et al., 2018), to build long-context evaluation datasets, which test the single-hop and multi-hop QA capabilities of LLMs in the long-context scenario.
- **NIAH:** The NIAH task is a synthetic retrieval task commonly used for long-context evaluation. The simplest version involves extracting a “needle”, such as an 8-digit number, from a long context. RULER includes four variants of the NIAH task: Single NIAH, Multi-keys NIAH, Multi-values NIAH, and Multi-queries NIAH.

More details about those datasets are shown in Appendix B.1.

We prepared the training data for each task accordingly. For the QA tasks, we used the supporting facts provided by HotpotQA and the sentences containing the answers in SQuAD as the golden retrieval results. Notably, each question in the SQuAD dataset only has one retrieved sentence, while HotpotQA questions typically have 2 to 6 supporting facts. These supporting facts are organized into a string of bulleted items. For the NIAH tasks, we treated the needle sentences as the retrieved results. The number of training examples from HotpotQA, SQuAD, and the four NIAH variants is 5,000, 25,000, and 1,600, respectively. The context lengths (in tokens) of the training examples are 4K, 8K, 16K, and 32K, with the data ratios for each being 1/10, 2/10, 3/10, and 4/10, respectively. Inference for longer contexts is done by extrapolation. More training details are in Appendix B.2. We used

<sup>5</sup>See §5.1 for more quantitative analysis for the retrieval.

Model	Prompt	4K	8K	16K	32K	64K	128K	Overall
<i>HotpotQA</i>								
GPT-3.5	DA	52.4	49.4	46.5	—	—	—	—
GPT-4	DA	59.8	58.1	57.8	53.8	51.0	48.4	54.8
GEMINI	DA	59.4	57.3	58.72	54.5	56.7	52.6	56.5
CLAUDE-3-HAIKU	DA	49.2	48.0	42.4	43.0	38.8	49.4	45.1
CMD-R	DA	51.6	51.2	47.4	47.4	43.6	34.29	45.9
	RR	58.8	55.6	54.8	54.6	51.4	44.8	53.3
	QF	53.4	54.4	49.4	48.6	44.8	32.8	46.0
	S2A	43.0	44.6	42.8	40.8	34.0	30.5	39.2
CMD-R-FT	DA	72.8	71.39	72.0	69.1	67.4	57.4	68.3
ALR <sup>2</sup>	RR	<b>77.2</b>	<b>76.8</b>	<b>77.2</b>	<b>76.6</b>	<b>77.5</b>	<b>75.2</b>	<b>76.7</b>
<i>SQuAD</i>								
GPT-3.5	DA	63.8	60.0	53.1	—	—	—	—
GPT-4	DA	82.3	75.6	72.3	70.1	63.0	59.4	70.4
GEMINI	DA	65.9	67.2	66.5	65.6	64.6	62.5	65.4
CLAUDE-3-HAIKU	DA	67.0	63.6	60.6	53.2	53.8	55.0	58.8
CMD-R	DA	65.6	61.4	63.8	62.4	54.0	49.6	59.4
	RR	58.8	55.1	52.4	51.8	51.3	49.2	53.1
	QF	63.6	61.6	60.6	57.9	46.6	43.2	55.5
	S2A	42.0	42.4	49.4	46.6	41.1	38.3	43.3
CMD-R-FT	DA	80.8	54.2	<b>68.6</b>	78.2	50.6	53.2	64.2
ALR <sup>2</sup>	RR	<b>89.8</b>	<b>63.0</b>	64.0	<b>78.2</b>	<b>66.6</b>	<b>71.2</b>	<b>72.1</b>
<i>NIAH</i>								
GPT-3.5	DA	99.85	98.55	93.05	—	—	—	—
GPT-4	DA	100.0	99.9	99.65	98.5	96.1	91.4	97.60
CMD-R	DA	94.6	91.9	89.29	94.37	93.84	87.99	92.00
CMD-R-FT	DA	<b>100.0</b>	<b>100.0</b>	99.95	<b>99.95</b>	<b>99.85</b>	98.79	99.75
ALR <sup>2</sup>	RR	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>99.95</b>	<b>99.85</b>	<b>98.99</b>	<b>99.79</b>

Table 1: Experiments on question-answering (QA) and needle-in-a-haystack (NIAH) tasks. The best results are highlighted by bold text. The evaluation metric is Extract Match (EM).

