Scaling External Knowledge Input Beyond Context Windows of LLMs via Multi-Agent Collaboration

Zijun Liu[*1], **Zhennan Wan**[*1], **Peng Li**[†2], **Ming Yan**[†3], **Fei Huang**[3], **Yang Liu**[†1,2]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China
[3]Institute of Intelligent Computing, Alibaba Group
zj-liu24@mails.tsinghua.edu.cn, zhennan018@gmail.com,
pengli09@gmail.com, ym119608@alibaba-inc.com, liuyang2011@tsinghua.edu.cn

Abstract

With the rapid advancement of post-training techniques for reasoning and information seeking, large language models (LLMs) can incorporate a large quantity of retrieved knowledge to solve complex tasks. However, the limited context window of LLMs obstructs scaling the amount of external knowledge input, prohibiting further improvement. Existing context window extension methods inevitably cause information loss. LLM-based multi-agent methods emerge as a new paradigm to handle massive input in a distributional manner, where we identify two core bottlenecks in existing agent orchestration designs. In this work, we develop a multi-agent framework, **ExtAgents**, to overcome the bottlenecks and enable better scalability in inference-time knowledge integration without longer-context training. Benchmarked with our enhanced multi-hop question answering test, **∞Bench+**, and other public test sets including long survey generation, ExtAgents significantly enhances the performance over existing non-training methods with the same amount of external knowledge input, regardless of whether it falls *within or exceeds the context window*. Moreover, the method maintains efficiency due to high parallelism. We believe further study in the coordination of LLM agents on increasing external knowledge input could benefit real-world applications.[1]

1 Introduction

Large Language Models (LLMs) have recently witnessed dramatic progress in parameter scales and context lengths, culminating in context windows that span more than a book-length of text (DeepSeek-AI, 2025b; OpenAI, 2025b; Anthropic, 2025). Yet even these impressive limits remain insufficient for many real-world tasks—multi-hop question answering with Internet, reasoning over enterprise knowledge bases, or writing surveys based on massive academic research—where more external knowledge input often results in better outcomes. Especially, recent research on post-training LLMs to generate long chains of thoughts on reasoning (OpenAI, 2025d; DeepSeek-AI, 2025a) and information seeking (OpenAI, 2025a; Li et al., 2025; Song et al., 2025; Jin et al., 2025) tasks, has shown that increasing the amount of retrieved knowledge within the context window could lead to better task performance (Yue et al., 2025b).

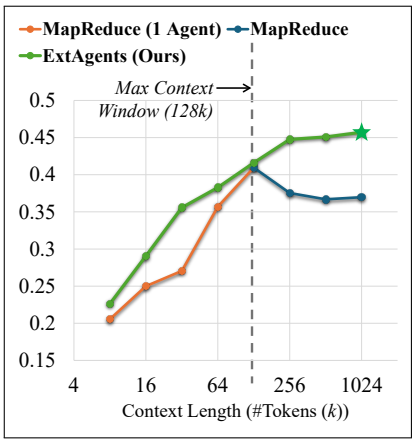

Figure 1: Performance of scaling external knowledge input with ExtAgents (Ours) and LLM×MapReduce (Zhou et al., 2025) on ∞Bench+ (detailed in Figure 4).

For larger input beyond the context window length, the situation is more complicated. When such knowledge is crudely truncated, essential evidence is lost and downstream performance suffers. A natural solution is to train ever longer-context models (Chen et al., 2023b; Peng et al., 2024; Xu et al., 2025; Shang et al., 2025), but this is economically prohibitive and practically brittle: (i) the quadratic complexity of attention (Vaswani et al., 2017) becomes intractable; and (ii) longer-context training data is scarce. Consequently, practitioners turn to *retrieval-augmented generation*

[*] Equal contribution.
[†] Corresponding authors.
[1] Code and data are available at https://github.com/THUNLP-MT/ExtAgents.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10284–10314
July 2-7, 2026

(RAG) (Lewis et al., 2020; Gao et al., 2024; Packer et al., 2024) or *context compression* (Jiang et al., 2024; Qian et al., 2024; Xiao et al., 2024b; Wang et al., 2024c; Hao et al., 2025) pipelines. Unfortunately, both strategies inevitably introduce information loss: RAG is limited by ranking errors that could exclude essential evidence during retrieval, while compressors may discard subtle cues that are only useful once the reasoning chain unfolds. Recent approaches (Trivedi et al., 2023; Zhao et al., 2024b; Zhang et al., 2024b; Zhou et al., 2025) let LLM-based agents collaborate to process long contexts distributedly, reaching state-of-the-art performance on long-context tasks. In this work, we take a step further by asking a question: **Could LLMs consistently improve task performance by scaling the amount of external knowledge input beyond the context window?** Achieving high scalability of external knowledge implies two requirements: (i) a scalable context extension method needs to accept the massive input, and (ii) the knowledge should be effectively integrated in the orchestration of LLMs and agents. Since it is impractical to re-train short-context LLMs, we mainly focus on the scalability of *inference-time knowledge integration beyond context windows*.

We focus on a few tasks that require massive external knowledge, including multi-hop question answering (QA), both over long documents and large knowledge bases, and long survey generation. We found current benchmarks on long-context tasks constructed with biases, that a quantity of queries could be answered by sweeping a small context window over the attached document. For comprehensive validation, we enhance the existing long-context benchmark, ∞Bench (Zhang et al., 2024a), with an automated pipeline, to obtain a long-document-based multi-hop QA evaluation set, **∞Bench+**, alongside with public multi-hop QA (Yang et al., 2018) and long survey generation (Wang et al., 2024d) benchmarks.

In preliminary experiments, we find that the current state-of-the-art LLM-based multi-agent system (Zhou et al., 2025) fails to consistently improve task performance with scaled external knowledge input, and even degrades the performance compared to truncated input (Figure 1). We systematically analyzed existing multi-agent methods, and then spotted two core bottlenecks in the shared designs of agent orchestration: (i) *knowledge synchronization* that agents comprehend the distributed contexts and provide condensed information, where the bottleneck is the "bandwidth" of accessible agents for each agent. and (ii) *knowledge-integrated reasoning*, where the bottleneck is the ratio of redundant information in the reasoning process. To overcome the bottlenecks, we develop a scalable multi-agent framework, **ExtAgents**. Following prior distributional paradigm, the framework partitions the full input into agent-specific context chunks, each sized to fit a small window. ExtAgents simplifies the roles of agents into two: Seeking Agents and Reasoning Agent; featuring two key components: *global knowledge synchronization*, where Seeking Agents to globally exchange and update salient intermediate results instead of locally sharing entire context chunks (Zhao et al., 2024b; Zhou et al., 2025), and *knowledge-accumulating reasoning*, which gradually integrates and increases the updated knowledge from Seeking Agents to Reasoning Agent throughout multiple rounds of reasoning.

We demonstrate the effectiveness and efficiency of ExtAgents with comprehensive experiments on the aforementioned benchmarks. We show that ExtAgents consistently improves task performance with scaled external knowledge input, outperforming the state-of-the-art non-training methods and achieves increasing performance when the input exceeds context windows. We show the generalization of ExtAgents across different QA and long generation tasks, and its compatibility with different LLM families. We also measure the efficiency gain of ExtAgents from high parallelism.

In summary, our contributions are:

- We *introduce and define* the problem of **scaling external knowledge input beyond context windows**, filling a critical gap in current LLM deployment. We also construct an enhanced multi-hop QA benchmark, **∞Bench+**, for corresponding evaluation.
- We systematically study existing LLM-based multi-agent systems for context window extension, and overcome their bottlenecks by proposing a novel framework, ExtAgents.
- We demonstrate the effectiveness and efficiency of ExtAgents on QA and survey generation tasks. With external knowledge input scaling beyond context windows, it consistently improves task performance and significantly outperforms baseline methods.

2 Related Work

Context Window Extension Methods for LLMs (1) ***Retrieval-Based Methods:*** For massive input breaking the context window, RAG (Lewis et al., 2020; Gao et al., 2024) is a common solution to chunk the input into smaller pieces and retrieve relevant ones through indexing (Zhao et al., 2024c), searching (Packer et al., 2024), or ranking (Wang et al., 2024b). The granularity ranges from token- (Xiao et al., 2024a) to document-level (Chen et al., 2025). Recently, iterative retrieval is shown to be effective for multi-hop tasks (Trivedi et al., 2023) and with scaled retrieved documents (Yue et al., 2025b). Since the amount of acceptable retrieved information is limited by the context window, ranking errors are the decisive factor in performance. (2) ***Compression-Based Methods:*** Orthogonal to RAG, long contexts can be compressed into smaller representations, including parametric states (Han et al., 2024; Xiao et al., 2024b; Wang et al., 2024c; Hao et al., 2025; Yang et al., 2025) and non-parametric summaries (Chen et al., 2023a; Jiang et al., 2024; Qian et al., 2024; Edge et al., 2025), which are then fed into LLMs. However, the compression is often lossy due to the limited context window and compression capabilities of compressor model. (3) ***Multi-Agent Collaboration Methods:*** LLM-based multi-agent systems have emerged as a new paradigm to handle massive input in a distributional manner. We analyze existing methods (Zhao et al., 2024b; Zhang et al., 2024b; Zhou et al., 2025; Li et al., 2024; Wang et al., 2025) in Section 3.2 in detail. Though the approach could be viewed as mixing retrieval and compression, it contains more nuanced orchestration of agents.

LLM-Based Multi-Agent Collaboration on General Tasks For general tasks, LLM-based multi-agent systems (Li et al., 2023; Du et al., 2024; Wu et al., 2024; Liu et al., 2024; Zhuge et al., 2024; Zhang et al., 2025; Yue et al., 2025a) have been proposed to collaboratively process workloads, resulting in improved performance. Different applications has been explored, including coding (Hong et al., 2024), science research (Yamada et al., 2025), decision making (Wang et al., 2024a), embodied game playing (Chen et al., 2024b), etc. Recent studies also post-train LLMs to enhance collaboration in various tasks (Qiao et al., 2024; Subramaniam et al., 2025; He et al., 2025; Liao et al., 2025).

