Tackling the Length Barrier: Dynamic Context Browsing for Knowledge-Intensive Task

Hongjin Qian[†]
Peking University
Beijing, China
Beijing Academy of Artificial Intelligence
Beijing, China
chienqhj@gmail.com

Zheng Liu[*][†]
Hong Kong Polytechnic University
Hong Kong, China
zhengliu1026@gmail.com

Peitian Zhang
Kelong Mao
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China

Yujia Zhou
Department of Computer Science and Technology
Tsinghua University
Beijing, China
zhouyujia@mail.tsinghua.edu.cn

Xu Chen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
xu.chen@ruc.edu.cn

Zhicheng Dou
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
dou@ruc.edu.cn

Abstract

Knowledge-intensive tasks often require complex reasoning and contextual understanding over long contexts. However, the learning and deployment of long-LLMs remains a challenging problem despite recent progresses. In this work, we propose that the short LLMs have great potentiality for solving knowledge-intensive tasks that have long context, i.e. they can be solved by purely working with oracle short-contexts within the input long-context. On top of this argument, we propose a framework called **DCISO** (DynamiC knowledge-Intensive task SOlver), which enables a short-LLM to address the knowledge-intensive tasks with long context via dynamic context browsing. In our framework, the short-LLM prompts itself to reason for two critical decisions: 1) how to access to the appropriate part of context within the input, 2) how to make effective use of the accessed context. By adaptively accessing and utilizing the context based on the presented tasks, DCISO can serve as a general framework to handle diversified knowledge-intensive long-context problems. We comprehensively evaluate different types of tasks from popular long-context benchmarks, where DCISO is able to achieve a substantially improved performance. Our codes will be released at this repository.

CCS Concepts

• **Computing methodologies → Natural language generation**.

[*]Corresponding Author.
[†]Equal contribution.

Keywords

Large Language Model, Context Length, Knowledge-Intensive Task

ACM Reference Format:
Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Yujia Zhou, Xu Chen, and Zhicheng Dou. 2025. Tackling the Length Barrier: Dynamic Context Browsing for Knowledge-Intensive Task. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3690624.3709240

1 Introduction

Large language models (LLMs) are widely adopted for real-world applications. Many of the applications are knowledge-intensive tasks associated with long-sequence inputs, such as long-document question answering and summarization. As such, the LLMs are commonly expected to have a long working context (*a.k.a.* long-LLMs) in order to confront such demanding scenarios [4]. Unfortunately, the learning and deployment of long-LLMs are still challenging in multiple perspectives. Particularly, many existing LLMs are initially introduced with a limited size of context (*e.g.*, 2K for Llama-1 [37], 4K for Llama-2 [37], 8K for Llama-3 [1]). Although the initial short-LLM can be fine-tuned to establish a much longer context, it is likely to take substantial costs; and more seriously, it is extremely resource-consuming to deploy the long-LLMs. The continually training may also compromise the LLMs' general capability over short contexts [25, 27]. In fact, it remains an open problem to explore new solutions which may tackle knowledge-intensive long-context tasks both effectively and efficiently.

In this paper, we propose that the short LLMs have great potentiality for solving knowledge-intensive tasks that have long context. That is to say, the knowledge-intensive tasks, despite associated with long-sequence inputs, can be addressed by merely working with short-contexts in a strategic way. For example, the reading comprehension or summarization of a book can be solved based on the extraction of necessary key facts from the book.

[1]https://llama.meta.com/llama3/

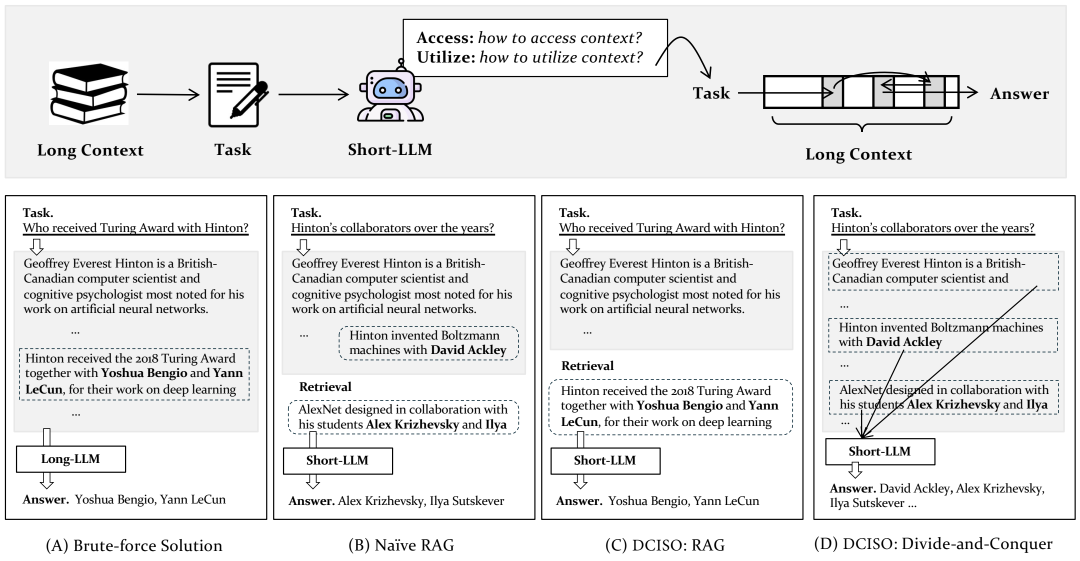
Access: how to access context?
Utilize: how to utilize context?
Long Context
Task
Short-LLM
Task
Answer
Long Context
Task.
Who received Turing Award with Hinton?
Geoffrey Everest Hinton is a British-Canadian computer scientist and cognitive psychologist most noted for his work on artificial neural networks.
Hinton received the 2018 Turing Award together with Yoshua Bengio and Yann LeCun, for their work on deep learning
Long-LLM
Answer. Yoshua Bengio, Yann LeCun
Task.
Hinton's collaborators over the years?
Geoffrey Everest Hinton is a British-Canadian computer scientist and cognitive psychologist most noted for his work on artificial neural networks.
Hinton invented Boltzmann machines with David Ackley
Retrieval
AlexNet designed in collaboration with his students Alex Krizhevsky and Ilya
Short-LLM
Answer. Alex Krizhevsky, Ilya Sutskever
Task.
Who received Turing Award with Hinton?
Geoffrey Everest Hinton is a British-Canadian computer scientist and cognitive psychologist most noted for his work on artificial neural networks.
Retrieval
Hinton received the 2018 Turing Award together with Yoshua Bengio and Yann LeCun, for their work on deep learning
Short-LLM
Answer. Yoshua Bengio, Yann LeCun
Task.
Hinton's collaborators over the years?
Geoffrey Everest Hinton is a British-Canadian computer scientist and
Hinton invented Boltzmann machines with David Ackley
AlexNet designed in collaboration with his students Alex Krizhevsky and Ilya
Short-LLM
Answer. David Ackley, Alex Krizhevsky, Ilya Sutskever ...
(A) Brute-force Solution
(B) Naïve RAG
(C) DCISO: RAG
(D) DCISO: Divide-and-Conquer

Figure 1: Illustration for DCISO. The LLM is prompted to reason for how to access to proper context and how to utilize the accessed context to solve the task. Toy Examples. (A) Brute-force solution. Despite correctness, it is unnecessarily expensive due to the processing of the entire context simultaneously. (B) Naive RAG. It is hard to handle problems like information aggregation, which leads to the incomplete answer. (C) DCISO leverages RAG to tackle the problem, which produces the correct answer in a small cost. (D) DCISO processes the long-context via sequential scan, which correctly solves the problem based on the comprehensively collected information.

