Unleashing Multi-Hop Reasoning Potential in Large Language Models through Repetition of Misordered Context

Anonymous ACL submission

Abstract

Multi-hop reasoning, which requires multi-step reasoning based on the supporting documents within a given context, remains challenging for large language models (LLMs). LLMs often struggle to filter out irrelevant documents within the context, and their performance is sensitive to the position of supporting documents within that context. In this paper, we identify an additional challenge: LLMs' performance is also sensitive to the order in which the supporting documents are presented. We refer to this as the **misordered context** problem. To address this issue, we propose a simple yet effective method called **co**ntext **re**petition (**CoRe**), which involves prompting the model by repeatedly presenting the context to ensure the supporting documents are presented in the optimal order for the model. Using CoRe, we improve the F1 score by up to 30%p on multi-hop QA tasks and increase accuracy by up to 70%p on a synthetic task. Additionally, CoRe helps mitigate the well-known "lost-in-the-middle" problem in LLMs and can be effectively combined with retrieval-based approaches utilizing Chain-of-Thought (CoT) reasoning.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023) are capable of performing various complex natural language processing tasks due to their emergent abilities in in-context learning, instruction following, and step-by-step reasoning (Wei et al., 2022; Kojima et al., 2022; Zhao et al., 2023). However, multi-hop reasoning still remains a challenging task for LLMs. In this task, multiple external source documents related to the given query are provided as context, and the model searches for useful supporting documents within this context to find the answer. The model derives an answer through multiple steps of reasoning based on these supporting documents (Mavi et al., 2024).

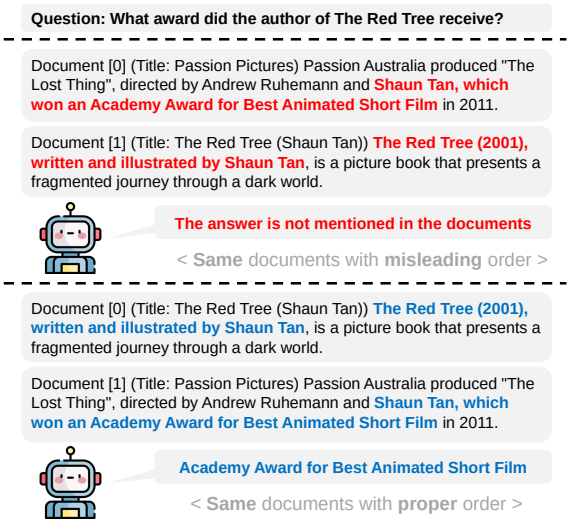

Figure 1: An example of the misordered context problem in multi-hop reasoning of large language models, sampled from the MuSiQue dataset. Model performance is sensitive to the order of given documents.

During this process, the model faces several challenges in terms of grounding to the context (Gao et al., 2023). First, the model struggles to filter out noisy documents that are irrelevant to the correct answer (Shi et al., 2023). In fact, noisy documents can significantly degrade the reasoning performance of LLMs (Cuconasu et al., 2024; Wu et al., 2024). Additionally, LLMs are highly sensitive to the position of supporting documents within the context. Specifically, when a document is positioned in the middle of the context, the model may fail to recognize it properly, resulting in a "lost-in-the-middle" problem (Liu et al., 2023). These issues become more critical as the length of the context grows and the number of retrieved documents increases (Lee et al., 2024; Li et al., 2024).

In this study, we address additional context grounding problems related to multi-hop reasoning that have not been deeply discussed so far. In a multi-hop reasoning task, there are multiple supporting documents, and the model derives the answer by considering each document through step-by-step reasoning. Due to the nature of the decoder-only structure, which performs causal attention, LLMs can only reference documents in a left-to-right order. If this order is presented unfavorably for the model's reasoning, the model's performance may be significantly degraded. Following this rationale, we observe that, even in ideal situations without noisy documents, LLMs exhibit up to a 26%p difference in F1 score depending on the order of supporting documents. Since there is no guarantee that supporting documents will be presented to the model in the optimal order during actual tasks, this problem inevitably arises. We call it the **misordered context** problem. Figure 1 shows an example of the misordered context problem.

To address the problem, we approach it from the perspective of context augmentation. Specifically, by repeating the given context multiple times, we construct an augmented context that allows the supporting documents to be presented to the model in the optimal order. We theoretically explain that by repeating the context containing k supporting documents k times, all possible orders of presenting the documents can be covered. That is, *the k-repeated context can always present the supporting documents to the model in the optimal order*. Based on this theoretical analysis, we propose the **co**ntext **re**petition (**CoRe**) method to tackle the misordered context problem.

We conduct experiments on 3 multi-hop QA benchmarks—HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022)—as well as an additional synthetic task. In the synthetic task, the model is required to search for elements in lists composed of non-negative integer elements, based on context where information is presented in a reverse order. We show that CoRe significantly improves multi-hop QA task performance across various LLMs. For example, in the 2WikiMultihopQA task using the Llama-3.1-8B-Instruct model, we observe up to 30%p improvement in F1 score, and in the synthetic task, we observe up to a 70%p improvement in accuracy. Additionally, empirical analysis demonstrates that CoRe provides performance benefits to the model by improving order-related reasoning. Furthermore, CoRe alleviates the previously known "lost-in-the-middle" problem and can be effectively combined with Chain-of-Thought (CoT) methods in retrieval-augmented tasks.

In summary, our work (1) introduces the misordered context problem, a critical challenge for LLMs in multi-hop reasoning, (2) theoretically proposes an approach of context augmentation to tackle this issue, which leads to our simple but practical CoRe method, and (3) demonstrates that this method can significantly improve model performance across various tasks.

2 Related Work

The process of modeling the multi-step reasoning required in a multi-hop QA task typically involves methods that decompose the query step-by-step to predict the answer (Min et al., 2019; Yadav et al., 2019; Malon and Bai, 2020). From the perspective of utilizing LLMs, methods have been proposed that combine with CoT techniques, allowing the model to self-generate subsequent questions and predict intermediate answers while performing multi-step reasoning (Yao et al., 2023; Khot et al., 2022; Lazaridou et al., 2022; Press et al., 2023; Jiang et al., 2023b; Xu et al., 2024). However, these methods often involve complex processes where errors propagate and accumulate. To reduce this complexity, we aim to enhance LLMs' innate multi-hop reasoning capabilities.

In retrieval-augmented generation (Lewis et al., 2020; Izacard et al., 2023; Shi et al., 2024a), the typical approach for multi-hop QA is to retrieve documents related to the query, and have the LLM reason based on this context in a "retrieve-and-reason" fashion. A key issue here is handling the noise introduced by irrelevant documents within the context. To address this, methods have been proposed to help the model correctly locate the evidence within the context (Shi et al., 2024b; Kim et al., 2024; Yu et al., 2023; Yan et al., 2024). With the recent increase in the LLMs' context length, Ziyan Jiang (2024) have also explored including more documents and leveraging the LLM's reasoning capabilities. We aim to propose a method focusing on document order to enable the LLM to reason correctly within the context.