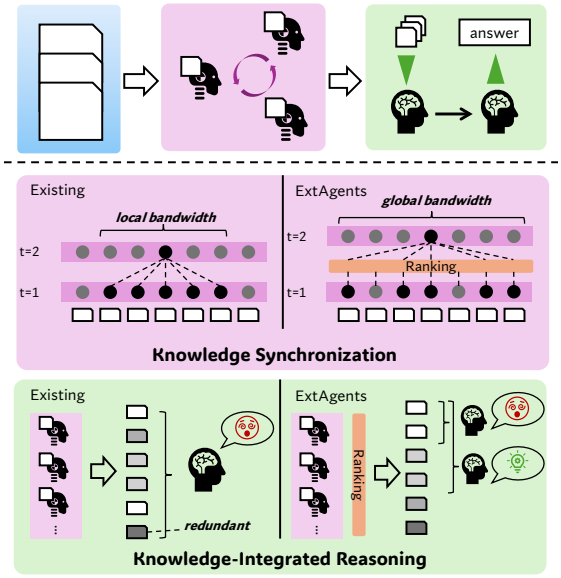

Figure 2: The illustration of multi-agent collaboration methods for context window extension on LLMs. EXTAGENTS alleviates bottlenecks in knowledge synchronization and reasoning processes (detailed in Table 1).

3 Scaling External Knowledge Input Beyond Context Windows of LLMs

3.1 Problem Definition

Since real-world applications (OpenAI, 2025a; Wei et al., 2025; Inc., 2025) frequently demand extensive external knowledge whose scale could dramatically surpass LLM context windows, the need for a scalable approach is paramount.

Formal View For each task query $q \in \mathcal{Q}$, a given external knowledge source $\mathcal{K}$ could be a long document attached ($\mathcal{K} = f_{\mathcal{Q}}(q)$) or several document pieces retrieved from large knowledge bases $\mathcal{C}$ ($\mathcal{K} = f_{\mathcal{C}}(q)$). In both cases, the knowledge source could be partitioned into N chunks $\mathcal{K} = \{d_1, \ldots, d_N\}$, where d_i is a chunk with length $|d_i|$. For long documents, this could be done by simple splitting, and sophisticated chunking methods are available for future work; for knowledge bases $\mathcal{C}$, chunk d_i could be a retrieved document piece with further aggregation or splitting. The former is often used in QA tasks attached to long documents (Zhang et al., 2024a), while the latter is common in open-domain tasks (Yang et al., 2018). The query is processed by an LLM θ with a maximum context length $L, \max_i\{|d_i|\} < L$ (e.g., 128k, or more tokens), under the guidance of pre-defined prompts and agent orchestration (workflows) π_θ, to give out answer y. In this work, we focus on tasks where the total length of knowledge

Method	Sync. Bandwidth	Reasoning Context ($\mathcal{M}_r$)	Parallelized Component
Chain of Agents (Zhang et al., 2024b)	2	$\{m_{N,N}\}$	None
LongAgent (Zhao et al., 2024b)	2	$\{m_{i,t}\}_{1\le i\le N, 1\le t\le T}$	Sync.
LLM×MapReduce (Zhou et al., 2025)	$O(\frac{L}{\lvert m\rvert})$	$\{m_{i,T}\}_{1\le i\le N}$	Sync.
ExtAgents (Ours)	N	$\mathbf{Top}_{2^s}(\{m_{i,t^\star}\}_{1\le i\le N})$	**Sync. & Reasoning**

Table 1: Comparisons of existing LLM-based multi-agent methods for context window extension and our EXTAGENTS on the bottlenecks in agent orchestration of knowledge synchronization (sync.) and reasoning processes.

source is much larger than the context window, i.e.,

$$y = \pi_\theta(q, \mathcal{K}) \quad \text{with} \quad |\mathcal{K}| \gg L. \tag{1}$$

Objective The overall objective is to maximize the task performance with respect to the amount of external knowledge input. For tasks with ground-truth answers $y^\star$, the objective is formulated as:

$$\max_{\pi} \mathbb{E}_{q\sim\mathcal{Q},\mathcal{K}\sim\{f_\mathcal{Q}(q),f_\mathcal{C}(q)\}}\left[\mathrm{Score}_{\mathrm{pair}}(y,\, y^\star)\right] \quad \text{with fixed} \quad \max\{|\mathcal{K}|\}, \theta \tag{2}$$

where $\mathrm{Score}_{\mathrm{pair}}(\cdot,\cdot) \in \mathbb{R}$ is a specific metric with reference (e.g., F1, LLM-as-a-Judge (Zheng et al., 2023)). For open-ended generation tasks without clear ground truths, the objective is

$$\max_{\pi} \mathbb{E}_{q\sim\mathcal{Q},\mathcal{K}\sim\{f_\mathcal{Q}(q),f_\mathcal{C}(q)\}}\left[\mathrm{Score}_{\mathrm{single}}(y)\right] \quad \text{with fixed} \quad \max\{|\mathcal{K}|\}, \theta \tag{3}$$

where $\mathrm{Score}_{\mathrm{single}}(\cdot) \in \mathbb{R}$ is a reference-free metric (e.g., LLM with rating principles (Wang et al., 2024d)). The control of maximum input length $\max\{|\mathcal{K}|\}$ is achieved by truncating $f_\mathcal{Q}(\cdot)$ or $f_\mathcal{C}(\cdot)$.

Noticeably, the setting of scaling external knowledge input is different from expanding retrieval knowledge bases (Shao et al., 2024), which does not increase inference costs of LLMs but of the retriever. We argue this is orthogonal to our primary goal towards the scalability of LLM-based agents.

3.2 Review of Existing Multi-Agent Methods

In this section, we review representative LLM-based multi-agent systems for context window extension, including **Chain of Agents** (Zhang et al., 2024b), **LongAgent** (Zhao et al., 2024b), and **LLM×MapReduce** (Zhou et al., 2025). These methods spin up a team of N LLM-based agents. Each agent is attributed a local context chunk d_i, and collectively decides on the answer y. We conclude that these multi-agent methods share a two-stage pattern of *knowledge synchronization* and *reasoning*. The former is designed to comprehend the distributed contexts and provide related knowledge for the latter to generate the final answer. We follow Liu et al. (2024) to incorporate timesteps for modeling agent orchestration, and we identify a core bottleneck in each stage (Figure 2):

1. **Knowledge Synchronization:** At timestep $t \le T$, each agent $a_{i,t}$ digests its local chunk d_i and messages $\mathcal{M}_{\mathcal{G}_{i,t-1},t-1} \subseteq \{m_{j,t-1} | a_{j,t-1} \in \mathcal{G}_{i,t-1}\}$ from a neighbourhood $\mathcal{G}_{i,t-1} = \{a_{i-k_1,t-1}, \ldots, a_{i,t-1}, \ldots, a_{i+k_2,t-1}\}$ of size $|\mathcal{G}_{i,t-1}|$, with the maximum $\max_{i,t}\{|\mathcal{G}_{i,t}|\}$ termed *bandwidth*. Original chunks may also be included ($\mathcal{D}_{\mathcal{G}_{i,t-1},t} \subseteq \{d_i | a_{i,t-1} \in \mathcal{G}_{i,t-1}\}$). It is then prompted to emit an updated message:
$$m_{i,t} = a_{i,t}(q, \mathcal{D}_{\mathcal{G}_{i,t-1},t}, \mathcal{M}_{\mathcal{G}_{i,t-1},t-1}). \tag{4}$$
With smaller bandwidths, more timesteps might be needed to synchronize all the inputs. The bandwidth of Chain of Agents and LongAgent is 2, and the bandwidth of LLM×MapReduce is $O\left(\frac{L}{|m|}\right)$, where $|m|$ is the expected length of a single message. The values of bandwidth generally reflect the reported performance (Zhou et al., 2025), and we conjecture larger bandwidth leads to better performance.
2. **Knowledge-Integrated Reasoning:** An agent a_r collects a subset of messages as the *reasoning context* $\mathcal{M}_r \subseteq \{m_{i,t}\}_{i,t}$ and produces the task answer following a workflow:
$$y = a_r(q, \mathcal{M}_r). \tag{5}$$
According to Jiang et al. (2024), the ratio of redundant information in the reasoning process is a key factor affecting the performance. Chain of Agents and LLM×MapReduce default to put as much information as possible into the reasoning context, which may lead to information overload.

Details are shown in Table 1 and Appendix A. Other multi-agent systems with additional designs orthogonal to the agent orchestration could be analyzed similarly, e.g., Li et al. (2024) is similar to LongAgent with a graph to organize retrieved information. Due to the high costs of long inputs, we use LLM×MapReduce and Chain of Agents as main baselines in our experiments.

Subset	Samples	#Samples	#Tokens (Avg.)
En.QA	All	351	~194k
	¬8k	**157**	**~188k**
Zh.QA	All	189	~1,302k
	¬8k	**56**	**~904k**

Table 2: Statistical information of ∞Bench (Zhang et al., 2024a). "¬8k" denotes samples answerable with an 8k-token chunk are *filtered out*.

Subset	Samples	#Samples
En.QA	All (∞Bench)	351
	¬4k	145
	¬8k (∞Bench+)	**157**
	¬16k	164
	¬32k	171

Table 3: Sensitivity checks on different sweeping window lengths on the En.QA subset of ∞Bench. The numbers of remaining valid samples are reported.

3.3 Challenges: Evaluation and Implementation of Scalable Approaches

∞Bench+ Zhang et al. (2024a) introduced ∞Bench to evaluate LLMs on long-document inputs. However, many samples exhibit bias, as answer-related content is confined to small regions easily handled within limited windows. To address this, we construct **∞Bench+**, an enhanced multi-hop QA test set requiring the benchmarked system to aggregate information across large segments of each document. Using gpt-4o-mini-2024-07-18 (OpenAI, 2024), we discard samples answerable with any 8k-token chunk, thereby exposing bias in some long examples (Table 2). The 8k-token chunk represents the standard that (1) the vast majority of modern LLMs can process *trivially*, and (2) is the shortest chunk size in *subsequent experiments* (Section 5). Sensitivity checks (Table 3) demonstrate that 8k-token sweeping window is robust, which is less coarse compared to 16k (remaining ~2% more samples) or 32k (~4%) and effectively eliminates the "shortcut" queries without aggressively decimating the dataset. As this filtering results in a limited number of samples, making evaluation unstable, we additionally incorporate original ∞Bench samples longer than 128k tokens into ∞Bench+. The resulting ∞Bench+ consists of two subsets: En.QA with 294 samples and Zh.QA with 184 samples. We claim that ∞Bench+ the benchmark does not significantly shift from but enhance the existing long-context QA benchmarks (Hsieh et al., 2024; Yen et al., 2025; Bai et al., 2024).