The above argument is akin to the working patterns of human beings and modern computers, where arbitrary long-form problems can always be decomposed and solved on top of a limited memory capacity [2, 5]. However, even if the above argument holds, it is still non-trivial to solve the knowledge-intensive tasks purely based on short contexts. This is because different tasks call for distinct ways of accessing and utilizing information from the long context; therefore, there can hardly be any fixed rules to handle all possible situations. To address this challenge, we propose a method, called **DCISO**, where short-LLMs are employed to solve general knowledge-intensive tasks in a dynamic decision-making manner. DCISO operates with two critical reasoning steps. One is the reasoning of **Access**, where the LLM prompts itself to plan for how to access the appropriate part of context within the input. The other one is the reasoning of **Utilize**, where the LLM figures out how to make effective use of the accessed context. Thanks to the above design, DCISO is able to adaptively handle diversified long-context tasks according to their unique nature. For example, given a knowledge-grounded QA problem, the LLM may directly access to the knowledgable context through retrieval, and generate the answer in the form of RAG. Besides, it may sequentially scan the long context chunk-by-chunk if the task calls for the aggregation of specific information from the entire input.

The following toy examples are presented to better illustrate the mechanism of DCISO (Figure 1). Particular, there are two common approaches to tackle knowledge-intensive tasks with long context: (A) the brute-force method based on long-LLMs, (B) the surrogate methods, like RAG [40]. Despite being straightforward, the brute-force method is likely to incur huge unnecessary costs as the problem could be directly solved by simple surrogate methods, like RAG. On the other hand, although the surrogate methods may help in certain cases, they are likely to become useless in other situations. For instance, the RAG-based methods are inappropriate to handle information aggregation problems, as showcased in (B). In contrast, DCISO can handle general knowledge-intensive tasks due to the proper reasoning of how to access and utilize the long-context information based on each specific task. As shown in (C), it can directly access to the needed information via retrieval and generate the answer based on RAG. Meanwhile, it can also process the entire context in a divide-and-conquer manner, which will fully collect the information and solve the problem presented in (D).

We perform comprehensive experiments for DCISO, including both popular real-world knowledge-intensive tasks, like long-context question-answering and summarization of long documents, and a wide variety of synthetic tasks. In our experiments, DCISO is able to achieve equivalent performances as the brute-force methods based on strong long-LLMs, *e.g.*, GPT-4-128K. In many cases, its performances can even notably surpass the brute-force methods, probably due to the elimination of distracting context. Besides, our experiments also underscore the importance of reasoning and adaptability, as DCISO outperforms all short-LLM surrogates with predefined access and utilization of context.

To summarize, our paper makes the following contributions. (1) We identify the research problem of solving long-context knowledge-intensive tasks with short-LLMs. To the best of our knowledge, it is the first study of its kind, which is important to not only address the problem itself but also meaningful to the sustainability and energy-efficient running of AI industry in a broader sense. (2) We propose a novel framework DCISO, which is able to adaptively handle general long-context knowledge-intensive tasks based on the reasoning of how to access and utilize the long context. (3) We empirically verify the effectiveness of DCISO based on its superior performances achieved from low resource-consumption.

2 Related Works

Dealing with knowledge-intensive tasks is a fundamental research problem for LLMs, as many real-world applications involve long-context inputs [15]. The most direct approach to address long-context knowledge-intensive tasks is to increase the working context size of LLMs [1, 3, 25]. A year ago, significant research efforts focused on extending the working context size of LLMs from 4K to 32K [7, 13, 22, 26]. Currently, many popular open-source and close-source LLMs still operate with a context size under 32K [31, 37], such as GPT-3.5-turbo, which has a 16K context length. Recently, research has shifted towards extending LLMs' working context to the million-level. Notably, GPT-4 was updated to a 128K context length not long ago, and the newly released GPT-4o also operates with a 128K context. Moreover, several recent open-source LLMs have been introduced with context lengths exceeding 100K, for example, the Yi series model supports up to 200K [3], and the Phi-3 model operates with 128K [1].

Instead of merely increasing the context length, another approach to address long-context knowledge-intensive tasks involves extracting a short surrogate context from the full context. This includes techniques like retrieval-augmented generation (RAG) and context refinement methods [16, 33, 34]. However, many of these methods utilize task-specific strategies to manage the long context. For instance, RAG methods often deploy retrievers to select relevant context chunks as supporting evidence [21, 40]. Recent studies have criticized the chunking process in RAG for undermining the semantic coherence of the long context and have proposed chunking-free methods to refine the long context into a concise surrogate context [28, 33]. Furthermore, some studies have also explored sequential processing strategies, such as Xu et al. [40], to sequentially process the context in a manner that preserves its integrity.

Lastly, reasoning-based methods also show significant potential for addressing long-context knowledge-intensive tasks [30]. These methods predominantly employ a decision-making process to navigate through the long context sequentially, utilizing reasoning techniques such as in-context learning [12], chain-of-thought [39], and self-reflection [35]. In this paper, DCISO incorporates a decision-making process that dynamically customizes the action trajectory for each query, thereby offering considerable flexibility in accessing and leveraging information to produce the final output answer.

3 Methodology

3.1 Preliminaries

LLMs can be succinctly defined as $\mathcal{Y} = \gamma(q)$, where $\gamma(\cdot)$ represents a selected LLM, q denotes a user query, and $\mathcal{Y}$ refers to the answer produced by the LLMs. As highlighted in many previous studies, *e.g.*, [24], the knowledge embedded in an LLM's parameters is static and, consequently, often fails to adequately address user queries requiring up-to-date or in-depth knowledge. To address this limitation, we can introduce external knowledge (refer to as context $\mathcal{X}$) into the LLMs. Additionally, tasks involving information aggregation (*e.g.*, summarization) also take a context $\mathcal{X}$ as input along with task instructions q. Thus, we can generally define the model's generation process w.r.t. a context $\mathcal{X}$ as: $\mathcal{Y} = \gamma(q, \mathcal{X})$.