From the perspective of repetition in LLMs, Springer et al. (2024) demonstrate that by repeating sentences during the sentence embedding extraction process, it is possible to compute embeddings

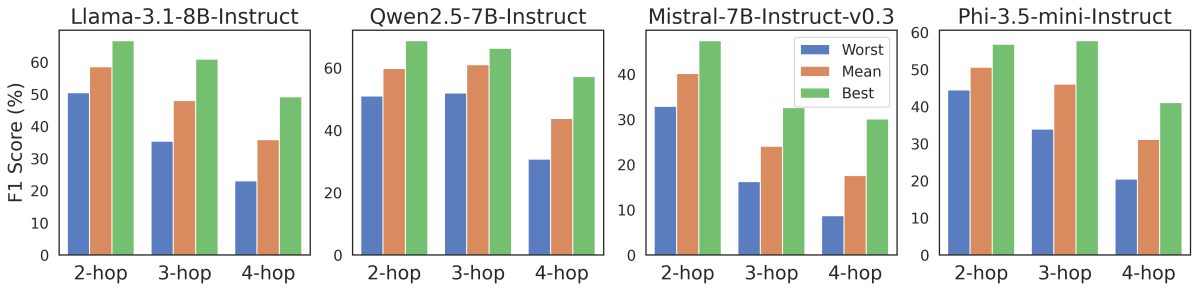

Figure 2: Evaluation results of F1 score for each query type in MuSiQue with permuted clean contexts.

that encode bidirectional context even in unidirectional LMs. Xu et al. (2023) show that in arithmetic reasoning tasks, repeating the query can improve performance by leveraging bidirectional understanding. While these methods are similar to ours in that they employ repetition, we differ fundamentally in that we focus on the issue of document order in multi-hop reasoning rather than on simple bidirectional understanding.

3 CoRe: Context Augmentation for misordered Supporting Documents

In this section, we empirically demonstrate that the performance of LLMs in multi-hop reasoning is sensitive to the order of supporting documents within the context, a phenomenon we refer to as the misordered context problem. For this issue, we theoretically formalize the problem from the perspective of context structure. Building on this formalization, we propose our method, context repetition, which ensures that supporting documents are presented to the model in an optimal order for reasoning.

3.1 Problem of misordered Context in Multi-Hop Reasoning

We begin by addressing the following research question: *Is the performance of LLMs on multi-hop reasoning tasks sensitive to the order of supporting documents within the context?* To explore this question, we conduct an empirical study in the context of the multi-hop QA task using the MuSiQue dataset. The dataset consists of samples categorized into 3 types: 2-hop, 3-hop, and 4-hop. For a given k-hop sample, there are k supporting documents and several noisy documents that distract from the correct answer. To isolate the effect of supporting document order, we remove the noisy documents and construct clean contexts solely from the supporting documents for our experiments.

For each k-hop query, we permute the order of the k supporting documents in every possible way, resulting in a total of $k!$ different contexts. We then evaluate the model's response to each of these contexts. In other words, for each k-hop query, the model generates a total of $k!$ answers, and we assess these answers using the F1 score. We categorize the evaluation results based on the types of order of the documents within contexts associated with each query. The type is determined by the conditional probability of the correct answer a for the model θ, $p_\theta(a \mid Q, C, I)$ where Q is the query, C is the context in the specific order of the supporting documents, and I is the instruction for question answering. The higher the probability, the more optimally the supporting documents are ordered within the context for the answering of the model. We evaluate the model's performance across the best-case (highest probability), the worst-case (lowest probability), and the mean performance across all possible contexts for each query.

Figure 2 presents the results of an empirical study on 2-hop, 3-hop, and 4-hop reasoning samples. For each k-hop type, we randomly sample 200 samples from the validation set for evaluation. We evaluate 4 different LLMs: Llama-3.1-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023a), Qwen2.5-7B-Instruct (Team, 2024), and Phi-3.5-mini-instruct (Abdin et al., 2024). Across all models, the performance gap between the best and worst cases is consistently pronounced, with at least 10 points in terms of F1 score. As the number of hops increases, indicating more complex reasoning types, the performance variation grows, reaching up to 26 points in terms of F1 score. Given that the context in this study consists solely of supporting documents, the performance disparity is expected to widen further in real-world tasks where noisy documents are included in the context.

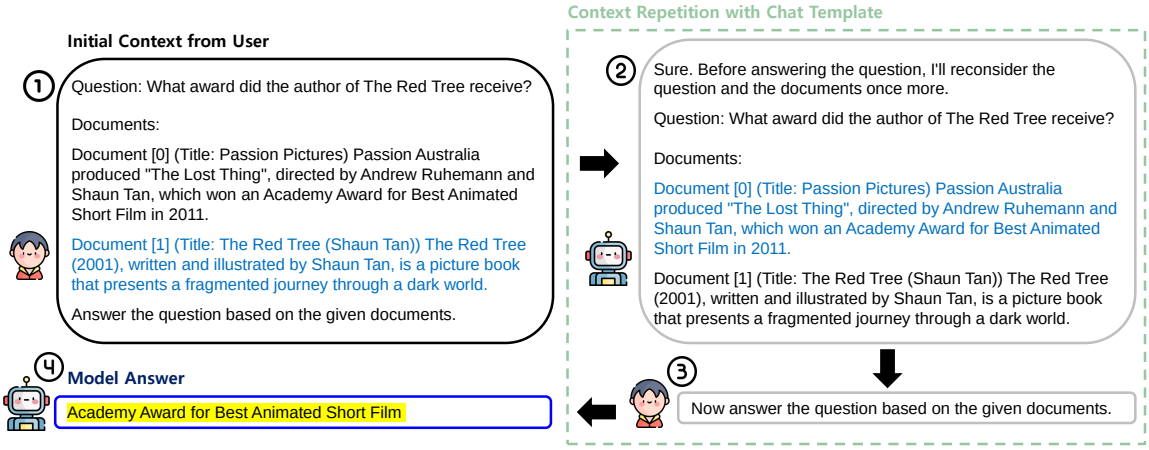

Figure 3: Illustration of CoRe where the model understands the context with the optimal order of documents [1] and [0]. Text with a white background is the prompt, and text with a yellow background is the model output.

Based on these findings, we demonstrate the answer to the research question: *the multi-hop reasoning performance of LLMs is indeed significantly sensitive to the order in which supporting documents are presented*. This aligns with the rationale that decoder-only LLMs, operating with causal attention, interpret the context in a fixed left-to-right order. In real-world scenarios, common prompting methods such as retrieval or search-based approaches do not guarantee that supporting documents will be presented to the model in an optimal order. Consequently, the model is often exposed to a misordered context that is disadvantageous for reasoning, leading to degraded performance. We term this issue the **misordered context** problem, and in the following sections, we describe our approach to addressing this problem.