Preliminary Experiments We test LLM×MapReduce on our ∞Bench+ benchmark with gpt-4o-mini-2024-07-18, and find that the method fails to consistently improve task performance with gradually increasing external knowledge input from 8k tokens (Figure 1). When scaling the input beyond the context window of 128k tokens, the performance shows no advantage over directly truncating context, which does not meet our expectations.

4 EXTAGENTS: A Scalable Solution

4.1 Agent Profiles

We adopt the distributional paradigm to partition the full input into agent-specific context chunks. EXTAGENTS simplifies agent roles into two, corresponding to the two-stage orchestration, compatible to process any amount of input:

- **Seeking Agents:** Comprehending assigned knowledge chunks and rating the relevance of a context chunk to the task query. Following Section 3.2, the number of Seeking Agents equals to chunk numbers, i.e., N. Optionally, they can exclude redundant chunks.
- **Reasoning Agent:** Integrating knowledge accumulated from Seeking Agents to generate the final answer. Reasoning Agent identifies the answerability and could refuse to answer if the provided information is insufficient. It is compatible for both multi-hop QA and long generation with switched task prompts.

4.2 Global Knowledge Synchronization

To overcome limited agent interaction bandwidth (Section 3.2), EXTAGENTS implements **global knowledge synchronization**. Unlike previous methods (Zhao et al., 2024b; Zhou et al., 2025), which restrict agent interactions to local neighborhoods, our approach grants every agent global visibility by ranking messages before context assembling, thus maximizing synchronization bandwidth and ensuring propagation of salient information.

Formally, each Seeking Agent $a_{i,t}$ at synchronization timestep t updates its message as:

$$m_{i,t} = a_{i,t}^{(\mathrm{EA})}(q, d_i, \mathcal{M}_{t-1}), \quad \mathcal{M}_{t-1} = \{m_{j,t-1}\}_{j=1}^{N}, \quad (6)$$

where $\mathcal{M}_{t-1}$ represents the global set of messages from the previous timestep, and EA stands for EXTAGENTS. However, when the number of Seeking Agents (N) is large, the amount of information exchanged can break the context window. To mitigate this, each agent $a_{i,t}$ rates the relevance of the

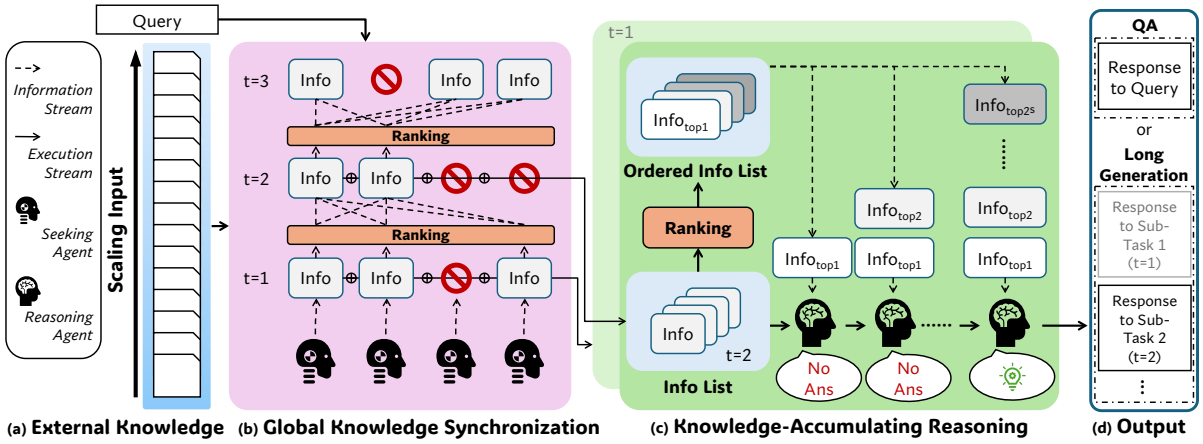

Figure 3: **Overview of EXTAGENTS**: Our framework consists of multiple agents with fixed context windows, that collaboratively process (a) scalable external knowledge inputs beyond the context limit. It features (b) global knowledge synchronization, and (c) knowledge-accumulate reasoning processes by sharing a ranking mechanism at each timestep. Moreover, EXTAGENTS support (d) both multi-hop QA and long survey generation tasks.

Method	HotpotQA		En.QA		Zh.QA	
	F1	Input	F1	Input	F1	Input
DeepSeek-R1-Distill-Llama-8B						
Direct Input	.159	32k	.097	32k	.143	32k
gpt-4o-mini-2024-07-18						
Direct Input	.204	128k	.182	128k	.204	128k
DRAG	.482	128k	-	-	-	-
IterDRAG	.413	128k	-	-	-	-
LLM×MapReduce	-	-	.374	128k	.436	128k
EXTAGENTS (Ours)	**.534**	**1024k**	**.382**	**1024k**	**.482**	**1024k**
Llama-3.1-8B-Instruct						
Direct Input	.254	128k	.237	128k	.315	128k
DRAG	.349	32k	-	-	-	-
IterDRAG	.368	32k	-	-	-	-
Chain of Agents	-	-	.168	32k	.246	32k
LLM×MapReduce	-	-	.254	256k	.345	128k
EXTAGENTS (Ours)	**.412**	**1024k**	**.291**	**1024k**	**.347**	**256k**
gpt-4o-2024-08-06						
EXTAGENTS ($N = 1$)	.553	128k	-	-	-	-
EXTAGENTS	**.597**	**1024k**	-	-	-	-

Table 4: Performance on Multi-Hop QA tasks with the optimal setting (chunk sizes and input lengths) and the corresponding input length (#tokens).

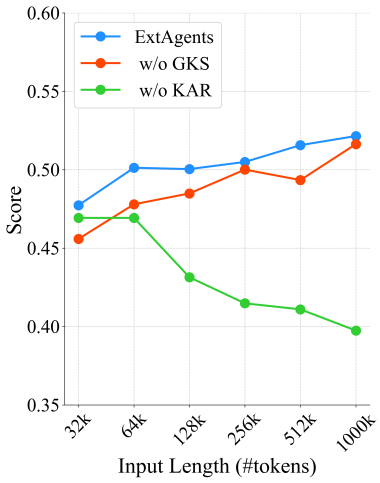

Figure 5: Ablation studies on the global knowledge synchronization (GKS) and knowledge-accumulating reasoning (KAR) on Hotpot QA with gpt-4o-mini.

message $m_{i,t} \in \mathcal{M}_t$ to the task query first, outputing scores $h_{i,t} \in \mathbb{R}_+, i = 1, ..., N$. The rating could be done by appending a prompt, or by using a separate metric tool, e.g., retrieval scores. The rating method is evenly designed for each message by using the same rating principle for LLMs or the same retriever, and is performed distributedly since putting all messages together could be enormous.

$$\text{Top}_k\left(\mathcal{M}_t\right) = \underset{\substack{\hat{\mathcal{M}}_t \subseteq \mathcal{M}_t \\ |\hat{\mathcal{M}}_t| = k}}{\arg\max} \sum_{j' \in \{j | m_{j,t} \in \hat{\mathcal{M}}_t\}} h_{j',t}. \quad (7)$$

By selecting the top-k pertinent messages with k as large as possible, we maintain the global bandwidth and fit within the context window:

$$m_{i,t} = a_{i,t}^{(\text{EA})}(q, d_i, \text{Top}_k(\mathcal{M}_{t-1})) \quad \text{with} \quad |q| + |d_i| + |\text{Top}_k(\mathcal{M}_{t-1})| < L. \quad (8)$$

Alike LLM×MapReduce, all Seeking Agents can run in parallel at each timestep, substantially reducing latency with high parallelism.

4.3 Knowledge-Accumulating Reasoning

After *each knowledge synchronization timestep*, the reasoning process is initiated. To tackle the reasoning bottleneck caused by redundant information overload, Reasoning Agent incrementally integrates the most pertinent messages. Formally, at each reasoning iteration s ($1 \leq s \leq S$), Reasoning Agent selects the top-2^s messages from