As discussed in Section 1, for knowledge-intensive tasks, the context $\mathcal{X}$ can be a long sequence, necessitating that LLMs manage long contexts. However, most existing LLMs were originally introduced with limited context sizes (*e.g.*, 4K). Consequently, these models are unable to process inputs that exceed their capacity without truncation. However, knowledge-intensive tasks require LLMs to have deep understanding over the full context from a global perspective. Therefore, LLMs can hardly accomplish knowledge-intensive tasks well when processing inputs that notably surpass their inherent context limitations. Such an issue can be formally described by:

$$\mathcal{Y} = \gamma(q, \mathcal{X}) \quad \text{s.t.} |\mathcal{X}| \gg L, \tag{1}$$

where L denotes the native context length limit of the LLM. The most straightforward way to address this problem is to increase the LLMs' context length L, mitigating the challenges of long contexts. In this paper, we instead explore solving knowledge-intensive long-context tasks using short-context LLMs (*e.g.*, 4K) without increasing the model's context length L.

3.2 Pilot Study

Despite the potential for fine-tuning LLMs to handle much longer contexts, this approach incurs substantial costs. Besides, directly processing long contexts during the inference stage exponentially increases computing resource consumption, which is not environmentally friendly. Moreover, knowledge-intensive tasks usually require high-level reasoning over multiple parts within the input long context. Compared to full context, providing precise oracle short context might lead to better accuracy. In the following, we conduct a pilot study from both theoretical and empirical perspectives to explore the question: how to utilize short LLMs for long-context knowledge-intensive tasks?

Theoretical Analysis. Suppose we have an input variable $\mathcal{X}$ and an output variable $\mathcal{Y}$, the relevant part of $\mathcal{X}$ given $\mathcal{Y}$ is denoted by $\tilde{\mathcal{X}}$. An ideal $\tilde{\mathcal{X}}$ should capture all relevant features of the original input variable $\mathcal{X}$ in relation to $\mathcal{Y}$. In other words, the optimal $\tilde{\mathcal{X}}$ represents the simplest mapping of $\mathcal{X}$ that accurately preserves the mutual information $I(\mathcal{X}; \mathcal{Y})$. We therefore propose a Markov chain $\mathcal{X} \to \tilde{\mathcal{X}} \to \mathcal{Y}$. According to the data processing inequality (DPI), we have $I(\mathcal{X}; \tilde{\mathcal{X}}) \geq I(\mathcal{X}; \mathcal{Y})$, with equality holding if and only if $\tilde{\mathcal{X}}$ constitutes a *sufficient statistics* [9, 36]. This suggests that, in an optimal setting, we can always find a subset $\tilde{\mathcal{X}} \subseteq \mathcal{X}$ that provides information at least as useful for generating the output $\mathcal{Y}$ as the full context $\mathcal{X}$.

In practical scenarios, obtaining the optimal $\tilde{\mathcal{X}}$ is challenging due to various factors, such as empirical errors [29]. Thus, we can only estimate $\tilde{\mathcal{X}}$. Estimating $\tilde{\mathcal{X}}$ directly from $\mathcal{X}$ might be challenging if $\mathcal{X}$ defines a large variable space. In this situation, we propose decomposing the original input variable $\mathcal{X}$ into a series of subsets, $\mathcal{X} = \{\mathcal{X}_1, \cdots, \mathcal{X}_n\}$ and process each subset variable separately. Thus, according to the *chain rule for mutual information* [9], we have:

$$I(\mathcal{X}, \tilde{\mathcal{X}}) = I(\mathcal{X}_1, \cdots, \mathcal{X}_n; \tilde{\mathcal{X}}) \tag{2}$$

$$= I(\mathcal{X}_1; \tilde{\mathcal{X}}) + \sum_{i=2}^{n} I(\mathcal{X}_i; \tilde{\mathcal{X}} | \mathcal{X}_1, \cdots, \mathcal{X}_{i-1}), \tag{3}$$

which indicates that the mutual information $I(\mathcal{X}, \tilde{\mathcal{X}})$ can be understood as the sum of the mutual information of each subset $\mathcal{X}_i$ and $\tilde{\mathcal{X}}_i$ given all previous subsets.

In the scenario of Eq. 1, the variable $\mathcal{X}$ represents a long context and the variable $\mathcal{Y}$ is the output answer produced by a LLM. Thus, $\tilde{\mathcal{X}}$ can be interpreted as the *minimal necessary context* from the long context $\mathcal{X}$ given the output answer $\mathcal{Y}$. Inspired by Eq. 3, we can estimate an optimal $\tilde{\mathcal{X}}$ using decomposed shorter contexts $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$. Thus, $I(\mathcal{X}; \tilde{\mathcal{X}})$ can be computed by processing each subset $\mathcal{X}_i$ individually. However, as the number of subsets n increases, accounting for all preceding subsets becomes computationally demanding. To alleviate this burden, we propose reducing the number of conditional subsets considered by replacing the entire sequence of previous subsets with a compressed surrogate $\hat{\mathcal{X}}_i$, which is iteratively derived using a compression function $\hat{\mathcal{X}}_i = g(\hat{\mathcal{X}}_{i-1}, \mathcal{X}_{i-1})$. Consequently, Eq. 3 can be reformulated as follows:

$$I(\mathcal{X}, \tilde{\mathcal{X}}) = I(\mathcal{X}_1, \cdots, \mathcal{X}_n; \tilde{\mathcal{X}}) \simeq I(\mathcal{X}_1; \tilde{\mathcal{X}}) + \sum_{i=2}^{n} I(\mathcal{X}_i; \tilde{\mathcal{X}} | \hat{\mathcal{X}}_i)). \tag{4}$$

The equality can be upheld under two specific conditions: (1) the decomposed variables $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ are mutually independent, and (2) the compression function $g(\cdot)$ is optimally designed, ensuring that the compressed surrogate $\hat{\mathcal{X}}_i$ encapsulates all relevant information from the preceding subsets with respect to $\tilde{\mathcal{X}}$. Otherwise, $I(\mathcal{X}, \tilde{\mathcal{X}})$ can only be approximately estimated.

Empirical Analysis. To empirically assess the accuracy of estimating the minimal necessary context $\tilde{\mathcal{X}}$ using decomposed short contexts $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$, we conduct pilot experiments across various tasks requiring long contexts. Specifically, we utilize GPT-4-128K to perform these tasks in two settings: (1) feeding the entire long context into GPT-4-128K in a brute-force manner, instructing the model to directly produce the output answer, and (2) strategically exploring effective decision processes for knowledge-intensive tasks using a short context window (the DCISO setting).

Figure 2 presents the experiment results, which generally indicate that DCISO consistently performs as well as or better than the brute-force setting. In particular, for tasks such as QA, few-shot learning, and synthetic tasks, DCISO outperforms the brute-force setting. This is because the decomposed short contexts for these tasks are more likely to be mutually independent given the input query which can be adequately supported by a few extracted contexts from the long context. By precisely locating these supported context, it can filter out irrelevant context of $\mathcal{X}$ that might otherwise undermine task performance. For tasks like summarization and code completion, the inherent properties of these tasks require considering the mutual dependencies among all decomposed short contexts, making the DCISO setting more challenging. However, as discussed in Eq. 4, when the compression function $g(\cdot)$ is optimal, we can achieve the optimal $\tilde{\mathcal{X}}$. GPT-4 serves as such a strong compression function, ensuring that the compressed surrogate $\hat{\mathcal{X}}_i$ is well-estimated. Consequently, in these tasks, DCISO achieves performance that is equal to or better than the brute-force setting.