3.2 Context Augmentation Approach

For the misordered context problem, we formalize a solution from the perspective of context manipulation. We consider a k-hop reasoning scenario, where the model must derive an answer based on a prompt consisting of a query Q, context C, and instruction I. The context C is composed of k supporting documents as well as multiple noisy documents, structured as follows:

$$C = (n_0, d_1, n_1, \cdots, d_k, n_k), \tag{1}$$

where d_i is the i-th supporting document within the context, and n_i is an arbitrary number of noisy documents located between supporting documents, as well as at the beginning and the end of the context. At this point, we define an order σ for the supporting documents as follows:

Definition 1. *Let $\sigma(\cdot)$ be a permutation function for $(1, 2, \cdots, k)$. Then, order σ is defined as*

$$\sigma(d_1, d_2, \cdots, d_k) \triangleq (d_{\sigma(1)}, d_{\sigma(2)}, \cdots, d_{\sigma(k)}).$$

Based on the definition of order, we define the set of contexts that include supporting documents in a specific order σ as follows:

Definition 2. *Let σ be an order of supporting documents. Then, we define $\mathcal{C}_\sigma$ as the set of all contexts that include the supporting documents in order σ:*

$$\mathcal{C}_\sigma \triangleq \{C \mid C = (n_0, d_{\sigma(1)}, n_1, \cdots, d_{\sigma(k)}, n_k), \\ \forall n_0, n_1, \cdots, n_k\}.$$

The order σ defined above can exist in a total of $k!$ different configurations, and thus there are $k!$ possible sets $\mathcal{C}_\sigma$. The initially provided context for answering the query belongs to some arbitrary $\mathcal{C}_\sigma$, and when σ is the optimal order, the model is expected to perform at its best. We define the optimal order σ^*_θ for a specific model θ as follows:

Definition 3. *Let θ be a model that performs multi-hop reasoning. Then, we define σ^*_θ as the optimal order of the given documents d for the model θ:*

$$\sigma^*_\theta \triangleq \underset{\sigma}{\operatorname{argmax}} \underset{C \in \mathcal{C}_\sigma}{\mathbb{E}} \left[p_\theta \left(a \mid Q, C, I \right) \right].$$

That is, the optimal order of supporting documents for a given model is defined as the order that maximizes the expected probability of generating the correct answer in the presence of arbitrary noisy documents. Based on the formalization so far, we aim to find a function that maps a given context C to another context C', which has the supporting documents in the optimal order. The goal is to

increase the probability of the model generating the correct answer. Therefore, we introduce an augmentation function $f(\cdot)$ that satisfies the following conditions:

$$f(C) \in \mathcal{C}_{\sigma^*_\theta}. \tag{2}$$

3.3 Context Repetition

Now, we study the augmentation function discussed earlier. We define an augmentation function $f^{(k)}_{\text{rep}}(\cdot)$ as follows:

$$f^{(k)}_{\text{rep}}(C) = \underbrace{C \oplus C \oplus \cdots \oplus C}_{k \text{ times}}. \tag{3}$$

Here, C represents the given context, and $\oplus$ denotes the concatenation operation. Hence, the function $f^{(k)}_{\text{rep}}(C)$ creates an augmented context by just repeating the given context k times. The following theorem provides an important property of $f^{(k)}_{\text{rep}}(\cdot)$ in relation to σ:

Theorem 1. *For a given context C with its supporting documents ordered according to some permutation τ, the augmented context $f^{(k)}_{rep}(C)$ belongs to the set $\mathcal{C}_\sigma$ for any permutation σ. Formally,*

$$f^{(k)}_{rep}(C) \in \mathcal{C}_\sigma, \quad \forall \sigma.$$

The proof of this theorem can be found in Appendix A. From Theorem 1, we can derive the following corollary:

Corollary 1.1. *$f^{(k)}_{rep}(C)$ always belongs to the set of all contexts that present the supporting documents in the optimal order. Formally,*

$$f^{(k)}_{rep}(C) \in \mathcal{C}_{\sigma^*_\theta}.$$

Based on Corollary 1.1, we conclude that $f^{(k)}_{\text{rep}}(\cdot)$ is an appropriate augmentation function that satisfies Eq. 2. We propose to use this function as a method for solving the misordered context problem. Specifically, we introduce the **co**ntext **re**petition (**CoRe**) methodology, which repeatedly presents the given context to help the model recognize the supporting documents in the optimal order. Figure 3 illustrates the detailed operation of our methodology. In practice, we implemented this method by utilizing a chat template to ensure that the model naturally repeats the context given by the user.

4 Experiments

In this section, we evaluate CoRe on multi-hop QA tasks and a synthetic task that requires complex reasoning. In addition to the main experiments, we conduct an analysis on the mechanism through which CoRe achieves performance improvements in the model.

4.1 Experimental Settings

We conduct experiments on 3 multi-hop QA benchmarks: HotpotQA, 2WikiMultihopQA, and MuSiQue. For HotpotQA and 2WikiMultihopQA, reasoning type annotations are provided for each query. For these 2 datasets, we create evaluation sets by sampling an equal number of queries from each reasoning type within the validation set. In the case of MuSiQue, as described in Section 3.1, query types are categorized by the number of hops. We use the entire validation set as the evaluation set for this dataset. To construct the context of multiple documents, we combine the supporting documents with distracting (noisy) documents assigned to each sample in the datasets. Detailed statistics for each evaluation set are provided in Appendix B.

In addition to the multi-hop QA tasks, we also evaluate model performance on an additional synthetic task. Specifically, we provide the model information about a list that contains non-negative integers as elements in a chained form consisting of two consecutive elements. Afterward, we ask the model for the first element of the list that contains a specific element. At this point, the specific element is the last element of a list. For example, when there are two lists with 3 elements each, the context and question are structured as follows:

```
All the 2 lists described below
    contain exactly 3 elements.
In the list 0, 381 is positioned
    immediately before 512.
In the list 1, 7123 is positioned
    immediately before 34.
In the list 0, 512 is positioned
    immediately before 1021.
In the list 1, 34 is positioned
    immediately before 6397.
Question: What is the first element
    of the list that contains 1021?
```

Since the elements of each list are presented in sequential order from the first to the last, the process of finding the list that contains the last element and then searching for the first element of that list must be performed in a right-to-left order.

Models	HotpotQA		2WikiMultihopQA				MuSiQue		
	Comparison	Bridge	Compositional	Comparison	Inference	Bridge-Comparison	2-hop	3-hop	4-hop
Llama-3.1-8B	**49.31**	61.19	38.10	55.77	40.75	34.86	32.05	26.24	27.06
+ CoRe	46.93	**66.82**	**48.15**	**58.13**	**44.81**	**65.11**	**39.10**	**38.23**	**34.08**
Mistral-7B	**34.42**	41.14	18.82	32.90	29.77	15.43	15.14	10.31	11.00
+ CoRe	30.41	**45.79**	**27.78**	**35.67**	**36.39**	**29.09**	**20.6**	**15.74**	**12.36**
Qwen2.5-7B	61.96	61.66	41.36	65.47	37.26	39.45	33.56	34.87	37.49
+ CoRe	**67.03**	**70.33**	**54.20**	**70.30**	**42.40**	**64.54**	**44.76**	**43.1**	**45.08**
Phi-3.5-mini	**26.44**	44.17	27.70	45.26	27.00	35.19	17.60	16.80	12.67
+ CoRe	23.29	**49.67**	**37.64**	**50.92**	**32.34**	**36.71**	**28.60**	**26.31**	**26.59**
GPT-4o-mini	**66.92**	68.66	51.96	76.63	43.84	60.96	45.26	42.04	40.04
+ CoRe	64.72	**72.67**	**57.43**	**79.14**	**52.03**	**66.96**	**52.47**	**49.47**	**42.53**

Table 1: Main results of F1 score in the multi-hop QA tasks. All models are instruction-tuned LLMs.