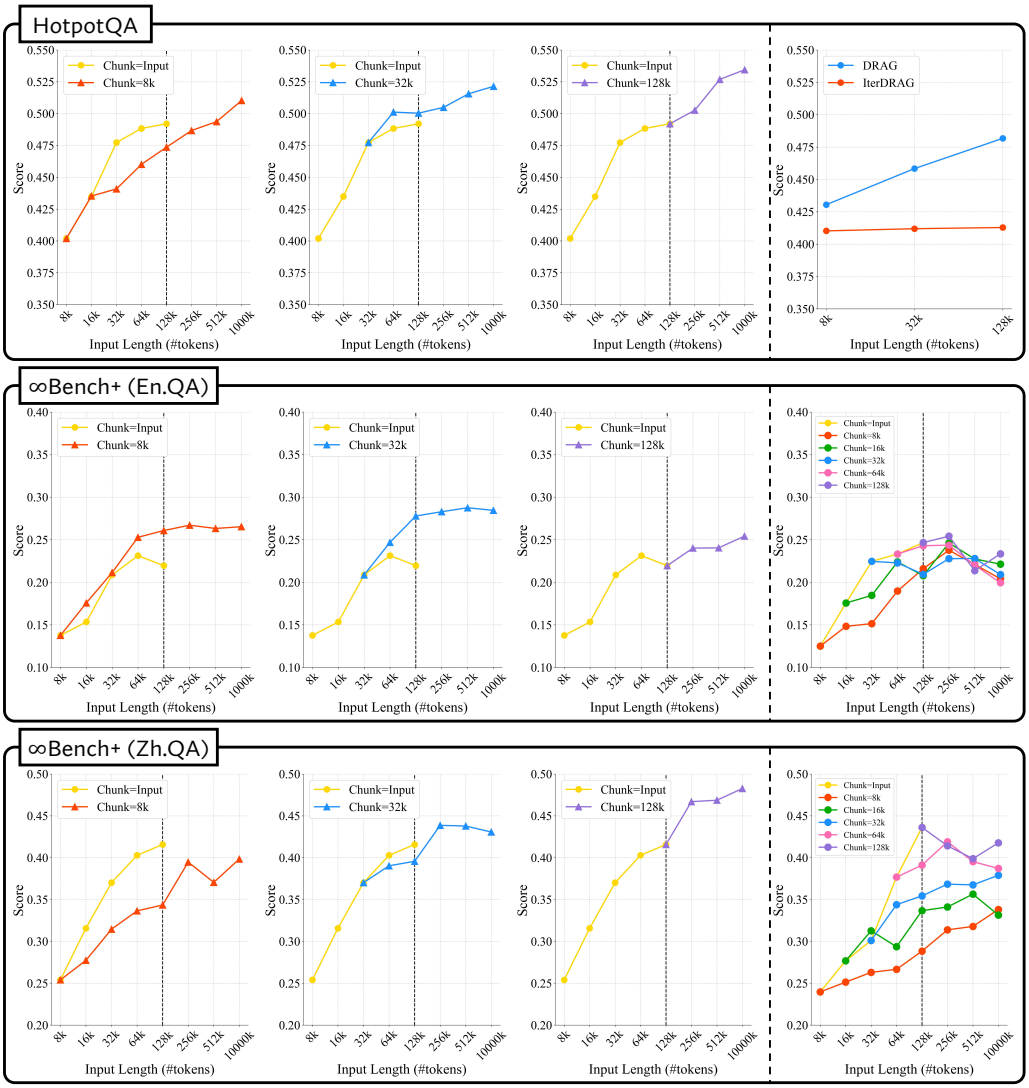

Figure 4: Experiment of scaling external knowledge input on multi-hop QA tasks. We plot Llama-3.1-8B-Instruct results for En.QA subset in ∞Bench+ and gpt-4o-mini results for other tasks. The *rightmost* subfigures are baseline results, including DRAG and IterDRAG on HotpotQA and LLM×MapReduce on ∞Bench+. "Chunk=Input" represents that the entire context truncated is fed directly into the LLM without any chunking according to the input length. Conversely, the "Chunk=N" labels indicate that the input sequence was segmented into fixed-size chunks of length $N \in \{8\text{k}, 16\text{k}, \dots\}$ tokens.

Seeking Agents at the intermediate synchronization timestep $t^\star$. The accumulated context is defined as:

$$\mathcal{M}_r^{(s)} = \text{Top}_{2^s}\left(\{m_{i,t^\star}\}_{i=1}^{N}\right). \tag{9}$$

The reasoning process is performed under Equation (5), but Reasoning Agent first checks the answerability of the query based on given information $\mathcal{M}_r^{(s)}$ and will only output the answer if the query is answerable, which then halts the whole process of ExtAgents. This iterative reasoning ensures that the Reasoning Agent progressively benefits from increased context without being overwhelmed. For QA tasks, the whole process is terminated when the answer is produced; or the maximum number of iterations S is reached without an answer, which then starts a new round of knowledge synchronization. For long survey generation, we adopt the drafting method (Wang et al., 2024d), where the outline is generated first, and then the reasoning process is performed to fill in each section. In this case, after filling up a section, the newly started process will take the previous section into the task query for continuous generation.

The separation of Seeking and Reasoning Agents in ExtAgents inherently enables parallelism. When using a separate tool for rating messages, each iteration of the reasoning process could independently select synchronized messages, which can also exploit parallel computation. For instance, the top $(2^{s-1}+1) \sim 2^s$ messages can be synchronized in parallel to the reasoning process on the top 2^{s-1} messages, forming an interleaved asyn-

Benchmark	LLM-as-a-Judge ($1 \sim 10$)	#Citations	Citation Density	Duplication Rate
AutoSurvey	6.75	113	1.00	2.41
ExtAgents (Ours)	**7.63**	**191**	**1.09**	**1.80**

Table 5: Experimental results on long survey generation tasks with gpt-4o-mini.

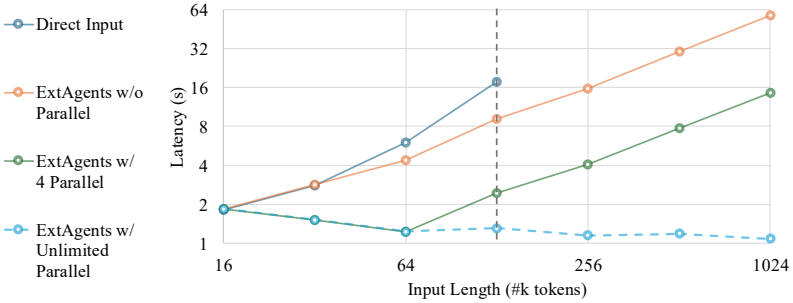

Figure 6: Latency analysis of ExtAgents with the chunk size of 16k tokens.

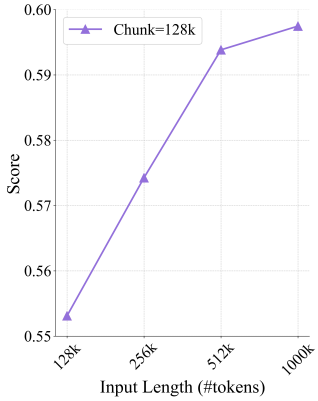

Figure 7: Results of ExtAgents with gpt-4o-2024-08-06 on HotpotQA benchmark.

chronous pipeline. Consequently, ExtAgents maintains high parallelism with scalability.

5 Experiments

5.1 Settings

Benchmarks (i) **∞Bench+**, our enhanced multi-hop QA benchmark with bi-lingual long documents attached to each query (Section 3.3), featuring Zh.QA and En.QA subsets, (ii) **HotpotQA** (Yang et al., 2018), containing open-domain multi-hop queries related to Wikipedia, and (iii) **AutoSurvey** (Wang et al., 2024d), generating long surveys with pre-retrieved papers, as a real-world application experiment. Metrics include F1 for multi-hop QA and LLM-as-a-Judge for generated surveys.

Methods On multi-hop QA tasks, we compare ExtAgents with (i) **Direct Input**, the baseline method that directly inputs the truncated context into LLMs, (ii) **LLM$\times$MapReduce** (Zhou et al., 2025), the state-of-the-art multi-agent method for long-context tasks, (iii) **Chain of Agents** (Zhang et al., 2024b), processing information in a linear topology where the bandwidth is 2, (iv) **DRAG** and (v) **IterDRAG** (Yue et al., 2025b), inference-time scalable retrieval methods for multi-hop QA with external knowledge bases. Notably, "Direct Input" is different from multi-agent methods without chunking, i.e., chunk size equals the input length ("Chunk=Input"), because of prompt templates and workflow designs. In the long survey generation application experiment, we compare ExtAgents with **AutoSurvey** (Wang et al., 2024d), only substituting the generation process in the pipeline for fair comparison. Concurrent works (Wang et al., 2025; Yan et al., 2025) propose survey generation methods with task-specific techniques, including skeleton evolving, heuristic generation, etc., which we decide is not directly comparable to our method. Details are in Appendix C.

5.2 Results and Analysis

Performance on Multi-Hop QA We plot the experimental results of scaling external knowledge input on multi-hop QA tasks in Figure 4. Detailed illustration is in Appendix C. The increasing trend of performance w.r.t. the input length indicates the scalability of each context window extension method. Empirically, ExtAgents consistently outperforms the baselines across all input lengths, achieving the significantly better performance on both HotpotQA and ∞Bench+ benchmarks. However, optimal chunk sizes are subjective to different tasks and require tuning. Moreover, the performance consistently improves with the increase of external knowledge input, demonstrating the scalability of ExtAgents. We also summarize the performance and input length of each method within its optimal setting in Table 4. ExtAgents achieves the best performance on all three multi-hop QA benchmarks by effectively utilizing more external knowledge compared to other methods. As another paradigm of inference-time scaling, long reasoning chains do not benefit as much (see DeepSeek-R1-Distill-Llama-8B line).

Performance on Long Survey Generation We test ExtAgents on long survey generation as a real-world application. ExtAgents could incorporate more related papers, and achieves better performance with more citations and lower duplication rate compared to AutoSurvey (Table 5). By aggregating eight pairwise scores from LLM-as-a-

Model Configuration	F1	Average Time per Sample
Llama-3.1-8B (Both Roles)	.383	115.87
Llama-3.2-3B (Both Roles)	.328	36.06
Llama-3.2-3B (Seeking) + Llama-3.1-8B (Reasoning)	.363	36.52

Table 6: Comparison of performance and inference time (s) across homogeneous and heterogeneous model configurations in different agent roles of EXTAGENTS on HotpotQA. Input length and chunk size are fixed to 512k and 32k, respectively. Time is measured without parallelization on 4 A100 GPUs.

Judge, we find that EXTAGENTS achieves a higher quality score with a significant margin. However, the evaluation of long surveys is challenging even for human experts. Thus, we also include a part of generated texts in Appendix B for qualitative comparisons besides designed metrics. Overall, it indicates that EXTAGENTS helps generate long surveys with higher quality and lower redundancy.

Latency and Cost Analysis In HotpotQA benchmark, we measure the latency of EXTAGENTS with fixed 16k-token chunks under different amounts of input with Llama-3.1-8B-Instruct on 4 A100 GPUs (Figure 6). The latency of direct input grows quadratically with input length, while EXTAGENTS maintains a linear growth for a fixed number of parallel threads. The theoretical complexity analyses are similar to Chain of Agents (Zhang et al., 2024b). Under unlimited parallel threads, ideally, EXTAGENTS could even decrease the latency because of fully paralleled Seeking Agents and the interleaved asynchronous pipeline (Section 4).

Ablation Studies We conduct ablation studies on the global knowledge synchronization (GKS) and knowledge-accumulating reasoning (KAR) in EXTAGENTS to analyze the identified bottlenecks (Figure 5). The results show that removing KAR leads to a significant drop in performance, especially as the amount of external knowledge increases. This demonstrates that the KAR effectively breaks the bottleneck of information overload. Removing GKS also leads to a drop in different input lengths, demonstrating that leveraging the range of exchanged information (i.e. "bandwidth") could help catch more relevant information.

Compatibility across LLM Families We further test EXTAGENTS with gpt-4o, and find that the performance is more significantly improved with the stronger LLM (Figure 7) compared to weaker models (Figure 4). Moreover, Llama-3.1-8B-Instruct suffers linguistic bias to achieve better performance on En.QA than Zh.QA, while gpt-4o-mini performs consistently well on both subsets (Table 4), affecting Llama's scalability on QA tasks in Chinese and with code-mixed knowledge bases (Figure 10). Further analyses are in Appendix C. Overall, **stronger LLMs benefit more from the scalability of EXTAGENTS**, implying promising future work with even stronger LLMs.