Through theoretical analysis, we can posit that many long-context knowledge-intensive tasks can be solved by short LLMs if we can estimate a better minimal necessary context $\tilde{\mathcal{X}}$ from the decomposed short contexts $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ than from the long context $\mathcal{X}$. Empirical analysis supports this assumption, demonstrating that in most cases, the estimation error of deriving $\tilde{\mathcal{X}}$ from the long context $\mathcal{X}$ is often larger than from the decomposed short contexts $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$. This indicates that using short contexts can be comparatively more advantageous than using the full context even for strong LLM like GPT-4 128K. Therefore, we can validate our argument in Section 1: **the short LLMs have great potentiality for solving knowledge-intensive tasks that have long context.**

3.3 The Proposed Method: DCISO

We propose a method called DCISO, which utilizes short LLMs to solve general long-context knowledge-intensive tasks. DCISO begins with an input query q and a long context $\mathcal{X}$, with the goal of producing an output answer $\mathcal{Y}$. Since the underlying LLM in DCISO has a limited context size (we limit DCISO working with 4K context length), directly generating the output answer $\mathcal{Y}$ is infeasible for long-context tasks. To address this, we propose solving long-context tasks by strategically understanding the decomposed short contexts $\mathcal{X} = \{\mathcal{X}_1, \cdots, \mathcal{X}_n\}$. From these short contexts, we aim to extract the minimal necessary context $\tilde{\mathcal{X}}$ to support the generation of the output answer $\mathcal{Y}$.

DCISO achieves this goal through a decision-making process involving iterative interactions between DCISO and the decomposed short contexts $\{\mathcal{X}_1, \cdots, \mathcal{X}_n\}$ with respect to the input query q. In the process, DCISO interact with each short context $\mathcal{X}_i$, employing two types of actions: information access and information utilization. We denote an action at time step i by a_i and denote the relevant context DCISO obtains from the i-th short context $\mathcal{X}_i$ by $\tilde{\mathcal{X}}_i$ The action a_i is predicted by considering the current short context $\mathcal{X}_i$, the input query q, as well as all previous extracted relevant information $\tilde{\mathcal{X}}_{1:i-1}$: $a_i = \gamma(q, \mathcal{X}_i | \tilde{\mathcal{X}}_{1:i-1})$, where $\gamma(\cdot)$ denotes DCISO's underlying LLM.

Predicting the action a_i in a continuous space is challenging as it requires the underling model to reason about highly implicit relations among the input query, the current context, and the previous contexts. Therefore, we define a discrete action space $\mathcal{A}$ comprising:

(1) `[Task Understanding]`: analyzing the query and task for initialization.
(2) `[Retrieve]`: accessing text evidence by a retrieval method.
(3) `[Move]`: accessing the next short text context directly.
(4) `[Append]`: generating relevant context $\tilde{\mathcal{X}}_i$ independently, denoting by $\tilde{\mathcal{X}}_i = a_i(\mathcal{X}_i)$;
(5) `[Merge]`: generating relevant context $\tilde{\mathcal{X}}_i$ with respect to previous extracted relevant information, denoting by $\tilde{\mathcal{X}}_i = a_i(\mathcal{X}_i | \tilde{\mathcal{X}}_{1:i-1})$;
(6) `[Answer]`: answering the user query and returning;
(7) `[Aggregation]`: aggregating all relevant information and returning.

We define our DCISO frame in Algorithm 1.

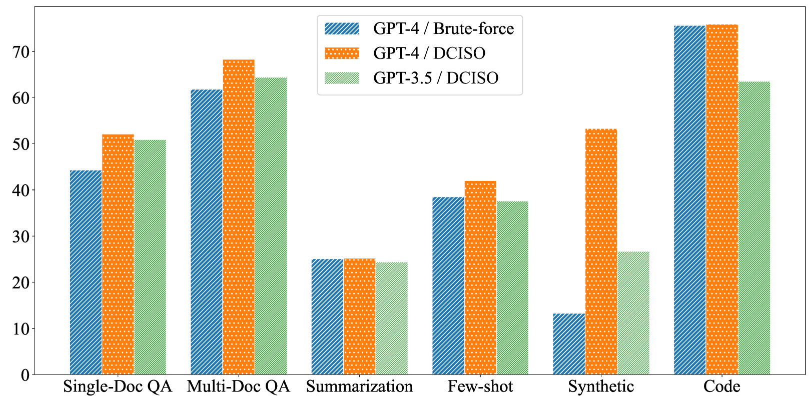

Figure 2: Pilot Study Across Various Tasks: In the Brute-force setting, the entire context is processed by GPT-4-128K. In the DCISO setting, the maximum context length is restricted to 4K, and DCISO is utilized to solve the long-context problem with short context.

Algorithm 1 DCISO Framework

```
1:  Input: Input query $q$, long context $\mathcal{X}$
2:  Output: Answer $\mathcal{Y}$
3:  Decompose long context $\mathcal{X} \leftarrow \{\mathcal{X}_1, \cdots, \mathcal{X}_n\}$
4:  Initialize extracted relevant context $\tilde{\mathcal{X}}_0 \leftarrow$ None
5:  Perform [Task Understanding]
6:  while $i \leq n$ do
7:      Select an action $a_i \leftarrow a_i = \gamma(q, \mathcal{X}_i | \tilde{\mathcal{X}}_{1:i-1}),\ a_i \in \mathcal{A}$
8:      if $a_i$ is [Move] then $i \leftarrow i + 1$, continue
9:      if $a_i$ is [Retrieve] then retrieve evidence from $\mathcal{X} = \{\mathcal{X}_1, \cdots, \mathcal{X}_n\}$
10:     if $a_i$ is [Append] then generate relevant context by $\tilde{\mathcal{X}}_i = a_i(\mathcal{X}_i)$
11:     if $a_i$ is [Merge] then generate relevant context by $\tilde{\mathcal{X}}_i = a_i(\mathcal{X}_i | \tilde{\mathcal{X}}_{1:i-1})$
12:     if $a_i \in$ {[Answer],[Aggregation]} then generate answer $\mathcal{Y} = \gamma(q, \tilde{\mathcal{X}}_{1:i})$, break
13:     $i \leftarrow i + 1$
14: end while
15: return answer $\mathcal{Y}$
```

Though the action space $\mathcal{A}$ comprises only seven actions, DCISO serves as a general framework sufficient for solving most long-context knowledge-intensive tasks. This effectiveness is based on the following reasons:

(1) Flexible accessibility: By utilizing both [Retrieve] and [Move] actions, DCISO can access any short context $\mathcal{X}_i \in \mathcal{X}$ in a flexible trajectory, avoiding the need to browse the entire long context. This makes the information accessing process more efficient.

(2) Accurate information acquisition: Through the [Append] and [Merge] actions, DCISO can either independently extract relevant information from the current short context, appending it to previously extracted information, or merge the current relevant information into the previous relevant information. This capability allows DCISO to acquire relevant information in a compatible manner, making it adaptable to many knowledge-intensive tasks.