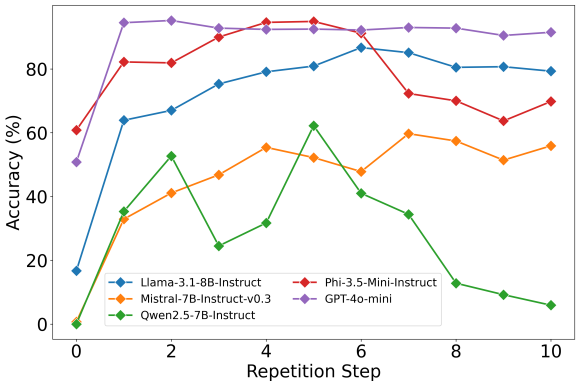

Figure 4: Main results in the synthetic task.

This poses a challenge for models that typically process context in a left-to-right order. Moreover, since this type of task does not involve specific factual knowledge, it minimizes the interference of the model's pre-existing factual knowledge (Xiong et al., 2024). Thus, these characteristics of the synthetic task enable us to establish an environment that clearly exposes the model's inherent misordered context problem. For our main evaluation, we create a synthetic dataset consisting of 1000 samples. Each sample is composed of 10 lists, and each list contains 3 integers between 0 and 9999 that do not overlap with each other.

We conduct experiments using a total of 5 models, including GPT-4o-mini (OpenAI, 2024), in addition to the 4 LLMs used in section 3.1. For Multi-hop QA tasks, we evaluate by measuring the F1 score based on the short answers generated by the models, and for synthetic tasks, we evaluate by measuring accuracy based on whether the answers generated by the models match the correct answers. All prompts used in the experiments can be found in the Appendix C.

4.2 Main Results

Table 1 shows the F1 scores of baseline models and models with CoRe applied for multi-hop QA tasks. Overall, performance improvements were observed across all models and datasets. In the case of HotpotQA, a significant performance improvement was seen in the bridge type. This is because answering bridge-type questions requires reasoning across documents by linking information between them, which is closely related to the order of documents. For 2WikiMultihopQA, a large performance improvement was observed in the bridge-comparison type, with Llama achieving a 30-point improvement and Qwen achieving a 25-point improvement in F1 score. This is because the bridge-comparison type is a complex 4-hop reasoning task that combines both bridge and comparison from HotpotQA. For the MuSiQue dataset, overall performance improved regardless of the number of hops. Taken together, these results indicate that **CoRe contributes significantly to improving the model's multi-hop reasoning performance, especially in tasks sensitive to the order of supporting documents**.

We measure the accuracy of each model as we increase the number of context repetitions up to 10 in the synthetic task. As shown in Figure 4, we can see that the accuracy when no repetition is performed has a low value between 0% and 60%. This demonstrates that existing LLMs are unable to properly perform the synthetic task with simple text inference alone. However, when CoRe is applied, we observe a significant performance improvement, with accuracy increasing by at least 30% points and up to 70% points. Additionally, we can see that many models show a tendency for performance to improve as the number of repetitions increases. This suggests that as repetition progresses multiple times, the model is finding the optimal order. How-

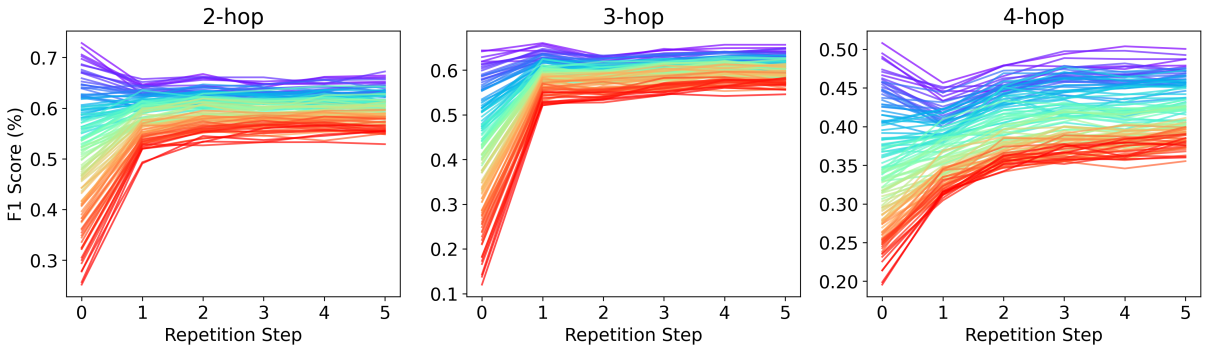

Figure 5: Performance of Llama-3.1-8B-Instruct with permuted contexts of MuSiQue during repetitions. The red line denotes the context in the worst order, and the purple line denotes the context in the best order.

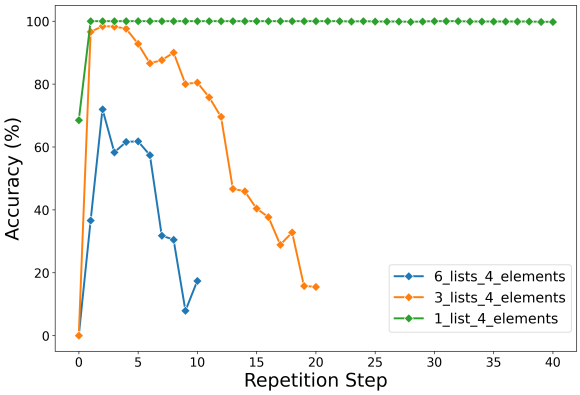

Figure 6: Performance of Qwen2.5-7B-Instruct for contexts with various noise powers in the synthetic task.

ever, for some models, particularly Qwen, there is a tendency for performance to drop again as the repetition steps increase. This appears to be due to noisy information in the context being repeated, negatively affecting its reasoning ability. We provide a detailed analysis of this in Section 4.3.

4.3 Analysis

Does CoRe really identify the optimal order? We conduct additional analysis to investigate whether the performance improvement from CoRe is indeed related to the order of the documents. Similar to the experiments described in Section 3.1, we permute the order of the documents in the context across all possible combinations and measure the model's performance for each permuted context. Specifically, for 200 k-hop samples from MuSiQue, we construct a context with 5 documents, including the k supporting documents and some random $5 - k$ noisy documents assigned to each sample. As a result, each sample has 120 permuted contexts, and we sort the performance of each context from the worst order to the best order, based on $p_\theta(a \mid Q, C, I)$. Figure 5 shows the results for Llama. We plot the worst order with a red line and the best with a purple line. For other cases, we plot their performance with the line of a continuous gradient of colors from red to purple, corresponding to the performance range from the worst to the best case. In other words, the closer the color is to red in the rainbow spectrum, the lower $p_\theta(a \mid Q, C, I)$ is, and the closer to purple, the higher $p_\theta(a \mid Q, C, I)$ is. Across all query types, we find compelling evidence that CoRe contributes to performance improvements in terms of document order: **The closer the context is to the worst order case, the greater the performance improvement from repetition**. Additionally, we observe that as the number of hops increases, performance improves progressively with iterative repetition, indicating that multiple repetitions are more effective when the required reasoning steps increase. Appendix D shows the results for other LLMs in this analysis.