Exact Match (EM) as our evaluation metric. Following the setup of RULER, we report the average EM score on 500 test cases for each dataset or sub-task. Importantly, to avoid data contamination, we ensured that both the context and the question-answer pairs for testing were unseen during training.

## 4.2 BASELINES

We employ several prompting techniques for the long-context tasks:

- **DA**: This prompting approach is used for Direct Answering, where the model directly generates an answer without explicit retrieval. The template is illustrated in Figure 4.
- **RR**: This prompting method, shown in Figure 3, is used for the Retrieve-then-Reason framework in our work. It first prompts the model to retrieve the key information from the long context and then complete the reasoning to answer the question.
- **QF**: Similar to the RR, Anthropic<sup>6</sup> and Lee et al. (2024) empirically find that guiding the LLM to generate Quotes and Citations First, before providing answers, improves long-context reasoning. We evaluate a variant of this approach, as shown in Figure 5.
- **S2A**: Instead of retrieving information directly from the long context, the System-2 Attention approach (Weston & Sukhbaatar, 2023) focuses on answering questions based on summarized and reorganized information extracted from the long context. Our prompt template for this approach is presented in Figure 6.

For our experiments, we use the Command-R (CMD-R) model<sup>7</sup> with 35 billion parameters as our backbone. We evaluate its performance using the four prompting techniques: DA, RR, QF, and S2A. These prompting-based baselines are referred to as CMD-R + X, where X is the name of

<sup>6</sup><https://docs.anthropic.com/claude/docs/advanced-text-analysis>

<sup>7</sup><https://cohere.com/blog/command-r>

Model	Prompt	4K	8K	16K	32K	64K	128K	Overall
<i>Hallucination Rate ↓</i>								
CMD-R	RR	66.57	61.04	54.54	57.01	60.79	66.74	61.1
ALR <sup>2</sup>	RR	<b>0.26</b>	<b>0.17</b>	<b>0.17</b>	<b>0.17</b>	<b>0.36</b>	<b>0.61</b>	<b>0.29</b>
<i>Recall ↑</i>								
CMD-R	RR	27.37	32.50	39.53	41.35	41.34	22.28	34.06
ALR <sup>2</sup>	RR	<b>69.47</b>	<b>69.39</b>	<b>69.64</b>	<b>69.47</b>	<b>67.78</b>	<b>66.99</b>	<b>68.79</b>

Table 2: Evaluation of retrieval quality on HotpotQA dataset. The hallucination rate is the percentage of retrieved sentences that are not presented in the context. The recall score is the percentage of golden facts that are displayed in retrieved sentences. The best results are highlighted by bold text.

the prompt used. Additionally, to ensure a fair assessment on the performance gains obtained from model fine-tuning, we include another CMD-R-FT + DA baseline. This model is built by fine-tuning CMD-R on the same training data, described in §4.1, to directly produce the answer, i.e., without a retrieval step, given the input context and the question. For a broader set of evaluations, we also test the DA prompt on other frontier models, including GPT-3.5<sup>8</sup> (Ouyang et al., 2022), GPT-4 (Achiam et al., 2023), GEMINI (Team et al., 2023), and CLAUDE-3-HAIKU (Anthropic).

### 4.3 RESULTS

As shown in Table 1, ALR<sup>2</sup> achieves significantly better performance on the two question-answering (QA) benchmarks. Notably, on the multi-hop QA benchmark, HotpotQA, ALR<sup>2</sup> maintains consistent performance from 4K to 128K tokens of context. Additionally, the retrieve-then-reason prompting baseline, i.e., CMD-R+RR, shows a smoother performance curve on HotpotQA compared to the direct-answer baseline, i.e., CMD-R+DA. However, on the single-hop SQuAD benchmark, CMD-R+RR performs worse than CMD-R+DA. We hypothesize that this is due to over-retrieval by CMD-R+RR, as it tends to retrieve excessive information that becomes redundant for single-hop questions. In contrast, after aligning the LLM to handle both single-hop and multi-hop questions, our ALR<sup>2</sup> approach achieves superior overall performance.