Heterogeneous Model Collaboration for Enhanced Efficiency A primary advantage of the decoupled agent profiles in ExtAgents is the flexibility to assign heterogeneous models to different roles based on their computational requirements. To explore the efficiency and performance trade-offs, we investigate whether a smaller, highly efficient model can serve as the Seeking Agent while a stronger model handles the more complex reasoning tasks. We conducted experiments pairing Llama-3.2-3B-Instruct for Seeking Agents with Llama-3.1-8B-Instruct for the Reasoning Agent on HotpotQA benchmark. Inference time was measured without parallelism on 4 A100 GPUs. For the heterogeneous setting, the two LLMs are deployed on 2 GPUs, respectively. As shown in Table 6, offloading the extensive synchronization and extraction workloads to the smaller model improves efficiency, achieving a ~3.1× speedup compared to using the 8B model for both roles, matching the computational efficiency of the pure 3B setup while delivering superior task performance.

6 Conclusion and Future Work

We introduce **EXTAGENTS**, a multi-agent framework that scales external knowledge input beyond context windows of LLMs without additional training. By decoupling knowledge synchronization and reasoning, EXTAGENTS overcomes core bottlenecks of existing methods, consistently improving multi-hop QA and long-form generation performance while maintaining high parallel efficiency. Future work will explore adaptive agent orchestration, cross-modal and tool augmentation, and fine-tuning specialized models.

Limitations

Model Alignment EXTAGENTS inherits both the strengths and weaknesses of its underlying LLMs: while substantial scalability on external knowledge input has shown in EXTAGENTS, the framework offers no principled defense on adversarial models, e.g., the aggregated evidence might be factually incorrect, biased, or policy-incompliant. Misaligned or adversarial Seeking Agents can propagate errors to every Reasoning Agent, amplifying harmful content or systemic biases. Incorporating alignment-aware scoring, preference-based post-training, or tool-based content filters could alleviate the problem but introduce training costs, which is left to future work.

Integration with Chunking Techniques EXTAGENTS partitions long inputs into fixed-size slices to simplify agent contexts, but makes no attempt to optimize those boundaries. Advanced chunking strategies—semantic segmentation, overlap windows, or hierarchical compression—could further reduce information loss (Chen et al., 2024a; Duarte et al., 2024; Zhao et al., 2024a), yet also introduce new coordination overhead and hyper-parameter choices. A systematic study of how adaptive chunking interacts with agent synchronization and reasoning quality is beyond the scope of this work.

Perfect Rankings with Further Calibration Though with current ranking design, the performance advancement has been empirically shown, the rankings could be unreliable in some extreme situations, e.g., out-of-domain or adversarial inputs to the rating method, rating biases in weaker LLMs or retrievers, etc. And, we indeed observe stronger LLMs could achieve better end-to-end performance in Section 5.2, potentially partially due to its stronger judging capability. We suggest further calibration methods to enhance the rating reliability in future work: (1) Calibration with labelled validation sets for fitting retriever scores, in-context learning in LLMs, etc.; (2) Calibration with multiple votes (Lin et al., 2024); (3) Rating with stronger or specifically trained models.

Ethical Considerations

EXTAGENTS should be viewed as a step towards more scalable and efficient knowledge-centric LLM workflow. Thus, different domains may be impacted either positively or negatively depending on the specific use case. For instance, in the educational domain, our work could enhance personalized learning and research productivity by allowing teachers and students to access and reason over extensive knowledge beyond textbooks, potentially democratizing expert-level insights and reducing barriers to advanced inquiry. However, there might also be negative implications: the improved scalability in integrating large-scale external knowledge may unintentionally amplify misinformation or biased viewpoints, as automated retrieval and reasoning processes could propagate inaccuracies present in the underlying data sources, especially for medical or economical industries. To mitigate these risks, further development of verification mechanisms and post-training techniques to align agent-produced knowledge is recommended.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62276152, 62236011), Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing, China. We thank Shuo Wang and Shengding Hu for the discussion during the early development of this project.

References

Anthropic. 2025. System card: Claude 4.5 sonnet. *Anthropic Technical Report*.

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)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023a. Walking down the memory maze: Beyond context limit through interactive reading. *Computing Research Repository*, arXiv:2310.05029.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023b. Extending context window of large language models via positional interpolation. *Computing Research Repository*, arXiv:2306.15595.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024a. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the*

2024 Conference on Empirical Methods in Natural Language Processing, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Yuxuan Chen, Dewen Guo, Sen Mei, Xinze Li, Hao Chen, Yishan Li, Yixuan Wang, Chaoyue Tang, Ruobing Wang, Dingjun Wu, Yukun Yan, Zhenghao Liu, Shi Yu, Zhiyuan Liu, and Maosong Sun. 2025. Ultrarag: A modular and automated toolkit for adaptive retrieval-augmented generation. *Computing Research Repository*, arXiv:2504.08761.

DeepSeek-AI. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(7952):633–638.

DeepSeek-AI. 2025b. Deepseek-v3 technical report. *Computing Research Repository*, arXiv:2412.19437.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11733–11763. PMLR.

André V. Duarte, João DS Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. 2024. LumberChunker: Long-form narrative document segmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6473–6486, Miami, Florida, USA. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Computing Research Repository*, arXiv:2404.16130.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Computing Research Repository*, arXiv:2312.10997.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-infinite: Zero-shot extreme length generalization for large 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 3991–4008, Mexico City, Mexico. Association for Computational Linguistics.

Jitai Hao, Yuke Zhu, Tian Wang, Jun Yu, Xin Xin, Bo Zheng, Zhaochun Ren, and Sheng Guo. 2025. OmniKV: Dynamic context selection for efficient long-context LLMs. In *The Thirteenth International Conference on Learning Representations*.

Zhitao He, Zijun Liu, Peng Li, Yi R. Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. 2025. Advancing language multi-agent learning with credit re-assignment for interactive environment generalization. In *Second Conference on Language Modeling*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

ByteDance Inc. 2025. Deerflow: A high-performance flow-based diffusion model. `https://github.com/bytedance/deer-flow`. Accessed: 2025-05-15.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Computing Research Repository*, arXiv:2503.09516.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng.

2024. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12758–12786, Miami, Florida, USA. Association for Computational Linguistics.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *Computing Research Repository*, arXiv:2501.05366.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. 2025. Marft: Multi-agent reinforcement fine-tuning. *Computing Research Repository*, arXiv:2504.16129.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling. *Computing Research Repository*, arXiv:2504.02495.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2024. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*.

OpenAI. 2024. Gpt-4o system card. *Computing Research Repository*, arXiv:2410.21276.

OpenAI. 2025a. Deep research system card. *OpenAI Technical Report*.

OpenAI. 2025b. Gpt-5 system card. *OpenAI Technical Report*.

OpenAI. 2025c. Openai gpt-4.5 system card. *OpenAI Technical Report*.

OpenAI. 2025d. Openai o3 and o4-mini system card. *OpenAI Technical Report*.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Computing Research Repository*, arXiv:2310.08560.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Yujia Zhou, Xu Chen, and Zhicheng Dou. 2024. Are long-llms a necessity for long-context tasks? *Computing Research Repository*, arXiv:2405.15318.

Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Jiang, Chengfei Lv, and Huajun Chen. 2024. AutoAct: Automatic agent learning from scratch for QA via self-planning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3003–3021, Bangkok, Thailand. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333389.

Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. 2025. LongroPE2: Near-lossless LLM context window scaling. In *Forty-second International Conference on Machine Learning*.

Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, and Pang Wei Koh. 2024. Scaling retrieval-based language models with a trillion-token datastore. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *Computing Research Repository*, arXiv:2503.05592.

Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. 2025. Multiagent finetuning: Self improvement with diverse reasoning chains. In *The Thirteenth International Conference on Learning Representations*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 60006010, Red Hook, NY, USA. Curran Associates Inc.

Haoyu Wang, Yujia Fu, Zhu Zhang, Shuo Wang, Zirui Ren, Xiaorong Wang, Zhili Li, Chaoqun He, Bo An, Zhiyuan Liu, and Maosong Sun. 2025. Llm×mapreduce-v2: Entropy-driven convolutional test-time scaling for generating long-form articles from extremely long resources. *Computing Research Repository*, arXiv:2504.05732.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao

Sang. 2024a. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shengnan Wang, Youhui Bai, Lin Zhang, Pingyi Zhou, Shixiong Zhao, Gong Zhang, Sen Wang, Renhai Chen, Hua Xu, and Hongwei Sun. 2024b. Xl3m: A training-free framework for llm length extension based on segment-wise inference. *Computing Research Repository*, arXiv:2405.17755.

Yan Wang, Dongyang Ma, and Deng Cai. 2024c. With greater text comes greater necessity: Inference-time training helps long text generation. In *First Conference on Language Modeling*.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024d. Autosurvey: Large language models can automatically write surveys. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *Computing Research Repository*, arXiv:2504.12516.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024a. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Chejian Xu, Wei Ping, Peng Xu, Zihan Liu, Boxin Wang, Mohammad Shoeybi, Bo Li, and Bryan Catanzaro. 2025. From 128k to 4m: Efficient training of ultra-long context large language models. *Computing Research Repository*, arXiv:2504.06214.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *Computing Research Repository*, arXiv:2504.08066.

Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Lei Bai, and Bo Zhang. 2025. SURVEYFORGE : On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12444–12465, Vienna, Austria. Association for Computational Linguistics.

Zeyuan Yang, Fangzhou Xiong, Peng Li, and Yang Liu. 2025. Rethinking long context generation from the continual learning perspective. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1922–1933, Abu Dhabi, UAE. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. 2025a. MasRouter: Learning to route LLMs for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15549–15572, Vienna, Austria. Association for Computational Linguistics.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025b. Inference scaling for long-context retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. 2025. G-designer: Architecting multi-agent communication topologies via graph neural networks. In *Forty-second International Conference on Machine Learning*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024a. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan O Arik. 2024b. Chain of agents: Large language models collaborating on long-context tasks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jihao Zhao, Zhiyuan Ji, Yuchen Feng, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024a. Meta-chunking: Learning efficient text segmentation via logical perception. *Computing Research Repository*, arXiv:2410.12788.