Impact of noisy documents We further analyze the impact of noisy documents on CoRe. In the synthetic task, we measure accuracy by increasing the number of repetitions in cases where the number of lists is 6, 3, and 1, respectively. Here, the decrease in the number of lists means a decrease in the proportion of information unrelated to the search target. Figure 6 shows the results of this analysis. Considering that the length of the context decreases as the number of lists handled in the context decreases, we perform more repetitions in cases with fewer lists. As a result, we confirm that as the proportion of noisy information within the context decreases, the performance degradation caused by excessive repetition is reduced. In

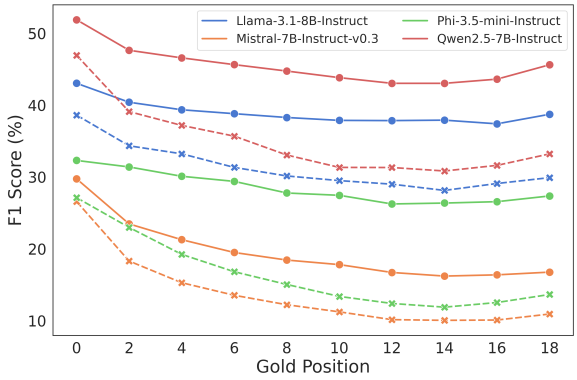

Figure 7: Performance of LLMs for various positions of the supporting document. The solid line represents CoRe, and the dotted line represents the baseline.

Methods	2-hop	3-hop	4-hop
RAG	12.88	8.06	6.70
+ CoRe	**17.04**	**12.44**	**9.07**
IRCoT	23.67	10.64	4.33
+ Dec.	29.03	11.78	6.62
+ CoRe	**37.65**	**19.35**	8.90
+ CoRe & Dec.	36.62	17.87	**11.52**

Table 2: F1 results for retrieve-and-reason task in MuSiQue. Dec. refers to Decompose

particular, when there is no noisy information at all, performance did not decrease even with a large number of repetitions. Since noisy information adversely affects model reasoning at high repetition counts, exploring the valid number of repetitions required for model reasoning could be a potentially interesting future work based on our findings.

5 Further Studies

5.1 Robustness to Positional Bias

We analyzed the positional bias in the context based on the position of supporting documents. Here, positional bias refers to the phenomenon where the QA performance of an LLM varies depending on the position of the supporting documents within the context. This phenomenon is commonly known as the "lost-in-the-middle" problem (Liu et al., 2023). We concatenated k supporting documents in MuSiQue and measured the F1 scores of the baseline and CoRe while shifting the position of the documents from 0 to 18 by increments of 2. As shown in 7, we can observe that the performance gap between the baseline and CoRe increases as the position of the supporting documents moves toward the middle. Particularly for Llama (blue line) and Phi (green line), while the baseline suffers from the "lost-in-the-middle" problem, CoRe demonstrates robustness against positional bias, exhibiting minimal performance variation depending on the position of the supporting documents. Based on these results, we propose CoRe as a solution to mitigate the "lost-in-the-middle" problem in long context QA.

5.2 Enhancement in Retrieve-and-Reason

We evaluate the effectiveness of CoRe in a retrieve-and-reason task, which closely mirrors real-world scenarios. We utilize Contriever (Izacard et al., 2021) as the retriever model, constructing the context with up to 100 retrieved documents from MuSiQue samples. We use Llama for this experiment. We first assess CoRe's performance with (1) the standard RAG, which directly retrieves the top 100 relevant documents to form the context. We also evaluate CoRe with two additional methods: (2) IRCoT (Trivedi et al., 2023), where retrieval is performed sequentially during the CoT reasoning process, and (3) Decompose, where the query is decomposed into sub-queries, and reasoning is conducted by deriving intermediate answers for each sub-query (Press et al., 2023). As shown in Table 2, CoRe significantly improves performance across all methods, demonstrating its effectiveness when integrated into various retrieval-based reasoning approaches. Appendix E presents the experimental details of this section.

Additionally, we report the performance of CoRe when applied in Chain-of-Thought (CoT) scenarios in Appendix D. We also conduct various ablation studies in Appendix F.

6 Conclusion

In this work, we introduce the misordered context problem, which has a critical impact on LLMs' multi-hop reasoning capabilities. After formalizing this problem in terms of context, we propose a simple yet highly effective solution: the CoRe method. We demonstrate the contributions of our approach across various multi-hop reasoning benchmarks, including synthetic tasks, and elucidate its effectiveness through further studies. Our work highlights a key issue crucial for future discussions in multi-hop reasoning tasks with LLMs.

Limitations

The context repetition method proposed in this work increases the input context length by repeating the prompt's contents, which introduces additional costs for memory and time during inference. While the time cost is kept manageable due to the efficient KV caching mechanism, future research could explore ways to repeat only the core contents of the context, thereby minimizing unnecessary expansion. Moreover, as discussed in the analysis section, it is crucial to address the performance degradation caused by noisy documents when repeating the context. A promising avenue for future work would be to combine the inherent capabilities of LLMs with techniques that minimize the influence of noisy documents while maximizing the effectiveness of context repetition. Overall, our work lays the foundational groundwork for addressing the misordered context problem through context augmentation, and future studies could build upon this foundation to further refine and extend the approach.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877–1901.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. 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*, SIGIR 2024. ACM.

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

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

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

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

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

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

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

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien MR Arnold, Vincent Perot, Siddharth Dalmia, et al. 2024. Can long-context language models subsume retrieval, rag, sql, and more? *arXiv preprint arXiv:2406.13121*.

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

Yanyang Li, Shuo Liang, Michael Lyu, and Liwei Wang. 2024. Making long-context language models better multi-hop reasoners. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2462–2475.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Christopher Malon and Bing Bai. 2020. Generating followup questions for interpretable multi-hop question answering. *arXiv preprint arXiv:2002.12344*.

Vaibhav Mavi, Anubhav Jangra, Adam Jatowt, et al. 2024. Multi-hop question answering. *Foundations and Trends® in Information Retrieval*, 17(5):457–586.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *ACL*.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

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

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

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

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024a. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8364–8377.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024b. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353, Bangkok, Thailand. Association for Computational Linguistics.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

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

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

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? In *First Conference on Language Modeling*.

Zheyang Xiong, Vasilis Papageorgiou, Kangwook Lee, and Dimitris Papailiopoulos. 2024. From artificial needles to real haystacks: Improving retrieval capabilities in llms by finetuning on synthetic data. *arXiv preprint arXiv:2406.19292*.

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

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian-guang Lou. 2023. Re-reading improves reasoning in language models. *arXiv preprint arXiv:2309.06275*.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2578–2589.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

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

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

Wenhu Chen Ziyan Jiang, Xueguang Ma. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.

A Proof of Theorem 1

We will prove this theorem by demonstrating that the augmented context $f_{\text{rep}}^{(k)}(C)$, constructed by repeating the original context C k times, inherently arranges the supporting documents in the order specified by the permutation σ.