We also compare different two-stage prompting techniques, namely RR, QF, and S2A. As shown in Table 1, CMD-R+RR outperforms CMD-R+QF on HotpotQA but falls short on SQuAD. We find that this discrepancy is closely related to the number of sentences retrieved by each prompt. Specifically, CMD-R+RR retrieves an average of 7.5 sentences across both QA datasets, while CMD-R+QF retrieves only 2.2 sentences. Since HotpotQA is a multi-hop QA dataset that requires more supporting facts, CMD-R+RR performs better on this dataset. Both CMD-R+QF and CMD-R+RR outperform the system-2 attention baseline, i.e., CMD-R+S2A, in our experiments.

In addition, on the NIAH tasks, all evaluated approaches obtain comparable performances. This is unsurprising given the fact that NIAH tasks only require straightforward information retrieval, which is relatively easy for LLMs to handle. When evaluating long-context performance, we caution researchers against averaging the performance of long-context retrieval and reasoning tasks, such as NIAH and QA, due to the significant performance gap between them.

## 5 ANALYSES

### 5.1 RETRIEVAL QUALITY BY LLM

**Setup.** We evaluate the hallucination rate of our retriever,  $p_\theta(z|q, c)$ , on the HotpotQA dataset. The dataset construction for the long-context setting follows the same procedure as described in the main experiments (§4). The hallucination rate is defined as the percentage of retrieved sentences that

<sup>8</sup>Since the maximum context length of GPT-3.5 is 16K, we only show the performance of GPT-3.5 within this context length.

Model	Prompt	4K	8K	16K	32K	64K	128K	Overall
<i>StrategyQA</i>								
CMD-R-FT	DA	61.57	60.26	60.69	60.69	58.95	56.33	59.74
ALR <sup>2</sup>	RR	<b>72.48</b>	<b>71.61</b>	<b>70.74</b>	<b>70.16</b>	<b>71.61</b>	<b>72.48</b>	<b>71.51</b>
<i>TriviaQA</i>								
CMD-R-FT	DA	74.6	74.2	73.2	70.0	69.0	69.19	71.69
ALR <sup>2</sup>	RR	<b>77.0</b>	<b>76.4</b>	<b>74.4</b>	<b>72.60</b>	<b>73.94</b>	<b>71.98</b>	<b>74.38</b>

Table 3: EM scores on StrategQA and TriviaQA tasks. The best results are highlighted by bold text.

do not match any sentences in the context, calculated as:

$$\text{Hallucination}(\mathcal{Z}, \mathcal{C}) = 1 - \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\{z_i \in \mathcal{C}_i\}},$$

where  $\mathcal{Z} = \{z_i\}_{i=1}^N$  is the collection of all retrieved sentences,  $z_i$  is a single retrieved sentence, and  $\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^N$  is the corresponding collection of input context. Additionally, since HotpotQA provides golden supporting facts for each question, we evaluate retrieval quality using the recall score, which measures the fraction of golden facts successfully retrieved by the LLM.

**Results.** As shown in Table 2, by aligning generation with retrieval objective, the hallucination rate of our approach is significantly better than the baseline. Additionally, ALR<sup>2</sup> also achieves notably higher retrieval recall scores compared to the prompting-based method, highlighting the importance of generation-retrieval alignment. The reduced hallucination rate and improved retrieval recall are not only crucial for enhancing the generator’s accuracy, but they also provide a more trustworthy rationale, helping users better understand the decision-making process of LLMs.

## 5.2 GENERALIZATION TO UNSEEN DATA

**Setup.** We are also interested in whether ALR<sup>2</sup>, trained on the SQuAD (Rajpurkar et al., 2016) and HotpotQA (Yang et al., 2018) datasets, can generalize to unseen datasets. To explore this, we follow the strategy of RULER (Hsieh et al., 2024) to pre-process two additional QA datasets: StrategyQA (Geva et al., 2021) and TriviaQA (Joshi et al., 2017). We compare the performance of the LLM fine-tuned on the direct-answer template (CMD-R-FT + DA) with our approach (ALR<sup>2</sup>).

**Results.** As shown in Table 3, ALR<sup>2</sup> consistently outperforms CMD-R-FT + DA across context lengths ranging from 4K to 128K. These results indicate that ALR<sup>2</sup> provides a more robust and stable framework for solving long-context QA problems, even when applied to unseen datasets.