(3) Dynamic answering: Using the [Answer] and [Aggregate] actions, DCISO can dynamically utilize the acquired relevant information to produce the target form of the answer (e.g., a short answer for QA tasks via the [Answer] action, or a long answer for summarization tasks via the [Aggregate] action).

In our pilot study depicted in Figure 2, we observe that while GPT-3.5 serves as an inferior foundation model compared to GPT-4, it still demonstrates significant effectiveness when incorporated with DCISO. Given considerations of efficiency and cost-effectiveness, we employ GPT-3.5 as the foundation model for DCISO in the subsequent experiments. We choose GPT-3.5 as the foundation model for DCISO, instead of open-source LLMs. The reason is that GPT-3.5 is a strong, yet efficient model that can generally understand most instructions. However, we found that most open-source LLMs lack these properties in a zero-shot setting. Fine-tuning these open-source LLMs might be helpful, but constructing such instruction data is infeasible and expensive.

4 Experiments

4.1 Experiment Settings

We evaluate DCISO and baseline models on 12 datasets, including: (1) Single-Doc QA: NarrativeQA [23], Qasper [10], and MultiFieldQA [4]. (2) Multi-Doc QA: HotpotQA [41], 2WikiMQA [19], and MuSiQue [38]. (3) Summarization: GovReport [20] and MultiNews [14]. (4) Few-shot Learning: SAMSum [17]. (5) Synthetic Task: Passage Count [4] and Self-Constructed Dataset. (6) Code Completion: LCC [18].

We compare our DCISO with three types of models: (1) Short LLMs (defined as with context length < 32K): Llama2-7B-Chat-4K [37], Llama3-8B-Instruct-8K and Vicuna-v1.5-7B-16K [8]; (2) Long LLMs (defined as with context length ≥ 32K): LongChat-v1.5-7B-32K [26], Mistral-7B-Instruct-v0.2-32K [22], Llama3-8B-80K [42], Phi-3-mini-128K [1] and Yi-9B-200K [3]; (3) Closed-Source LLMs: DeepSeek-v2 (236B MoE model, ranks top-tier in MT-Bench) [11], Claude-3-Haiku[2] and GPT-3.5-turbo-16K[3]. In the experiments, if the context length exceed the model's length limit, following Bai

[2]https://www.anthropic.com/claude
[3]https://platform.openai.com/docs/models

Table 1: Main experiment results. The best results are in bold and the secondary results are marked with underline. We report the average scores (%) on the main tasks. The detailed scores over all dataset are shown in Table 5.

Models	Single-Doc	Multi-Doc	Summ.	Few-shot	Synthetic	Code
Short LLMs (Context Length $<$ 32K)						
Llama2-7B-Chat-4K	24.9	22.5	26.6	40.7	6.3	52.4
Llama3-8B-Instruct-8K	37.3	36.0	26.5	42.7	15.0	57.5
Vicuna-v1.5-7B-16K	28.0	18.6	27.5	40.8	8.9	51.0
Long LLMs (Context Length $\geq$ 32K)						
LongChat-v1.5-7B-32K	28.7	20.6	28.6	34.2	6.8	53.0
Mistral-7B-Instruct-v0.2-32K	31.9	26.0	29.3	<u>43.0</u>	14.0	55.4
Llama3-8B-80K	43.6	43.1	30.2	42.9	19.6	53.6
Phi-3-mini-128K	33.5	38.2	28.8	36.0	19.9	<u>60.1</u>
Yi-9B-200K	29.6	38.7	28.4	14.6	6.5	**72.1**
Closed-Source LLMs						
DeepSeek-v2 (32K)	37.6	<u>49.1</u>	<u>30.8</u>	39.3	14.5	37.0
Claude-3-Haiku (200K)	<u>41.9</u>	45.4	30.1	7.2	<u>25.5</u>	16.9
GPT-3.5-turbo-16K	39.8	38.7	28.1	41.7	18.7	54.7
DCISO (4K)	**47.8**	**56.4**	**31.8**	**44.1**	**27.5**	59.0

Table 2: Statistical information of the datasets utilized in this paper.

Dataset	Narrative	Qasper	MultiField	Hotpot	MuSiQue	2Wiki
Category	Single-Doc QA			Multi-Doc QA		
Num of Samples	200	200	150	200	200	200
Ave. Length	18,409	3,619	4,559	9,151	11,214	4,887
Metric	F1	F1	F1	F1	F1	F1
Dataset	GovReport	MultiNews	SAMSum	PCount	Self	LCC
Category	Summarization		Few-shot	Synthetic		Code
Num of Samples	200	200	200	200	32	500
Ave. Length	8,734	2,113	6,258	11,141	39,420	1,235
Metric	Rouge-L	Rouge-L	Rouge-L	Accuracy	F1&Accuracy	Edit Sim

et al. [4], we truncate the context from the middle since the front and end of the context may contain crucial information.

We evaluated all models on 12 datasets, as shown in Table 2. Most of these datasets are provided by the LongBench benchmark [4]. Following LongBench, we used F1-score, accuracy, and edit similarity as the evaluation metrics. Additionally, we manually annotated a self-constructed dataset comprising long contexts from practical scenarios, such as the full schedule of the Olympic Games and the complete list of accepted papers at ACL. The queries in the self-constructed dataset involve reasoning over the entire long context. For example, “Who has the most accepted papers at ACL 2023?” These queries require the model to accurately understand the long context and perform reasoning, making them highly challenging. The details of the self-constructed dataset are in Table 4.

4.2 Implementation Details

DCISO begins with the `[Task Understanding]` action after receiving the input query and context. For the synthetic task, we prompt the LLM to reformulate the query for better adaptation to DCISO. Based on the output of the `[Task Understanding]` action, DCISO adopts different strategies to perform the task. Specifically, “option [1]” directs DCISO to utilize a retriever to rank all chunks of the long context. In this paper, we employ BGE-Reranker-Large as the retriever [6]. For “option [2]” and “option [3]”, DCISO sequentially processes each short context, respectively. After processing each short context, if the output is not "null", the newly summarized context is added to the "previous summarization".