By applying the augmentation function, we obtain:

$$\begin{aligned} f_{\text{rep}}^{(k)}(C) = & \underbrace{(n_0, d_{\tau(1)}, n_1, \cdots, d_{\tau(k)}, n_k)}_{\text{1st repetition}} \oplus \\ & \underbrace{(n_0, d_{\tau(1)}, n_1, \cdots, d_{\tau(k)}, n_k)}_{\text{2nd repetition}} \oplus \\ & \cdots \\ & \underbrace{(n_0, d_{\tau(1)}, n_1, \cdots, d_{\tau(k)}, n_k)}_{k\text{-th repetition}}. \end{aligned}$$

We can extract the supporting documents in the order specified by σ by selecting the $d_{\sigma(i)}$ from the i-th repetition. All other documents in each repetition of C (i.e., all d_j where $j \neq \sigma(i)$ in the i-th repetition and all n_j) are considered noisy documents. This means they are treated as n'_j in the definition of $\mathcal{C}_\sigma$. The augmented context $f_{\text{rep}}^{(k)}(C)$ can thus be viewed as:

$$f_{\text{rep}}^{(k)}(C) = (n'_0, d_{\sigma(1)}, n'_1, d_{\sigma(2)}, \ldots, d_{\sigma(k)}, n'_k),$$

where each $d_{\sigma(i)}$ is extracted from the i-th repetition, and all n'_j are noisy documents. This directly matches the structure of some $C_\sigma \in \mathcal{C}_\sigma$, satisfying:

$$f_{\text{rep}}^{(k)}(C) \in \mathcal{C}_\sigma.$$

Since σ was arbitrary, this holds for any permutation σ.

B Data Statistics

Table 3 shows the statistics of the evaluation sets in the multi-hop QA tasks conducted in the main experiments.

C Prompts for Experiments

Table 4 and 5 show the prompt template of CoRe for the multi-hop QA tasks and the synthetic task conducted in the section 4, respectively.

D Additional Results

In this section, we describe additional experimental results not reported in the main text. Specifically, we extend the analysis from Section 4.3, where the impact of applying CoRe to each permuted context was only evaluated on Llama, to additional models, including Mistral, Qwen, and Phi. Figures 8 to 10 present the analysis results for each model. Similar to the findings for Llama, we observe that the performance improvement from CoRe increases as the context be close to the worst-order cases for all models. This further solidifies that the performance gains of CoRe stem from augmenting the original context into an optimal context. Meanwhile, we notice a slight performance drop in contexts close to the best order as repetition increases, particularly for models like Mistral and Phi. This trend is more pronounced compared to Llama and can be interpreted as a result of these models being less capable of handling noisy documents.

Additionally, we evaluate the performance of CoRe when applied to Chain-of-Thought (CoT) reasoning in multi-hop QA tasks. As shown in Table 6, CoRe consistently leads to significant improvements, particularly in bridge and compositional question types, aligning with the main results presented earlier. In the case of GPT-4o-mini, some performance metrics remain stable, which we attribute to an observed trade-off: while recall increases, precision decreases. This suggests that when CoT reasoning is combined with CoRe, the model is better at deducing the correct answer, but the length of the generated response tends to increase, resulting in a slight drop in precision. In conclusion, our results demonstrate that CoRe can meaningfully enhance performance even in the context of CoT reasoning. Table 7 shows the prompt template of CoRe for CoT reasoning in the multi-hop QA tasks.

E Experimental Details of Retrieve-and-Reason Task

In this section, we provide the experimental details of the retrieve-and-reasoning task conducted in Section 5.2. The retrieval database consists of 101,962 documents, constructed by ensuring no overlap between the documents assigned to each sample from the MuSiQue training and validation sets. The evaluation set is built by randomly sampling 480 samples from the MuSiQue validation set, comprising 249 2-hop samples, 149 3-hop sam-

ples, and 82 4-hop samples.

The standard RAG approach retrieves the top 100 documents at once, which are then used as the context for prompting the LLM. IRCoT operates by having the LLM perform Chain-of-Thought (CoT) reasoning while generating sentences. After each sentence is generated, the top 10 documents related to the generated sentence are retrieved and concatenated with the previously retrieved documents, updating the context. If the LLM concludes its reasoning with an expression such as "So the answer is," the subsequent output is taken as the final answer. If the context update process exceeds 10 iterations without reaching a conclusion, the entire output generated during the CoT process is considered the final answer. This setup follows the one described in Trivedi et al. 2023. When IRCoT is combined with CoRe, after each CoT sentence is generated, the context of the retrieved documents and previously generated CoT responses is repeated in the prompt.

In the question decomposition setting, before answering the query, the LLM is instructed to decompose the query into several sub-queries. Once the sub-queries are generated, the LLM is directed to perform step-by-step reasoning by following each sub-query, with the process guided by an instruction that ensures the sub-queries are considered during the answering phase. This setting aligns with the prompt design of Press et al. (2023). Table 8 shows the prompt template of query decomposition.

F Ablation Studies

In this section, we conduct an ablation study to analyze the effectiveness of the CoRe methodology. Specifically, we investigate the impact of changing the position where CoRe is applied within the prompt. In our default setup, CoRe is applied through the assistant role, where the context is repeated in the responses generated by the assistant. However, we also consider an alternative setup where the context is repeated in the user role. We evaluate the performance of Llama, Mistral, Qwen, and Phi on 2WikiMultihopQA using F1 scores.

The results, presented in Table 9, reveal that in some cases, applying CoRe in the user role leads to significant performance improvements. Notably, except for Qwen, the other models show substantial gains when applying CoRe in the user role. For instance, Mistral demonstrates a 17-point improvement in compositional-type questions and a 15-point gain in comparison-type questions. Phi also shows an 18-point increase in F1 score on bridge-comparison-type questions. We hypothesize that these improvements are due to the task being framed as a single-turn chat when CoRe is applied in the user role. Some models are predominantly tuned on single-turn chat tasks rather than multi-turn interactions, making it easier for them to reason when CoRe is applied in the user role. By adapting the CoRe methodology to align with the model's training environment, we would create a more familiar reasoning context, leading to larger performance gains. Table 11 shows the prompt template of CoRe applied in the user role position within the prompt.

Additionally, we conduct an ablation study to explore the effects of modifying the repeated context. Specifically, we examine 4 cases where the content of the context is altered during repetition: (1) each documents are paraphrased, (2) each documents are summarized, (3) the order of the documents is shuffled randomly, and (4) the order of the documents is reversed. We evaluate the performance of each of these cases. The experiments are conducted on 2WikiMultihopQA using Llama, where the paraphrasing and summarizing are directly performed by Llama responsible for reasoning.

Table 10 presents the results for each case when repetition is applied once. Interestingly, in comparison tasks, all ablation settings yield higher performance than the baseline setting, which involves simply repeating the context as is. This suggests that the process of recognizing the same content in different forms is particularly beneficial for comparison tasks. For the cases of shuffling and reversing the order of documents, performance improvements are observed in three out of four types, excluding inference. This indicates that varying the order of documents during repetition enables the model to consider a wider range of reasoning chains. As discussed in Section 4.3, future research could explore applying CoRe while reducing the noise power within the context, and actively modifying the form of the context, as in this study, could be another potential direction to consider. Table 12 and 13 show the prompt template of paraphrasing and summarizing, respectively.

G Computational Environment

For the main experiments, we utilize NVIDIA A40 GPU. For the experiments for the retrieve-and-reason task, we utilize NVIDIA H100 80GB HBM3 GPU. All the models except GPT-4o-mini in our work are from `huggingface`. We utilize GPT-4o-mini through `OpenAI` API package.