Jun Zhao, Can Zu, Xu Hao, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024b. LONGAGENT: Achieving question answering for 128k-token-long documents through multi-agent collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16310–16324, Miami, Florida, USA. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024c. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zihan Zhou, Chong Li, Xinyi Chen, Shuo Wang, Yu Chao, Zhili Li, Haoyu Wang, Qi Shi, Zhixing Tan, Xu Han, Xiaodong Shi, Zhiyuan Liu, and Maosong Sun. 2025. LLM×MapReduce: Simplified long-sequence processing using large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27664–27678, Vienna, Austria. Association for Computational Linguistics.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62743–62767. PMLR.

A Details of Review on Existing Multi-Agent Methods

In this section, we provide the implementation of existing LLM-based multi-agent systems for context window extension, including Chain of Agents (Zhang et al., 2024b), LongAgent (Zhao et al., 2024b), and LLM×MapReduce (Zhou et al., 2025) in our framework in Section 3.2. We also explain the comparison results in Table 1.

Knowledge Synchronization Here, we explain how existing methods synchronize knowledge across agents according to Equation (4). For **Chain of Agents**, each agent a_i, t incorporate the message from previous agent $a_{i-1,t-1}$ in a linear topology, and the message is passed to the next agent $a_{i+1,t-1}$ in the next timestep. So in Equation (4), $\mathcal{D}^{(\text{CoA})}_{\mathcal{G}_{i,t-1},t} = \{d_i\}$ and $\mathcal{M}^{(\text{CoA})}_{\mathcal{G}_{i,t-1},t-1} = \{m_{i-1,t-1}\}$. Since $\mathcal{G}^{(\text{CoA})}_{i,t-1} = \{a_{i-1,t-1}, a_{i,t-1}\}$, the bandwidth is 2. It is also clear that the process could not be parallelized. For **LongAgent**, the leader agent identify conflicts between two agents $a_{i,t-1}$ and $a_{j,t-1}, (j \neq i)$, and the individual message and original chunk are passed to each other. So in Equation (4), $\mathcal{D}^{(\text{LA})}_{\mathcal{G}_{i,t-1},t} = \{d_i, d_j\}$ and $\mathcal{M}^{(\text{LA})}_{\mathcal{G}_{i,t-1},t-1} = \{m_{i,t-1}, m_{j,t-1}\}$. Thus, $\mathcal{G}^{(\text{LA})}_{i,t-1} = \{a_{i,t-1}, a_{j,t-1}\}$, and the bandwidth is 2. For each timestep, the two agents could function in parallel. Though, the total timesteps T is $O(\frac{L}{|m|})$, which is the number of synchronization decisions made by the leader agent. However, due to the capacity of the leader agent, T is rather small and most agents barely access all other contexts. Specifically, the task query q might be changed to generated sub-queries by the leader agent during the process. For **LLM×MapReduce**, agents function in parallel and aggregate their messages in groups with adjacent agents. The group size is $O(\frac{L}{|m|})$ for agent to process messages within the context window at the next timestep. And $\mathcal{M}^{(\text{MR})}_{\mathcal{G}_{i,t-1},t-1} = \phi$. Thus, the bandwidth is $O(\frac{L}{|m|})$. Each group could be processed in parallel at the same timestep.

Knowledge-Integrated Reasoning Here, we explain how existing methods integrate knowledge in reasoning according to Equation (5). For **Chain of Agents**, the reasoning agent only takes the message from the last agent $a_{N,T}$ as the reasoning context $\mathcal{M}^{(\text{CoA})}_r = \{m_{N,T}\}, (T = N)$. For **LongAgent**, the leader agent takes all messages from all agents in sequential as the reasoning context $\mathcal{M}^{(\text{LA})}_r = \{m_{i,t}\}_{1\leq i\leq N, 1\leq t\leq T}$. Since $T = O(\frac{L}{|m|})$, the context would not exceed the context window. For **LLM×MapReduce**, the reduce agent takes the last messages from all agents as the reasoning context $\mathcal{M}^{(\text{MR})}_r = \{m_{i,T}\}_{1\leq i\leq N}$. The method guarantees that the reasoning context is within the context window by recursively aggregating the messages in groups with adjacent agents. The last two methods tend to include as much information as possible in the reasoning context, which may lead to overload with redundant, noisy information.

B Qualitative Analysis

For multi-hop QA tasks, we show three cases as follows. The first case demonstrates EXTAGENTS' superior ability to connect disparate pieces of information across multiple documents to answer a complex, multi-step question accurately, where other methods fail to synthesize the necessary facts or get sidetracked by redundant details. The second case highlights EXTAGENTS' effectiveness in pinpointing the correct answer by intelligently scoring and prioritizing the most relevant textual evidence, especially when multiple, potentially conflicting pieces of information are present within the source material. The third case shows EXTAGENTS' strength in accurately identifying the main subject by discerning the most pertinent information through its scoring mechanism, successfully avoiding misdirection from other related but less central entities mentioned in the text in different languages.

For long survey generation, we present a comparison of one subsection generated by our method and AutoSurvey for the same topic. The survey generated by EXTAGENTS presents a more related and more diverse range of innovative LLM applications in education, making the overall argument more persuasive and resulting in overall 68 pages compared to 58 pages from AutoSurvey.

B.1 Case: Multi-Hop Question Answering

Example from HotpotQA: Question Answering on William Hodges' Voyage

Question: William Hodges visited a polar region as part of a voyage, the region is how big?

Answer: 14 million km

Excerpts from retrieved documents:

William Hodges RA (28 October 1744 6 March 1797) was an English painter. He was a member of James Cook's second voyage to the Pacific Ocean, and is best known for the sketches and paintings of locations he visited on that voyage, including Table Bay, Tahiti, Easter Island, New Zealand, Dusky Sound and the Antarctic.

The Antarctic (US English, UK English or and or) is a polar region around the Earth's South Pole, opposite the Arctic region around the North Pole. The Antarctic comprises the continent of Antarctica, the Kerguelen Plateau and other island territories located on the Antarctic Plate or south of the Antarctic Convergence. The Antarctic region includes the ice shelves, waters, and all the island territories in the Southern Ocean situated south of the Antarctic Convergence, a zone approximately wide varying in latitude seasonally. The region covers some 20 percent of the Southern Hemisphere, of which 5.5 percent (14 million km) is the surface area of the Antarctic continent itself.

EXTAGENTS

Extracted information (partial):

William Hodges, an English painter, accompanied James Cook on his second voyage to the Pacific Ocean between 1772 and 1775. During this voyage, he traveled to various locations, including Antarctic regions. However, the chunk does not provide the exact size of the polar region Hodges visited. It mentions that Hodges is known for his sketches and paintings of locations he visited on that voyage, including those in the Antarctic, but does not specify the size of the region.

William Hodges was likely referring to the polar regions explored during his voyages. Specifically, he could relate to the Antarctic region, which is noted as a large landmass and describes being covered by about 98% ice, averaging in thickness. The Antarctic continent itself encompasses an area of approximately 14 million square kilometers (or around 5.5% of the Southern Hemisphere).

Prediction: Approximately 14 million square kilometers.

DRAG prediction: The region is thought to have covered about 1.5 million square kilometers.

IterDRAG prediction: The specific size of the broader region William Hodges visited is not mentioned in the context; however, one specific area he explored, Hodges Knoll, is 2.2 km in northeast-southwest direction and 1 km wide.

Example from En.QA in ∞Bench+: Question Answering on Davion's Fortune

Question: Where is the Davion' s fortune hidden?

Answer: On the island of Brayan Annabel

Excerpts from original text:

Davion followed him with an excited look. "and now," he said, when he saw that Pascal had read the last line, "put the two fragments together, and judge for yourself." Pascal obeyed, and the conjointed pieces gave the following: – "This 25th day of April, 1498, be...ing invited to dine by his Holiness Alexander VI., and fearing that not...content with making me pay for my hat, he may desire to become my heir, and re...serves for me the fate of Cardinals Caprara and Bentivoglio, who were poisoned...I declare to my nephew, Guido Spada, my sole heir, that I have bu...ried in a place he knows and has visited with me, that is, in...the caves of the small Island of Brayan Annabel all I poss...ssed of ingots, gold, money, jewels, diamonds, gems; that I alone...know of the existence of this treasure, which may amount to nearly two mil...lions of Roman crowns, and which he will find on raising the twentieth ro...ck from the small creek to the east in a right line. Two open...ings have been made in these caves; the treasure is in the furthest a...ngle in the second; which treasure I bequeath and leave en...tire to him as my sole heir."

ExtAgents

Extracted information and corresponding scores (partial):

Davion's fortune is hidden in a coffer located in a cave on the Island of Brayan Annabel. The coffer is described as being bound with iron and engraved with the arms of the Spada family. It contains piles of golden coins, bars of unpolished gold, and a significant amount of diamonds, pearls, and other gems. The treasure is concealed beneath a large circular rock that Pascal discovers and manages to move, revealing the coffer underneath. Score: 100.0

Prediction: Island of Brayan Annabel

LLM×MapReduce

Extracted information (partial):

The Davion's fortune is hidden in a treasure chest buried in the caves on the Island of Brayan Annabel.

The Davion's fortune is hidden in a diamond that belonged to a character named Armando Pascal, who bequeathed it to Corbin, the innkeeper, before his death.

Davion's fortune is hidden in a grotto at Brayan Annabel, of which Brisa knows the secret.

The Davion's fortune is hidden in a chest concealed under the closet in the little house in the Allees de Meillan, at Marseilles.

Prediction: The Davion's fortune is hidden in multiple locations: a treasure chest in the caves on the Island of Brayan Annabel, a diamond bequeathed to Corbin, a grotto at Brayan Annabel, and a chest under a closet in a house in Allees de Meillan, Marseilles.