## 6 RELATED WORK

### 6.1 LONG-CONTEXT LARGE LANGUAGE MODEL

Long-context ability is critical for employing a large language model (LLM) to solve real-world problems, e.g., document analysis (Wang et al., 2024a), tool use (Yao et al., 2023; Schick et al., 2023), etc. Many components of an LLM may affect the long-context performance. For instance, researchers find that the design of the position embedding is important for the extrapolation performance of an LLM (Su et al., 2024a; Press et al., 2022; Chen et al., 2023; Ruoss et al., 2023; Men et al., 2024), i.e., testing the LLM on context length that is larger than the training window size. In addition, the attention distribution and scale of the Transformer model (Chiang & Cholak, 2022; Xiao et al., 2024) may also affect the long-context ability.

To extend the context length of LLM, some works try to compress the long context into shorter text (Jiang et al., 2023; 2024; Xu et al., 2024), limited hidden representations (Chevalier et al., 2023; Ge et al., 2024), or model parameters (Wang et al., 2024b), saving the number of input tokens in the context. Moreover, splitting long context into fixed segments and caching them for re-use is



also widely used for increasing the effective context length of LLMs (Dai et al., 2019). Different from Transformer-XL (Dai et al., 2019) that only leverages local attention, Munkhdalai et al. (2024) introduce an additional global momentum to store the knowledge in all the previous context. Yen et al. (2024) leverages a small bi-directional model (Liu, 2019) to encode the segments independently, and allows the LLM to attend to those segments during generation.

In our work, the target is not to extend the context length. In contrast, we find that the performance of QA will degenerate more significantly with longer context for most popular LLMs, similar to the findings in Liu et al. (2024). Therefore, we propose a new formulation inspired by retrieval-augmented generation to alleviate this common issue of LLMs.

Recently, there are also many benchmarks are proposed to evaluate the long-context performance of LLMs (Hsieh et al., 2024; Wang et al., 2024a; Lee et al., 2024; Zhang et al., 2024; Krishna et al., 2023; Karpinska et al., 2024; Kuratov et al., 2024). Considering our focus is the general QA problems and the construction of long-context QA data is similar in many benchmarks, we use RULER (Hsieh et al., 2024) as our main benchmark.

## 6.2 RETRIEVAL-AUGMENTED GENERATION

Retrieval-Augmented generation (RAG) has been extensively used to solve NLP problems, e.g., open-domain question-answering (Lewis et al., 2020; Asai et al., 2024), language modeling (Guu et al., 2020; Lan et al., 2023; Shi et al., 2024), machine translation (Gu et al., 2018; Cai et al., 2021), etc. For most RAG works, the dual-encoder-based retriever (Karpukhin et al., 2020; Izacard et al., 2022; Ni et al., 2022), which measures the similarity between inputs and retrieval documents by their encoded embeddings, has become the *de facto* approach for retrieval. The retrieved top- $K$  documents will be integrated into the generator, e.g., a language model (Radford et al., 2019), to enhance the generation. The most related work in this research line is Xu et al. (2024), which aims to save the number of input tokens to an LLM by compressing the retrieved  $K$  documents. However, our work focuses on enhancing the long-context reasoning of LLMs.

## 6.3 CITATION GENERATION

There exist many works that aim to attribute the prediction of an LLM to the information source that the LLM is based on, i.e., citation generation. Some train the model to learn this ability on a standardized citation format (Menick et al., 2022; Taylor et al., 2022; Khalifa et al., 2024; Ye et al., 2024). While others use the instruction-following ability and post-processing techniques to enable this feature at inference (Gao et al., 2023a; Kamaloo et al., 2023; Gao et al., 2023b; Liu et al., 2023).

Our work aims to precisely retrieve the bag of facts that can support the question-answering in long-context scenario, which is also beneficial for the explainability of the prediction. However, the focus of our work is more about enhancing the long-context QA ability of LLMs, which is orthogonal to works in citation generation.

## 7 CONCLUSION

In this work, we addressed the challenge of long-context reasoning in large language models (LLMs) by introducing ALR<sup>2</sup>, a retrieve-then-reason approach that aligns LLMs with both retrieval and reasoning objectives. Through extensive experiments on both single-hop and multi-hop QA tasks, we demonstrated that ALR<sup>2</sup> significantly outperforms existing baselines, especially in long-context scenarios. Our analysis also highlighted the importance of aligning generation with retrieval to reduce hallucinations and improve retrieval accuracy. Moreover, we showed that ALR<sup>2</sup> generalizes well to unseen datasets, offering a robust solution for long-context QA problems.