Once all short contexts are processed, DCISO aggregates all relevant information to produce the final answer. At this stage, we use the prompt provided by LongBench, replacing the full context with the surrogate context produced by DCISO. For “option [4]”, DCISO utilizes the prompts provided by LongBench to process each short context and produces the answer as soon as the proper information is found. We modified the prompt by adding the instruction “If no answer can be found in the text, please output "null"”. This allows DCISO to skip irrelevant short contexts, performing the `[Move]` action. Specifically, for the Code Completion task, DCISO reversely browses the context code from near to far as the near context are

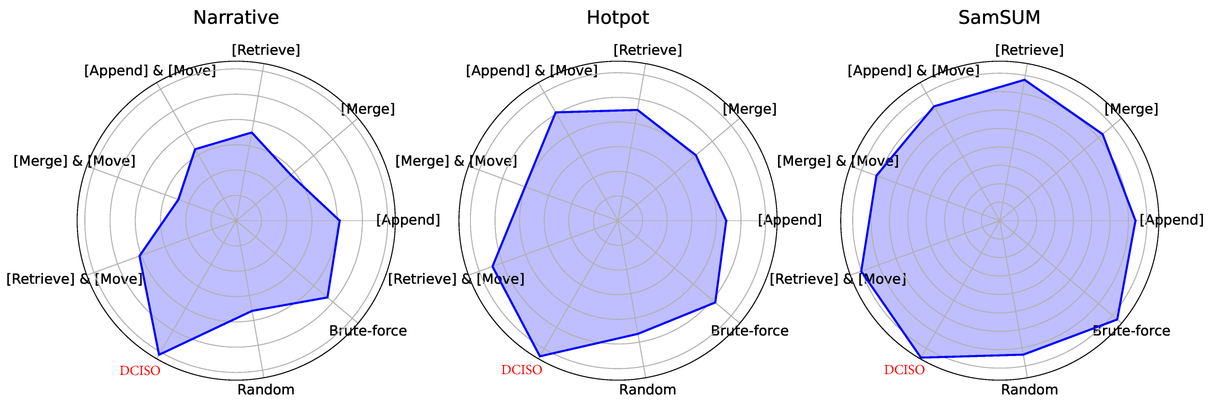

Figure 3: Performance comparison on different context processing strategies in the ablation study. NarrativeQA (left) is a single-doc QA task. HotpotQA (middle) is a multi-doc QA task. SamSUM (right) is a few-shot learning task.

more useful to predict the code completion. For detailed information on the prompts utilized for all actions in DCISO, please refer to our code repository.

We evaluate all baseline models following the settings provided in LongBench [4]. We use a node with 8 A100 80G GPUs to conduct all experiments.

4.3 Main Results

Table 1 shows the overall experimental results for all models across all tasks. From the table, we derive several key findings:

First, DCISO, with a context length of 4K, outperforms all baseline models in all tasks except for the Code Completion task. This result verifies DCISO's capability to effectively solve long-context tasks by strategically processing decomposed short contexts.

Second, long LLMs generally perform better than short LLMs, indicating the effectiveness of fine-tuning LLMs to adapt to long contexts. However, the performance of long LLMs is not consistently stable across different tasks. For example, Yi-9B-200K excels in the Code Completion task but does not show consistent performance in other tasks such as single-doc QA, few-shot learning, and synthetic tasks. This inconsistency suggests that adapting LLMs to long contexts may compromise their general abilities.

Last, DCISO consistently surpasses its underlying LLM, GPT-3.5-turbo-16K, across all tasks by a notable margin. This demonstrates that DCISO can achieve improved performance while simultaneously reducing resource costs, making DCISO an environmentally friendly method.

4.4 Ablation Study

To investigate the necessity of DCISO's design, we conduct ablation studies by changing DCISO's action space $\mathcal{A}$, resulting in different information acquisition strategies.

We experiment with the following settings: (1) `[Retrieve]` only: Directly retrieve the most relevant short context. (2) `[Merge]` only: Sequentially process all short contexts while considering the previously processed context. (3) `[Append]` only: Sequentially process all short contexts independently. (4) `[Merge]` & `[Move]`: Selectively process short contexts while considering the already processed context. (6) `[Append]` & `[Move]`: Selectively process short contexts independently. (7): `[Retrieve]` & `[Move]`: Retrieve the top-k relevant short contexts and selectively process a few of them. (8): Brute-force: Directly produce the answer based on the entire long context. (9) Random: For each short context, randomly select an action. Based on the acquired information from each strategy, DCISO then selects either the `[Answer]` or `[Aggregation]` action to produce the answer.

[4]https://github.com/THUDM/LongBench

Figure 3 illustrates the results, from which we find that:

(1) Compared to fixed processing strategies, DCISO customizes the action trajectory for each query, resulting in notable performance improvements. This finding emphasizes the importance of the dynamic capabilities of DCISO.

(2) DCISO is particularly effective in single-doc QA and multi-doc QA tasks, as it can accurately select the minimal necessary context required to answer the input query, filtering out irrelevant information from the long context.

(3) In the few-shot learning task, DCISO does not significantly outperform the fixed strategies. This is attributed to the numerous in-context examples provided within the task, which offer substantial guidance, thus diminishing the impact of the number of in-context examples on the final performance.

4.5 Case Study: Model Behavior Analysis on Self-Construct Dataset

In Table 3, we present two case studies from the self-constructed dataset. These cases are particularly challenging as they require reasoning across the entire long context. Despite having sufficient context size, LLMs struggle to generate correct responses. In contrast, DCISO dynamically customizes solutions for each case, thereby effectively solving the problems using a shorter context length.

For the first query, DCISO performs `[Append]` or `[Move]` actions across all short context along with a rewritten query, "Extract paper information in the following list that have only one author," derived via `[Task Reasoning]`. After processing all short contexts, DCISO employs the `[Aggregation]` action to compile the final answer. This approach simplifies the task compared to directly extracting a

Table 3: Case study on the self-constructed dataset. Correct answers are marked in teal, incorrect answers in red, and ambiguous answers in orange.

Query: How many papers in ACL 2023 only have one author? **Context**: Full accepted paper list in ACL 2023 main conference. (Context length: 45K) **Ground-truth target**: 8 papers
Phi-3-mini-128K: 11 papers **GPT-3.5-turbo-16K**: 0 papers **Claude-3-Haiku-200K**: 1 papers (Acc. Score: 0)
DCISO's action trajectory: `[Task Reasoning]` → `[Append]`→ · · · → `[Append]`→ `[Aggregation]` **DCISO**: 8 papers (Acc. Score: 1)
Query: List all people names that are petrified, separated by comma. **Context**: Full content of Harry Potter and the Chamber of Secrets. (Context length: 122.6K) **Ground-truth target**: Colin Creevey, Justin Finch-Fletchley, Penelope Clearwater, Hermione Granger
Phi-3-mini-128K: Hermione Granger, Ginny Weasley, Mrs Norris (F1-Score: 0.29) **GPT-3.5-turbo-16K**: Colin Creevey, Mrs Norris (F1-Score: 0.33) **Claude-3-Haiku-200K**: Nick, Hermione, Ron (F1-Score: 0.18)
DCISO's action trajectory: `[Task Reasoning]` → `[Move]`→ · · · → `[Merge]`→ `[Aggregation]` **DCISO**: Colin Creevey, Penelope Clearwater, Hermione Granger, Nick, Mrs Norris (F1-Score: 0.71)

numeric answer from the entire long context, mimicking the human process of reading comprehension and producing accurate results.

In the second case, the query necessitates conditional reasoning on each short context. As highlighted in previous research [27], reasoning directly from the entire context risks losing crucial information, particularly in the middle of the long context. Thus LLMs tend to miss key details such as people's names. DCISO addresses this issue by processing only one short context at a step where it extracts information from arbitrary position of the long text with equal accuracy. Additionally, answers marked in orange include non-human names (*e.g.*, cat, ghost) that are misconstrued as people, illustrating a common challenge where models fail to differentiate in-depth entity properties.