Example from Zh.QA in ∞Bench+: Question Answering on Who Infiltrated the Scarlet Research Institute

Question: 谁混入猩红研究院开启他的卧底计划

Answer: 贾易

Excerpts from original text:

“我决定让我的学生贾易开始接触生物科学领域，希望他不会让我失望，以他的学习能力，如果努力的话，不需要太久就能精通这方面的内容，并混入猩红研究院。”
“贾易已经通过了猩红研究院的两轮考核，即将混入猩红研究院开启他的卧底计划，希望这次能从中得到有用的信息。”
“贾易已经混入猩红研究院了，我等待着他传来的好消息。”

EXTAGENTS

Extracted information and corresponding scores (partial):

在故事中，混入猩红研究院开启卧底计划的角色是贾易。他是封棋的学生，负责保护可能被暗杀的人类天才，并试图从猩红研究院获取有用的信息。贾易的任务是潜入猩红研究院，调查其内部情况，尤其是与符文改造人计划相关的秘密。Score: 100.0

在文章中，混入猩红研究院并开启卧底计划的是“黑影”。黑影是灵能族的叛徒，曾经在灵能族中拥有强大的实力。文章提到黑影在科技研究院布局未来，并且在战争线中与封棋等人产生了交集。封棋在与黑影的对抗中，逐渐意识到黑影的真实身份和目的。Score: 90.0

Prediction: 贾易

LLM×MapReduce

Extracted information (partial):

在小说中，混入猩红研究院开启卧底计划的是“贾易”。他是未来的封棋的学生之一，在封棋未进入未来梦境之前，已经通过了猩红研究院的考核，成为了组织的成员，负责保护被暗杀的人类天才。

根据文章内容，混入猩红研究院开启卧底计划的是“黑影”。在封棋与黑影的对话中，黑影透露自己来自一个由多个弱小族群组成的联盟，负责刺杀人类新生代天才等行动。同时，黑影也提到自己和他的族群并不像科技研究院或虎魄研究院那般强大，而是被迫在外打拼，只能通过暗杀等方式潜入人类社会，表明其立场和背景。

根据您提供的文章，混入猩红研究院并开启卧底计划的角色是“黑影”。黑影是掌控科技研究院的潜伏者，与黑玄派系的灵能族成员有关，暗中培养力量，目的是为了对抗族地灵能族。有关黑影的信息在文章的多处被提及，其作用和动机与整个情节的发展密切相关。

Prediction: 黑影

B.2 Case: Long Survey Generation

A Subsection in Long Survey Generated by EXTAGENTS

1.6 Case Studies Demonstrating LLM Impact on Education

The integration of Large Language Models (LLMs) into educational contexts has resulted in transformative changes, showcasing their potential to enhance teaching, learning, and administrative processes. Numerous case studies illustrate the successful implementation of LLMs in various educational settings, providing insights into their effectiveness and the tangible benefits they deliver.

One notable case study focused on using LLMs to simulate student learning behaviors, where researchers leveraged LLMs to create virtual student models that replicate real learner patterns based on demographic data. This experiment involved 145 participants and revealed that the simulated results aligned closely with the actual students' learning behaviors across diverse demographics. This application demonstrates how LLMs can enhance inclusivity in curriculum design by providing insights into how different student characteristics influence learning outcomes [40].

In the realm of personalized learning, case studies have highlighted the development of LLM-driven intelligent tutoring systems. These systems adapt to individual student needs, providing real-time support tailored to specific learning styles and paces. In one study, LLMs were implemented as personalized tutors in mathematics, demonstrating that such systems could significantly improve comprehension and engagement among learners. The ability to receive tailored guidance directly addresses students' challenges, ultimately enhancing educational efficacy [5].

Automated grading systems powered by LLMs have also gained traction in educational institutions. These systems evaluate students' assignments consistently and objectively, thereby saving educators time and minimizing biases often associated with manual grading processes. In practical studies, LLMs have shown reliable scoring that correlates well with human assessment, allowing educators to focus more on quality instruction rather than administrative tasks [41].

Interactive tools facilitated by LLMs, such as chatbots, have revolutionized student engagement models. A significant case involved an LLM-based tool providing on-demand programming assistance to students in an introductory computer science course. Over a period of 12 weeks, the tool managed more than 2,500 queries, primarily related to immediate help with assignments. The findings suggested that students who engaged frequently with the tool had higher success rates in the course, indicating that LLM-powered assistance can significantly enhance the learning experience in large classroom settings [42].

Furthermore, LLMs have been effectively employed to enhance language acquisition among learners. One study examined how situational dialogue models fine-tuned on LLMs facilitate conversational practice for language students. The models allowed for rich, simulated dialogues that mirrored authentic conversations, leading to significant improvements in fluency and confidence. Participants who engaged with these LLM-driven dialogues exhibited enhanced communication skills compared to those who did not, showcasing the crucial role of LLMs in supporting language learning [43].

In the context of automated content generation, LLMs have proven capable of producing educational resources quickly and effectively. A noteworthy implementation is the development of AutoPlan, which leverages LLMs to create interactive decision-making tasks. This intelligent system significantly increases the efficiency of educational environments centered on decision-making and planning, illustrating how LLMs can enhance problem-solving instruction by breaking down complex tasks into manageable components [44].

Moreover, LLMs have demonstrated their capability to contribute to informal learning situations

through chat-based environments. Researchers have explored how learners utilize LLMs to seek answers outside traditional classroom boundaries, supporting knowledge acquisition and empowering students to take charge of their own learning. The results indicate that LLM interactions not only facilitate knowledge gain but also promote lifelong learning habits that are essential in a rapidly changing world [45].

Another compelling case involved utilizing LLMs to support health literacy in educational settings. A comprehensive assessment using various prompts with several LLMs, including ChatGPT and Google Bard, demonstrated their effectiveness in tailoring health-related content for different reading levels. The findings revealed that LLMs could produce responses at appropriate reading levels for younger audiences, thus enhancing health communication capabilities and aligning with the goal of promoting understanding among diverse learner populations [46].

A unique implementation highlights how LLMs can serve as reading companions for students. Through a study involving a user-centric assistant named Claude, researchers found tangible improvements in reading comprehension and engagement among students utilizing the AI tool compared to traditional study methods. This innovative use of LLMs showcases their potential as supplemental educational resources, helping learners tackle complex literature and academic texts effectively while fostering enhanced interaction with reading materials [47].

Moreover, the case of employing LLMs in hybrid workplace decision support exemplifies their adaptability and decision-making capabilities. In educational contexts, these systems guide students in creating effective study plans and managing their time better. By analyzing users' decision-making processes, LLMs assist learners in balancing various factors, thus enhancing their academic experience [48].

Finally, research has begun to explore the ethical implications and potential biases associated with LLM implementations in education. One study examined how LLMs were evaluated for ethical considerations in real-world educational practices, underscoring the importance of transparent LLM development processes. The findings emphasized that ethical reflection and intervention could significantly enhance educational outcomes while addressing biases, highlighting the dual need for technological innovation and ethical responsibility when deploying LLMs in educational settings [49].

These case studies collectively illustrate the transformative impact of LLMs in various educational contexts. They highlight the models' dual role as supportive tools for educators and personalized learning facilitators for students. From dialogic interactions and intelligent tutoring systems to automated content generation and tailored health communication, the versatility of LLM applications is evident. However, they also reveal the complexities and challenges of integrating AI into educational systems, necessitating a balanced approach that prioritizes ethical considerations, pedagogical integrity, and student engagement.

As the landscape of education continues to evolve, the integration of LLMs holds immense potential for enriching the learning experience. Continuous research and iterative refinements will be essential to harnessing the full capabilities of LLMs, ensuring they complement traditional educational practices while addressing the diverse needs of learners. The future of education may present unprecedented opportunities through the effective deployment of these advanced AI systems, paving the way for personalized, engaging, and impactful learning experiences that prepare students for the challenges of the 21st century.

A Subsection in the Long Survey Generated by AutoSurvey

1.6 Case Studies Demonstrating LLM Impact on Education

Large Language Models (LLMs) have taken center stage in educational innovation, demonstrating significant potential to enhance teaching and learning experiences across various contexts. This subsection presents a variety of case studies that illustrate successful deployments of LLM technologies in educational settings, showcasing their effectiveness and the tangible benefits achieved in real-world scenarios.

One notable application of LLMs is highlighted in the study titled "Future-proofing Education: A Prototype for Simulating Oral Examinations Using Large Language Models." This research explored a prototype system designed to simulate oral examinations in higher education. Educators and students evaluated this system, which aimed to automate and enhance the examination process. The outcomes showed that the prototype provided personalized feedback to students while significantly reducing the workload on educators, demonstrating the LLM's capability to streamline assessment processes and improve educational efficiency [27].

Another innovative implementation involved using LLMs to simulate student learning behavior. In the paper "Leveraging generative artificial intelligence to simulate student learning behavior," researchers revealed how they utilized LLMs to craft virtual students with distinct demographics. The findings from three experiments showcased the ability of LLMs to replicate intricate learning behaviors and experiences, unveiling correlations between course materials, engagement levels, and understanding. This simulation approach empowers educators to design curricula that adapt dynamically to diverse student needs, thus enhancing inclusivity and educational effectiveness [29].

Additionally, the integration of LLMs within role-playing scenarios has proven effective in fostering engagement and active learning among students. The study titled "Role-Playing Simulation Games using ChatGPT" illustrated how incorporating ChatGPT into role-playing simulations could enhance the quality of teaching by providing realistic environments for learners to practice skills. This approach led to increased student interest and participation, exemplifying the potential of LLMs to create immersive and interactive learning experiences [30].

LLMs have also shown promise in automating the grading process, a crucial area of concern in educational assessments. In "Three Questions Concerning the Use of Large Language Models to Facilitate Mathematics Learning," the study discusses how LLMs can provide adaptive feedback on mathematical problem-solving tasks. By assessing students' answers, LLMs can identify misconceptions and offer tailored guidance, potentially leading to better learning outcomes. Such systems emphasize the transition toward more personalized assessment methods in education [31].

Moreover, the implementation of LLMs in content generation showcases their value in educational resource development. The research titled "Prototyping the use of Large Language Models (LLMs) for adult learning content creation at scale" examined how LLMs could assist in creating quality learning materials for adult education. By leveraging a human-in-the-loop approach, the study found that LLMs could produce high-quality content quickly, marking significant advancements in Generative AI's application for education. This capacity to automate content creation not only reduces the burden on educators but also ensures resource availability and accessibility [32].

In a collaborative learning context, LLMs have been utilized to foster peer interaction and engagement. The paper titled "The Use of Multiple Conversational Agent Interlocutors in Learning" explored how LLMs could simulate conversations among various personas in an educational environment. This method assists learners in problem-solving by exposing them to different viewpoints and areas of expertise, enriching the collaborative learning experience. Such interactive dynamics can enhance comprehension and engagement among students, urging creative exploration

of subject matter [33].

Implementing LLMs within existing educational frameworks also presents advantages for developing personalized learning experiences. The study "Adapting Large Language Models for Education: Foundational Capabilities, Potentials, and Challenges" provides insights into how LLMs can adapt to individual student requests, addressing diverse learning obstacles. By utilizing LLMs capable of interpreting and responding to students' needs, educational institutions can design more effective and customized learning modules, reinforcing the model's role as a facilitator in education [34].