**Limitations.** We also note some of the limitations of our work. For example, as most works based on the RAG formulation, our method is hard to enhance the summarization tasks, in which all the information in the long context is important for the final prediction. In addition, only considering one granularity, i.e., sentence, in the retrieval may be limited in some scenarios. A better way is to allow the users to choose the retrieval granularity, e.g., phrase, sentence, or passage, in the instruction. We leave the addressing of those limitations in future works.

## 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.
- Anthropic. The claude 3 model family: Opus, sonnet, haiku. URL <https://api.semanticscholar.org/CorpusID:268232499>.
- 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, 2024. URL <https://openreview.net/forum?id=hSyW5go0v8>.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020.
- Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. Autoregressive search engines: Generating substrings as document identifiers. Advances in Neural Information Processing Systems, 35:31668–31683, 2022.
- Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. Neural machine translation with monolingual translation memory. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 7307–7318, 2021.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. arXiv preprint arXiv:2306.15595, 2023.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 3829–3846, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.232. URL <https://aclanthology.org/2023.emnlp-main.232>.
- David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 7654–7664, 2022.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL <https://aclanthology.org/P19-1285>.
- Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. Rarr: Researching and revising what language models say, using language models. In The 61st Annual Meeting Of The Association For Computational Linguistics, 2023a.
- 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, 2023b.
- Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In The Twelfth International Conference on Learning Representations, 2024. URL <https://openreview.net/forum?id=uREj4ZuGJE>.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. Transactions of the Association for Computational Linguistics, 9:346–361, 2021.

- Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=B184E5qee>.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- 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.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Krizan, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.
- 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*, 2022.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.825. URL <https://aclanthology.org/2023.emnlp-main.825>.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.acl-long.91>.
- 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.
- Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*, 2023.
- Gregory Kamradt. Needle In A Haystack - pressure testing LLMs. *Github*, 2023. URL <https://github.com/gkamradt/LLMTest.NeedleInAHaystack/tree/main>.
- Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. One thousand and one pairs: A “novel” challenge for long-context language models, 2024. URL <https://arxiv.org/abs/2406.16264>.
- 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.
- Muhammad Khalifa, David Wadden, Emma Strubell, Honglak Lee, Lu Wang, Iz Beltagy, and Hao Peng. Source-aware training enables knowledge attribution in language models. *arXiv preprint arXiv:2404.01019*, 2024.
- Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. Longeval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1650–1669, 2023.

- 594 Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and  
595 Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack.  
596 arXiv preprint arXiv:2406.10149, 2024.
- 597
- 598 Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. Copy is all you need.  
599 In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=CR010A9Nd8C>.
- 600
- 601 Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko,  
602 Yi Luan, Sébastien MR Arnold, Vincent Perot, Siddharth Dalmia, et al. Can long-context language  
603 models subsume retrieval, rag, sql, and more? *arXiv preprint arXiv:2406.13121*, 2024.
- 604
- 605 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,  
606 Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented genera-  
607 tion for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:  
608 9459–9474, 2020.
- 609 Nelson F Liu, Tianyi Zhang, and Percy Liang. Evaluating verifiability in generative search engines. In  
610 *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7001–7025, 2023.
- 611
- 612 Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and  
613 Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of*  
614 *the Association for Computational Linguistics*, 12:157–173, 02 2024. ISSN 2307-387X. doi:  
615 10.1162/tacl.a.00638. URL <https://doi.org/10.1162/tacl.a.00638>.
- 616 Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint*  
617 *arXiv:1907.11692*, 2019.
- 618
- 619 Xin Men, Mingyu Xu, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng  
620 Chen. Base of rope bounds context length. *arXiv preprint arXiv:2405.14591*, 2024.
- 621
- 622 Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick,  
623 Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching  
624 language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.
- 625 Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. Leave no context behind: Efficient  
626 infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 2024.
- 627
- 628 Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao,  
629 Yi Luan, Keith Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. In  
630 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp.  
631 9844–9855, 2022.
- 632 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong  
633 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow  
634 instructions with human feedback. *Advances in neural information processing systems*, 35:27730–  
635 27744, 2022.
- 636 Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S  
637 Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th*  
638 *annual acm symposium on user interface software and technology*, pp. 1–22, 2023.
- 639
- 640 Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables  
641 input length extrapolation. In *International Conference on Learning Representations*, 2022. URL  
642 <https://openreview.net/forum?id=R8sQPpGCv0>.
- 643 Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language  
644 models are unsupervised multitask learners. 2019.
- 645
- 646 Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for  
647 machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in*  
*Natural Language Processing*, pp. 2383–2392, 2016.

- Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. Randomized positional encodings boost length generalization of transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1889–1903, 2023.
- 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. URL <https://openreview.net/forum?id=Yacmpz84TH>.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 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)*, pp. 8364–8377, 2024.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), mar 2024a. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL <https://doi.org/10.1016/j.neucom.2023.127063>.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024b.
- Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843, 2022.
- Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.
- 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.
- A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, et al. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. *arXiv preprint arXiv:2406.17419*, 2024a.
- Y Wang, D Ma, and D Cai. With greater text comes greater necessity: Inference-time training helps long text generation. *arXiv preprint arXiv:2401.11504*, 2024b.
- Jason Weston and Sainbayar Sukhbaatar. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*, 2023.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=NG7sS51zVF>.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=m1JLVigNHp>.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.
- 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*, 2023. URL [https://openreview.net/forum?id=WE\\_vluYUL-X](https://openreview.net/forum?id=WE_vluYUL-X).

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. Effective large language model adaptation for improved grounding and citation generation. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 6237–6251, 2024.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2588–2610, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.acl-long.142>.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al.  $\infty$ Bench: Extending long context evaluation beyond 100k tokens. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 15262–15277, 2024.

**Stage-1 Input:**

Please retrieve all the sentences in the given documents that are important and relevant to answer the question.

Question: {QUERY}

The following are given documents.

{CONTEXT}

Please retrieve all the sentences in the given documents that are important and relevant to answer the question. The question is highlighted again at below.

Question: {QUERY}

Retrieved sentences:

(For each retrieved sentence, please start from the bullet symbol “-”)

**Stage-1 Output:**

{MODEL.RETRIEVED.SENTENCES}

**Stage-2 Input:**

Please answer the question based on the given retrieved information. Only give me the answer and do not output any other words.

Question: {QUERY}

Answer:

**Stage-2 Output:**

{ANSWER}

Figure 3: The retrieve-then-reason (RR) prompt for long-context QA. The {CONTEXT} is the placeholder for long context and {QUERY} is for user question. The red parts {MODEL.RETRIEVED.SENTENCES} and {ANSWER} are generated by LLM.

**Stage-1 Input:**

Answer the question based on the given documents. Only give me the answer and do not output any other words.

The following are given documents.

{CONTEXT}

Answer the question based on the given documents. Only give me the answer and do not output any other words.

Question: {QUERY}

**Stage-1 Output:**

{ANSWER}

Figure 4: The direct-answering (DA) prompt for long-context QA. The {CONTEXT} is the placeholder for long context, {QUERY} is for user question, and the red part {ANSWER} is the answer directly generated by LLM.

## A ADDITIONAL RESULTS FOR PRELIMINARY STUDY

We also evaluate the effectiveness of the retrieve-then-reason framework on the challenging task in the section of preliminary study (§2). As shown in Table 4, simply using the RR prompt (Figure 3)

Model	Prompt	16K	32K	64K	128K	Overall
CMD-R-PLUS	DA	0.70	0.16	0.0	0.0	0.21
CMD-R-PLUS	RR	<b>0.84</b>	<b>0.86</b>	<b>0.56</b>	<b>0.0</b>	<b>0.56</b>

Table 4: EM scores on Task 3, i.e., Multi-hop Passkey Retrieval + Reasoning, in our preliminary study (§2). The best results are highlighted by bold text.

for the Command-R-Plus model can improve the overall EM score from 0.21 to 0.56, significantly alleviating the performance drop on the task that requires long-context reasoning.

## B DETAILS OF EXPERIMENT SETUP

### B.1 RULER TASKS

**QA Tasks** We use the validation sets of SQuAD (Rajpurkar et al., 2016) and HotpotQA (Yang et al., 2018) datasets to build the long-context benchmarks, where each question in those datasets is associated with some relevant passages. Following Hsieh et al. (2024), We use concatenated and shuffled passages as the context, which include both relevant passages and randomly sampled ones.