4.6 Energy Consumption Analysis

Recently, we have witnessed the remarkable success of LLMs, which are becoming an indispensable part of our daily lives. We believe that in the near future, LLMs will become as ubiquitous as electricity or gas supply, serving as fundamental infrastructure in human society. At that point, the energy consumption of LLMs will emerge as a significant environmental concern. Therefore, it is imperative for the research community to focus on reducing the energy consumption associated with these models. Figure 4 presents an analysis of energy consumption, comparing the brute-force approach with our DCISO method. The y-axis is measured in Joules. The theoretical energy consumption is estimated for 7B LLMs across varying context lengths. We roughly estimate the energy consumption using the formula $\left(\frac{\text{Total Float Operation}}{\text{312 TFLOPS}}\right) \times 400W$, assuming the use of an A100 GPU with a compute capability of 312 TFLOPS for BFLOAT16 operations and a maximum TDP of 400W[5]. The practical energy consumption is estimated by recording the GPU time and GPU power during inference with different context lengths. We use a Llama2-7B-128K [32] and a Llama2-7B-chat-4K [37] for the brute-force setting and DCISO, respectively.

[5]The calculation of total float operations is based on the method outlined in https://www.harmdevries.com/post/context-length/

Figure 4 clearly indicates that longer context lengths significantly increase energy consumption with the brute-force method, especially evident in practical measurements. This difference is primarily due to the need to distribute sequence activation tensors across multiple GPUs in practical experiment, with tensor I/O exacerbating inference latency and thereby inflating energy costs. In contrast, our DCISO method, working with 4K context lengths, shows only a mild increase in energy consumption across contexts, thereby confirming its energy efficiency while maintaining comparable or superior performance on long-context tasks.

4.7 Token Consumption Analysis

In Section 4.6, our analysis confirms that DCISO significantly reduces energy consumption compared to long LLMs. However, most closed-source LLMs, such as the underlying model of DCISO, GPT-3.5-turbo, charge based on token consumption, *e.g.*, US$0.50 per 1M tokens for input and US$1.50 per 1M tokens for output[6]. Consequently, it is crucial to examine whether the decision-making process of DCISO increases token consumption compared to the brute-force method.

To address this issue, we recorded the end-to-end token consumption for three datasets: NarrativeQA, GovReport, and LCC. After token counting, we conclude that DCISO's token consumption was 34.1% of the brute-force method's consumption in NarrativeQA, 112% in GovReport, and 29.5% in LCC. These results indicate that DCISO's token consumption varies significantly across different tasks. For tasks requiring precise context location, such as QA and code completion, DCISO can respond as soon as the relevant context is identified, thereby avoiding the need to process the full context. However, for tasks that necessitate information aggregation, such as summarization, DCISO may require more tokens for prompts in each iteration. In practice, for token-consumption-sensitive LLMs, there might be a trade-off between performance and cost-efficiency, which also varies considerably across different tasks.

[6]https://openai.com/api/pricing/

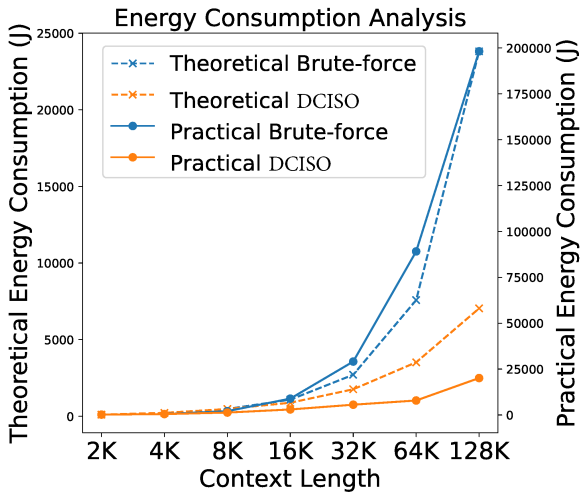

Figure 4: Energy consumption analysis.

5 Conclusion

In this paper, we propose that short LLMs have great potentiality for solving long-context knowledge-intensive tasks, and we validate this claim through both theoretical and empirical pilot study. We propose a method called DCISO to solve long-context knowledge-intensive tasks by decomposing the long context into short contexts and processing them using a decision-making process. We conduct experiments on 12 datasets to compare DCISO with long LLMs and other baseline models. Empirical results verify DCISO's effectiveness in solving long-context knowledge-intensive tasks. Additionally, we discuss the energy consumption of DCISO versus long LLMs, demonstrating that DCISO can achieve comparable performance with significantly less energy consumption.

6 Limitations and Broad Impact

In this paper, we propose DCISO, a method dedicated to solving long-context tasks using short contexts. However, there are several limitations we would like to address in the future work: (1) Although we conduct comprehensive experiments on many tasks and provide theoretical analysis to support our major claim that most long-context tasks are short-context solvable, there may be more complicated scenarios that require understanding the full context in a brute-force setting. DCISO might not be able to process such tasks effectively. (2) As mentioned in Section 3.3, DCISO selects actions from a discrete action space. While we argue that the pre-defined action space is versatile enough to handle most scenarios, a more elegant solution would be to predict actions in a continuous space. We conducted preliminary experiments to explore allowing DCISO to prompt itself to predict actions without a predefined action space, such as writing prompts or code autonomously. These experiments resulted in highly unstable performance, particularly for models like GPT-3.5, as such requirements are still challenging. We believe that with a much stronger foundation model, DCISO could be expected to predict actions in a continuous space. (3) We choose GPT-3.5 as the foundation model for DCISO, instead of open-source LLMs. The reason is that GPT-3.5 is a strong, yet efficient model that can generally understand most instructions. However, we found that most open-source LLMs lack these properties in a zero-shot setting. Fine-tuning these open-source LLMs might be helpful, but constructing such instruction data is infeasible and expensive.

As discussed in Section 4.6, LLMs are likely to become a fundamental infrastructure in the near future. At that scale, their energy consumption will pose significant environmental challenges. As shown in Figure 4, DCISO avoids processing long contexts directly by decomposing them into shorter contexts. This approach significantly reduces energy consumption as the context length increases, leading to substantial positive environmental impacts. We believe that in the future, more research will focus on green AI initiatives. This paper could serve as an initial spark to inspire further research in this direction, potentially resulting in broader social impact.

Acknowledgment

This work was supported by Beijing Municipal Science and Technology Project No. Z231100010323009, National Natural Science Foundation of China No. 62272467, Beijing Natural Science Foundation No. L233008, the fund for building world-class universities (disciplines) of Renmin University of China. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE. Zheng Liu is the corresponding author.

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL]

[2] Ralph Adolphs. 1999. Social cognition and the human brain. *Trends in cognitive sciences* 3, 12 (1999), 469–479.

[3] 01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open Foundation Models by 01.AI. arXiv:2403.04652 [cs.CL]

[4] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 3119–3137.