H Licenses

HotpotQA, 2WikiMultihopQA, and MuSiQue are licensed under the CC BY-SA 4.0, CC-BY-4.0, and Apache-2.0 licenses, respectively. Llama-3.1, Mistral, Qwen, Phi, and GPT-4o-mini are licensed under the Llama 3.1 COMMUNITY LICENSE, Apache-2.0, Apache-2.0, MIT, and OpenAI.

I Usage of AI Writing Assistant

This paper benefited from linguistic support provided by the AI assistant ChatGPT-4o, which contributed by offering paraphrasing and enhancing the original content. No other assistance was sought beyond these services.

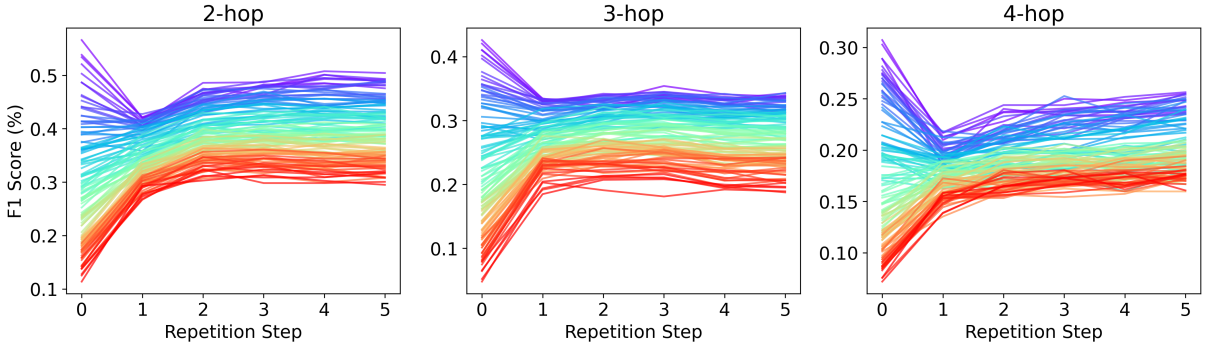

Figure 8: Performance of Mistral-7B-Instruct-v0.3 with permuted contexts of MuSiQue during repetitions. The red line denotes the context in the worst order, and the purple line denotes the context in the best order.

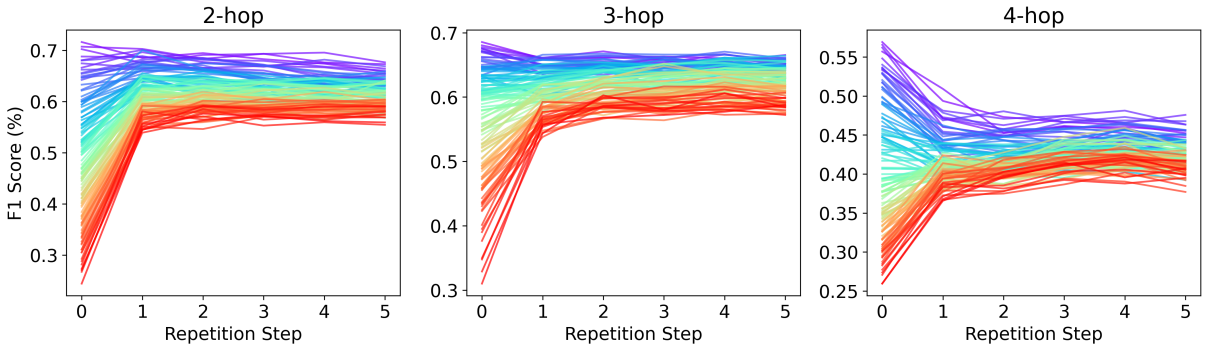

Figure 9: Performance of Qwen2.5-7B-Instruct with permuted contexts of MuSiQue during repetitions. The red line denotes the context in the worst order, and the purple line denotes the context in the best order.

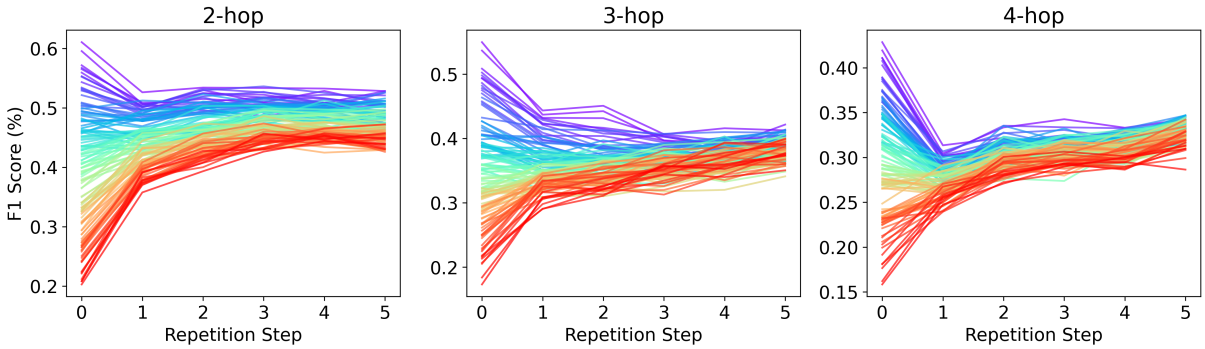

Figure 10: Performance of Phi-3.5-mini-instruct with permuted contexts of MuSiQue during repetitions. The red line denotes the context in the worst order, and the purple line denotes the context in the best order.

Datasets	HotpotQA		2WikiMultihopQA				MuSiQue		
	Comparison	Bridge	Compositional	Comparison	Inference	Bridge-Comparison	2-hop	3-hop	4-hop
# of samples	1000	1000	500	500	500	500	1252	760	405
Context length	869.17	977.33	701.15	567.84	772.21	593.87	1694.4	1827.92	1778.8

Table 3: Data statistics of evaluation sets for each dataset of multi-hop QA tasks in the main experiments. Context length denotes the mean number of words included in the context of each type of sample.

[User role]
Question: {*Question*}

Documents:
Document [0] {*Document 0*}
Document [1] {*Document 1*}
...
Document [N] {*Document N*}

Answer the question based on the given documents. Respond only the answer within a few words after 'Answer:'.

[Assistant role]
Sure. Before answering the question, I'll reconsider the question and the documents once more.

Question: {*Question*}

Documents:
Document [0] {*Document 0*}
Document [1] {*Document 1*}
...
Document [N] {*Document N*}

[User role]
Now answer the question based on the documents. Respond only the answer within a few words after 'Answer:'.

[Assistant role]
Answer:

Table 4: Prompt template of CoRe used in main experiments for the multi-hop QA tasks.

[User role]

```
Information:
All the {k} lists described in the below contain exactly {n} elements.
In the list 0, {lists[0][0]} is positioned immediately before {lists[0][1]}.
…
In the list {k}, {lists[k][0]} is positioned immediately before {lists[k][1]}.
…
In the list 0, {lists[0][n-2]} is positioned immediately before {lists[0][n-1]}.
…
In the list {k}, {lists[k][n-2]} is positioned immediately before {lists[k][n-1]}.

Question: What is the first element of the list that contains {lists[0][n-1]}?
Answer the question based on the given information. Respond only the answer without any explanation
after 'Answer:'.
```

[Assistant role]

```
Sure. Before answering the question, I'll reconsider the question and the documents {t} times
more.

Information:
{Information}

Question: {Question}
```

…

(repeat t-times)

[User role]

```
Now answer the question based on the documents. Respond only the answer without any explanation
after 'Answer:'.
```

[Assistant role]

```
Answer:
```

Table 5: Prompt template of CoRe used in main experiments for the synthetic task.