**NIAH Tasks** Following Hsieh et al. (2024), we consider four kinds of Needle-in-a-Haystack (NIAH) tasks in our work and report the averaged performance of them:

- **Single NIAH (S-NIAH)**: The “haystack” text is from Paul Graham essays (Kamradt, 2023) is from “needle” sentence of this task is “One of the special magic numbers for {KEY} is: {VALUE}”, where the “{KEY}” and “{VALUE}” are placeholders for English words and 7-digit numbers, respectively. For S-NIAH, we only insert one needle sentence into the long-context.
- **Multi-keys NIAH (MK-NIAH)**: In this setting, we insert four needle sentences into the context and ask an LLM to retrieve only one of them. Each of the four needle sentences has unique “{KEY}” and “{VALUE}”.
- **Multi-values NIAH (MV-NIAH)**: In this setting, part of the needle sentences may share the same key. We will ask an LLM to retrieve all the values associated with the same key.
- **Multi-queries NIAH (MQ-NIAH)**: Compared with MK-NIAH, we need the LLM to retrieve multiple values in needle sentences associated with specific keys.

### B.2 TRAINING DETAILS

During training, we set the base value of rotary position embedding to  $8e6$  (Men et al., 2024; Su et al., 2024b). The training batch size is 64 on V4-512 TPU. For every 3 updates on the prepared training dataset, we will also update our parameters on pre-training data with 64K tokens for one step. For both of the two-stage and one-stage fine-tuning methods, we will train the command-r model with 35 billion parameters for 3 epochs.



**Stage-1 Input:**

I'm going to give you some documents. Then I'm going to ask you a question about it. I'd like you to first write down exact quotes of parts of the document that would help answer the question, and then I'd like you to answer the question using facts from the quoted content. Here is the document:

```
<document>
{CONTEXT}
</document>
```

First, find the quotes from the document that are most relevant to answering the question, and then print them in numbered order. Please start the quotes with "Relevant quotes:".

Then, answer the question, starting with "Answer:". Do not include or reference quoted content verbatim in the answer. Don't say "According to Quote [1]" when answering. After "Answer:", please only give me the answer and do not output any other words.

Thus, the format of your overall response should look like what's shown between the tags. Make sure to follow the formatting and spacing exactly.

```
<example>
```

Relevant quotes:

- [1] "Company X reported revenue of \$12 million in 2021."
- [2] "Almost 90% of revenue came from widget sales, with gadget sales making up the remaining 10%."

Answer: \$12 million [1].

```
</example>
```

Here is the first question: {QUERY}

Give me the quotes and answer for the question immediately without preamble.

**Stage-1 Output:**

```
{QUOTES_AND_ANSWER}
```

Figure 5: The quotes-and-citation-first (QF) prompt for long-context QA. The {CONTEXT} is the placeholder for long context and {QUERY} is for user question. The format of {QUOTES\_AND\_ANSWER} is similar to the example embraced by <example> and </example>.

**Stage-1 Input:**

Given the following text by a user, extract the part that is unbiased and not their opinion within a single paragraph, so that using that text alone would be good context for providing an unbiased answer to the question portion of the text.

----- Start of Text by User -----

Answer the question based on the given documents. Only give me the answer and do not output any other words.

The following are given documents.

{CONTEXT}

Answer the question based on the given documents. Only give me the answer and do not output any other words.

Question: {QUERY}

----- End of Text by User -----

Given the above text by a user, extract the part that is unbiased and not their opinion within a single paragraph, so that using that text alone would be good context for providing an unbiased answer to the question portion of the text.

Please include the actual question or query that the user is asking. Separate this into two categories labeled with "Unbiased text context (includes all content except user's bias):" and "Question/Query (does not include user bias/preference):".

**Stage-1 Output:**

{COMPRESSED\_TEXT}

**Stage-2 Input:**

% Previous Chat History Will Be Discarded %

{COMPRESSED\_TEXT}

**Stage-2 Output:**

{ANSWER}

Figure 6: The system-2 attention (S2A) prompt for long-context QA. The {CONTEXT} is the placeholder for long context and {QUERY} is for user question. The {COMPRESSED\_TEXT} contains the copied question and the compressed long context based on the question. In stage-2, we only input the {COMPRESSED\_TEXT} to LLM, discarding the previous chat history.