[5] Randal E. Bryant and David R. O'Hallaron. 2001. Introducing computer systems from a programmer's perspective. In *Proceedings of the 32rd SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2001, Charlotte, North Carolina, USA, 2001*, Henry MacKay Walker, Renée A. McCauley, Judith L. Gersting, and Ingrid Russell (Eds.). ACM, 90–94. https://doi.org/10.1145/364447.364549

[6] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2309.07597 [cs.CL]

[7] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

[8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[9] Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

[10] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4599–4610.

[11] DeepSeek-AI. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. arXiv:2405.04434 [cs.CL]

[12] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[13] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 320–335.

[14] Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 1074–1084.

[15] Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data Engineering for Scaling Language Models to 128K Context. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

[16] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL]

[17] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (Eds.). Association for Computational Linguistics, Hong Kong, China, 70–79.

[18] Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian McAuley. 2023. LongCoder: A Long-Range Pre-trained Language Model for Code Completion. arXiv:2306.14893 [cs.SE]

[19] 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*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 6609–6625.

[20] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient Attentions for Long Document Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 1419–1436.

[21] Gautier Izacard and Edouard Grave. 2021. Distilling Knowledge from Reader to Retriever for Question Answering. In *International Conference on Learning Representations*.

[22] 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. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[23] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2017. The NarrativeQA Reading Comprehension Challenge. arXiv:1712.07040 [cs.CL]

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. 9459–9474.

[25] Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How Long Can Context Length of Open-Source LLMs truly Promise?. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

[26] Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How Long Can Open-Source LLMs Truly Promise on Context Length? https://lmsys.org/blog/2023-06-29-longchat

[27] 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. arXiv:2307.03172 [cs.CL]

[28] Kun Luo, Zheng Liu, Shitao Xiao, and Kang Liu. 2024. BGE Landmark Embedding: A Chunking-Free Embedding Method For Retrieval Augmented Long-Context Large Language Models. arXiv:2402.11573 [cs.CL]

[29] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press.

[30] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332 [cs.CL]

[31] OpenAI. 2023. GPT-4 Technical Report. https://cdn.openai.com/papers/gpt-4.pdf.

[32] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. YaRN: Efficient Context Window Extension of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

[33] Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding Language Model with Chunking-Free In-Context Retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 1298–1311.

[34] Hongjing Qian, Yutao Zhu, Zhicheng Dou, Haoqi Gu, Xinyu Zhang, Zheng Liu, Ruofei Lai, Zhao Cao, Jian-Yun Nie, and Ji-Rong Wen. 2023. WebBrain: Learning to Generate Factually Correct Articles for Queries by Grounding on Large Web Corpus. arXiv:2304.04358 [cs.CL] https://arxiv.org/abs/2304.04358

[35] Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366* (2023).

[36] Naftali Tishby and Noga Zaslavsky. 2015. Deep Learning and the Information Bottleneck Principle. arXiv:1503.02406 [cs.LG]

[37] 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* (2023).

[38] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics* 10 (2022), 539–554.

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.).

[40] Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets Long Context Large Language Models. In *The Twelfth International Conference on Learning Representations*.

[41] 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. arXiv:1809.09600 [cs.CL]

[42] Peitian Zhang, Ninglu Shao, Zheng Liu, Shitao Xiao, Hongjin Qian, Qiwei Ye, and Zhicheng Dou. 2024. Extending Llama-3's Context Ten-Fold Overnight. arXiv:2404.19553 [cs.CL]

A More Experiment Details

In Table 4, we show the details of the self-constructed dataset. In Table 5, we show the detailed experiment results for all 12 datasets.

Table 4: Data details of the self-constructed dataset.

Source	Length	# Queries	Example Query
Accepted paper list of ACL 2023 Main Conference	44,490	7	Who has the most accepted paper in ACL 2023?
The Diamond Sutra	19,993	3	How many chapters of the Sutra?
Schedule of The 2024 Olympic Games	15,844	9	Which day has the most gold medal events?
Subtitle of The Big Bang Theory S3E14	11,136	6	How long does this episode?
The Little Prince	22,471	4	How many planets does the little prince visit?
Harry Potter and the Chamber of Secrets	122,591	3	How many times has the chamber of secret been opened?

Table 5: Main experiment results. The best results are in bold and the secondary results are marked with underline. We report the average scores (%) on all tasks.

Model	Narrative	Qasper	MultiField	Hotpot	MuSiQue	2Wiki
Short LLMs (Context Length $<$ 32K)						
Llama2-7B-Chat-4K	18.7	19.2	36.8	25.4	9.4	32.8
Llama3-8B-Instruct-8K	21.5	43.0	47.5	47.3	23.3	37.5
Vicuna-v1.5-7B-16K	19.4	26.1	38.5	25.3	9.8	20.8
Long LLMs (Context Length $\geq$ 32K)						
LongChat-v1.5-7B-32K	16.9	27.7	41.4	31.5	9.7	20.6
Mistral-7B-Instruct-v0.2-32K	21.6	29.2	47.9	37.7	18.6	21.8
Llama3-8B-80K	28.8	47.4	<u>54.5</u>	55.8	27.4	46.0
Phi-3-mini-128K	21.0	39.4	51.5	48.1	28.2	38.1
Yi-9B-200K	15.6	39.3	33.8	51.4	26.6	38.2
Closed-Source LLMs						
DeepSeek-v2 (32K)	18.3	<u>45.7</u>	48.9	<u>57.7</u>	22.6	**66.9**
Claude-3-Haiku (200K)	<u>30.2</u>	44.0	51.5	51.5	<u>32.5</u>	52.1
GPT-3.5-turbo-16K	23.6	43.3	52.3	51.6	26.9	37.7
DCISO (4K)	**30.6**	**50.6**	**62.1**	**63.5**	**42.5**	<u>63.1</u>

Model	GovReport	MultiNews	SAMSum	LCC	PCount	Self
Short LLMs (Context Length $<$ 32K)						
Llama2-7B-Chat-4K	27.3	25.8	40.7	52.4	2.1	10.5
Llama3-8B-Instruct-8K	30.1	27.6	42.7	57.5	8.0	21.9
Vicuna-v1.5-7B-16K	27.9	27.2	40.8	51.0	6.5	11.3
Long LLMs (Context Length $\geq$ 32K)						
LongChat-v1.5-7B-32K	30.8	26.4	34.2	53.0	1.0	12.5
Mistral-7B-Instruct-v0.2-32K	31.7	26.9	43.0	55.4	2.6	25.4
Llama3-8B-80K	32.3	<u>28.1</u>	<u>42.9</u>	53.6	3.5	35.7
Phi-3-mini-128K	32.6	24.9	36.0	<u>60.1</u>	3.2	36.5
Yi-9B-200K	30.3	26.5	14.6	**72.0**	4.2	8.7
Closed-Source LLMs						
DeepSeek-v2 (32K)	**35.2**	26.3	39.3	37.0	**12.7**	16.2
Claude-3-Haiku (200K)	34.1	26.1	7.2	16.9	5.0	<u>46.0</u>
GPT-3.5-turbo-16K	29.5	26.7	41.7	54.7	4.5	32.9
DCISO (4K)	<u>34.4</u>	**29.2**	**44.1**	59.0	<u>7.2</u>	**47.7**