Models	HotpotQA		2WikiMultihopQA				MusiQue		
	Comparison	Bridge	Compositional	Comparison	Inference	Bridge-Comparison	2-hop	3-hop	4-hop
Llama-3.1-8B	**54.07**	67.74	52.36	70.46	61.21	42.23	49.71	43.03	31.28
+ CoRe	53.07	**71.93**	**53.36**	**72.12**	**63.54**	**77.55**	**51.94**	**44.7**	**37.98**
Mistral-7B	**32.19**	43.68	27.31	39.74	37.22	26.47	21.87	14.18	9.41
+ CoRe	36.04	**52.74**	**37.37**	**45.94**	**48.51**	**53.47**	**25.99**	**15.96**	**11.22**
Qwen2.5-7B	65.93	68.01	49.29	84.83	55.37	77.38	45.33	38.18	36.53
+ CoRe	**65.67**	**72.03**	**51.09**	**85.59**	**59.97**	**83.96**	**49.94**	**45.21**	**40.25**
Phi-3.5-mini	**44.58**	59.76	44.92	81.44	48.48	81.14	28.53	23.39	19.49
+ CoRe	46.23	**62.4**	**46.01**	**83.26**	**51.23**	**82.83**	**34.98**	**31.24**	**30.61**
GPT-4o-mini	**59.29**	71.74	**55.75**	**78.91**	76.44	**88.46**	55.06	49.68	41.84
+ CoRe	57.68	**74.04**	54.93	77.53	**77.65**	88.38	**56.8**	**51.41**	**44.35**

Table 6: Chain-of-Thought results of F1 score in the multi-hop QA tasks. All models are instruction-tuned LLMs.

Chain-of-Thought reasoning

[User role]
Question: {*Question*}

Documents:
Document [0] {*Document 0*}
Document [1] {*Document 1*}
...
Document [N] {*Document N*}

Answer the question based on the given documents.

[Assistant role]
Sure. Before answering the question, I'll reconsider the question and the documents once more.

Question: {*Question*}

Documents:
Document [0] {*Document 0*}
Document [1] {*Document 1*}
...
Document [N] {*Document N*}

[User role]
Now answer the question based on the documents.

[Assistant role]
Let's think step by step.

Extract short answer from CoT response

(After same prompt of user-assistant-user as above prompt for chain-of-thought reasoning ...)

[Assistant role]
Let's think step by step. {*CoT response*}

[User role]
Respond only the answer in a few words after 'Answer:'.

[Assistant role]
Answer:

Table 7: Prompt template of CoRe used in Chain-of-Thought reasoning for the multi-hop QA tasks.

[User role]
Decompose the following question into several sub-questions.
Question: {Question}

[Assistant role]
1.

Table 8: Prompt template of query decomposition for the retrieve-and-reason task.

Models	Compositional	Comparison	Inference	Bridge-Comparison
Llama-3.1-8B	38.1	55.77	40.75	34.86
+ CoRe in assistant	48.15	58.13	44.81	65.11
+ CoRe in user	**53.45**	**69.36**	**47.59**	**68.76**
Mistral-7B	18.82	32.9	29.77	15.43
+ CoRe in assistant	27.78	35.67	36.39	**29.09**
+ CoRe in user	**44.07**	**50.7**	**39.64**	25.29
Qwen2.5-7B	41.36	65.47	37.26	39.45
+ CoRe in assistant	**54.2**	**70.3**	**42.4**	64.54
+ CoRe in user	54.05	69.73	41.64	**68.3**
Phi-3.5-mini	27.7	45.26	27	35.19
+ CoRe in assistant	37.64	50.92	32.34	36.71
+ CoRe in user	**42.56**	**54.89**	**34.51**	**54.37**

Table 9: Results in the ablation study about the position of context repetition

Methods	Compositional	Comparison	Inference	Bridge-Comparison
Baseline (CoRe)	48.15	58.13	**44.81**	65.11
Paraphrase	48.23	61.17	42.32	57.19
Summary	48.44	61.68	40.35	64.79
Shuffle	**51.25**	**63.4**	43.33	65.97
Reverse	50.65	63.35	42.9	**67.11**

Table 10: Results for Llama in the ablation study about the content of repeated contexts.

[User role]
Question: {*Question*}

Documents:
Document [0] {*Document 0*}
Document [1] {*Document 1*}
...
Document [N] {*Document N*}

Answer the question based on the given documents. Respond only the answer within a few words after 'Answer:'.

Look again the input prompt:

Question: {*Question*}

Documents:
Document [0] {*Document 0*}
Document [1] {*Document 1*}
...
Document [N] {*Document N*}

Answer the question based on the given documents. Respond only the answer within a few words after 'Answer:'.

[Assistant role]
Answer:

Table 11: Prompt template of CoRe applied in the user role position.

[System role]
You are a professional paraphraser. Your task is to paraphrase the given text based on the below instructions. Follow the instructions to achieve the desired output.

- Objective: Rewrite the text more thoroughly, changing both vocabulary and sentence structure while preserving the original meaning.
- Instructions: (MOST IMPORTANT) Use your own style for natural paraphrase
Introduce new expressions, alter sentence structure, and rearrange clauses.
Use synonyms and change the form of words (e.g., verbs to nouns, or active to passive voice).
Retain the original message but express it in a noticeably different way.

- Example:
Original: "The quick brown fox jumps over the lazy dog."
Paraphrase: "With swift movements, the brown fox leaps over the dog lying lazily."

Format of the paraphrasing task
- Original: The original text.
- Paraphrase: The paraphrased version of the text based on the above guideline. Provide the output text immediately.

[User role]
Paraphrase the original text below.
Original: {input text}

[Assistant role]
Paraphrase:

Table 12: Prompt template of paraphrasing for the ablation study.

[System role]
ummarize the following text while ensuring that no key information, factual accuracy, or essential meaning is lost. Follow these guidelines:

- Maintain Key Details: All critical points, facts, and arguments from the original text must be preserved.
- Conciseness: The summary should be significantly shorter than the original text while capturing its essence. - Clarity and Precision: Use clear, professional language. Avoid vague phrasing.
- No Alteration of Meaning: Do not add, alter, or infer information that is not present in the original text.

Please follow below pattern of example. Provide the output text immediately.

- Example:
Original Text: The rapid development of artificial intelligence over the last decade has led to significant breakthroughs in various fields, including healthcare, finance, and transportation. However, these advancements also raise concerns about data privacy, job displacement, and the ethical use of AI technologies.
Summary: AI advancements in healthcare, finance, and transportation have been substantial, though concerns about data privacy, job displacement, and ethical issues have emerged.

[User role]
Summarize the following text below.
Original Text: {input text}

[Assistant role]
Summary:

Table 13: Prompt template of summarizing for the ablation study.