Furthermore, the integration of LLMs into LMS (Learning Management Systems) has revealed additional benefits. The research "Identifying Critical LMS Features for Predicting At-risk Students" discussed how integrating LLMs into LMSs could analyze student interaction data to identify those at risk of underperforming. By using predictive analytics, educators can intervene proactively, thus promoting student success through personalized academic support [35].

Finally, ethical considerations, crucial when implementing LLMs in educational environments, have been addressed in the study "The teachers are confused as well: A Multiple-Stakeholder Ethics Discussion on Large Language Models in Computing Education." The authors conducted stakeholder interviews to explore the ethical challenges faced by both educators and students in using LLMs. Their findings underscore the importance of establishing guidelines and fostering digital literacy among students to ensure a balanced approach to utilizing LLMs responsibly in education [25].

Overall, these case studies collectively illustrate the transformative impact of LLMs in education. By automating assessments, enhancing content generation, enriching collaborative learning experiences, and facilitating personalized education, LLMs hold the potential to revolutionize pedagogical practices. As educational institutions continue to explore and refine LLM applications, efforts to address ethical concerns and optimizing their deployment will be critical to fully realizing their capabilities in fostering an effective and inclusive learning environment. The integration of LLMs can pave the way for innovative educational solutions and provide a solid foundation for future research and development in the field.

C Implementation Details

C.1 Experiment Details

All datasets and models are used under their individual licenses without intention to violate any terms.

For evaluation metrics, we found existing principles (Wang et al., 2024d, 2025) for LLM-based judges have very low discriminative power on generated surveys, and thus we use the prompting template from Liu et al. (2025) to judge the result against baselines. We also include the Helmet correctness score (Yen et al., 2025) and additional quantitative measurement (e.g., #citations, citation density, duplication rate, etc.) as supplementary metrics for QA and long generation tasks, respectively. We control the whole input length for each methods in the range of $\{8k, 16k, 32k, 64k, 128k, 256k, 512k, 1024k\}$ tokens with the maximum context window $128k$, and also control the chunk sizes of each agent for LLM-based multi-agent methods (Figure 4). Specifically, for a minority of samples with contexts longer than $1024k$, we input all contexts for the $1024k$ setting.

We utilized several large language models in our experiments. Closed-source models included gpt-4o-mini-2024-07-18 and gpt-4o-2024-08-06, accessed via API. For these models, the sampling temperature was set to 0. Open-source models employed were Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct. These models were deployed on four NVIDIA A100 80GB GPUs, and the sampling temperature was set to 0.1. The maximum input context length was 128,000 tokens for the closed-source models and 131,092 tokens for the open-source models. We test these methods with the optimal configuration of input lengths and chunk sizes (Table 4). For stable reproduction, we report the median results of three runs. For HotpotQA, we use BM25 retriever (Robertson and Zaragoza, 2009), and for AutoSurvey, we use the original retrieval method.

Baseline implementations are adjusted slightly for each task. For En.QA and Zh.QA test sets, the direct input method is implemented using the official InfiniteBench repository (Zhang et al., 2024a). The LLM×MapReduce is re-implemented by us to align with our settings with ExtAgents. For HotpotQA, we re-implement the DRAG and Iter-DRAG methods due to the lack of official code, utilizing the prompts provided in the appendix of Yue et al. (2025b). For the survey generation task, the AutoSurvey baseline was adapted from the official implementation (Wang et al., 2024d), with the reflection and refinement removed from the original pipeline. For the metrics, the citation density is calculated as the number of citations divided by the number of thousand tokens in the generated survey. The duplication rate is calculated as the number of duplicate citations divided by the total number of citations. The LLM-as-a-Judge method on long survey generation is implemented with gpt-4.1-mini-2025-04-14 (OpenAI, 2025c), with temperature set to 1.

For ExtAgents, in En.QA and Zh.QA tests, information extracted from different chunks is ranked based on scores rated by Seek Agents, based on the task query. For the HotpotQA test, the information is ranked based on the retrieval priority of chunks. The chunk exclusion mechanism is disabled in long-document QA tasks. For AutoSurvey task, we set the number of chunks of input retrieved papers to 4, and thus assigning 4 Seeking Agents to input papers. Other hyper-parameters are set to the default values. The overall process for our methods involves a maximum of $T = 5$ synchronization timesteps. We only employ the knowledge accumulation strategy at $t = 1$. This strategy processes information incrementally, starting with the top 1, then top 2, top 4, top 8, and finally all information ($S = 5$), following a power-of-two sequence. The whole process halts once the answer is obtained.

C.2 Additional Experimental Results

Helmet Correctness Scores on Multi-Hop QA We provide corresponding Helmet correctness scores (Yen et al., 2025) on multi-hop QA tasks complementary to Table 4 in Table 7. F1 scores may misjudge the performance of LLMs, especially when the response is long and the answer is not concise enough. In our experiments, we observe that the trend of Helmet correctness scores are consistent with the F1 scores, indicating that the performance of ExtAgents is robust and reliable.

Results on the Original ∞Bench We provide the result on the original ∞Bench in Table 8. We observe the same trend as in ∞Bench+ in Table 4, that ExtAgents achieves the highest performance with the longest input contexts. And other methods, including inference-time scaling with long CoT, fail to utilize longer contexts beyond the context window and reach inferior results.

Method	HotpotQA		En.QA		Zh.QA	
	Helmet	Input	Helmet	Input	Helmet	Input
DeepSeek-R1-Distill-Llama-8B						
Direct Input	1.56	32k	0.69	32k	0.66	32k
gpt-4o-mini-2024-07-18						
Direct Input	1.83	128k	1.41	128k	1.04	128k
DRAG	1.53	128k				
IterDRAG	1.70	128k				
LLM×MapReduce			1.12	256k	1.04	128k
ExtAgents (Ours)	**1.71**	**1024k**	**1.20**	**1024k**	**1.10**	**256k**
Llama-3.1-8B-Instruct						
Direct Input	0.96	128k	0.93	128k	0.89	128k
DRAG	1.20	32k				
IterDRAG	1.14	32k				
Chain of Agents			0.51	32k	0.57	32k
LLM×MapReduce			0.78	256k	0.79	256k
ExtAgents (Ours)	**1.38**	**1024k**	**1.09**	**1024k**	**0.85**	**256k**
gpt-4o-2024-08-06						
ExtAgents ($N = 1$)	1.73	128k				
ExtAgents	**1.86**	**1024k**				

Table 7: Performance on Multi-Hop QA tasks in Helmet correctness scores with the optimal setting and the corresponding input length (#tokens). The settings are the same as Table 4.

Method	En.QA		Zh.QA	
	F1	Input	F1	Input
DeepSeek-R1-Distill-Llama-8B				
Direct Input	.104	32k	.144	32k
gpt-4o-mini-2024-07-18				
Direct Input	.189	128k	.206	128k
LLM×MapReduce	.385	128k	.443	128k
ExtAgents (Ours)	**.421**	**1024k**	**.491**	**1024k**
Llama-3.1-8B-Instruct				
Direct Input	.267	128k	.316	128k
LLM×MapReduce	.287	256k	.347	128k
ExtAgents (Ours)	**.322**	**512k**	**.352**	**256k**

Table 8: Performance on ∞Bench with the optimal setting and the corresponding input length (#tokens). Other settings are the same as Table 4.

Detailed Results on Multi-Hop QA We provide the detailed results on multi-hop QA tasks in Figure 8 and Figure 10. The results are consistent with the main findings in Figure 4 with enriched results.

Results on Weaker & Stronger LLMs We also test the performance of ExtAgents on a weaker LLM, Llama-3.2-3B-Instruct, besides a stronger LLM, gpt-4o-2024-08-06, on HotpotQA benchmark. The results are shown in Figure 9 and Figure 7, respectively. We observe that ExtAgents achieves consistent performance improvements over scaled external knowledge input, and the performance gap is larger on the stronger model, potentially due to the better collaboration capability of stronger LLMs.

Method	HotpotQA	En.QA	Zh.QA
DRAG	0.019		
IterDRAG	0.048		
LLM×MapReduce		0.022	0.021
ExtAgents (Ours)	0.021	0.025	0.029

Table 9: Average costs ($) of ExtAgents and baseline methods on gpt-4o-mini-2024-07-18.

Detailed Costs Analysis We demonstrate the average costs of ExtAgents and baseline methods on gpt-4o-mini-2024-07-18 in Table 9. The costs are calculated based on the average number of tokens in the input and output, where the cost of 1M token input is $0.15 and 1M token output is $0.60. The extra cost of ExtAgents is due to the global knowledge synchronization, which introduces larger bandwidth and according costs.

Scaling External Knowledge Input with Chain of Agents We test the scaling performance of Chain of Agents under several configurations of input lengths and chunk sizes. Results are shown in Table 10. Alike Zhou et al. (2025), we also observes inferior scaling behaviors in Chain of Agents compared to LLM×MapReduce and ExtAgents, which also aligns to our theoretical analysis of the bottleneck of knowledge synchronization (Chain of Agents has a smaller bandwidth value of constant

Method	En.QA				Zh.QA			
	F1	Helmet	Input	Chunk	F1	Helmet	Input	Chunk
DeepSeek-R1-Distill-Llama-8B								
Direct Input	.097	0.69	32k	-	.143	0.66	32k	-
gpt-4o-mini-2024-07-18								
EXTAGENTS (Ours)	**.382**	**1.20**	**1024k**	128k	**.482**	**1.10**	**1024k**	128k
Llama-3.1-8B-Instruct								
Direct Input	.237	0.93	128k	-	.315	0.89	128k	-
	.168	0.51	32k	32k	.246	0.57	32k	32k
	.075	0.27	128k	32k	.123	0.32	128k	32k
Chain of Agents	.027	0.12	128k	128k	.070	0.21	128k	128k
	.073	0.27	1024k	32k	.158	0.34	1024k	32k
	.068	0.21	1024k	128k	.064	0.20	1024k	128k
LLM×MapReduce	.254	0.78	256k	128k	.345	0.79	128k	128k
EXTAGENTS (Ours)	**.291**	**1.09**	**1024k**	16k	**.347**	**0.85**	**256k**	128k

Table 10: Performance of Chain of Agents on ∞Bench+ in different configurations of chunk sizes and input lengths (#tokens). Results of other methods are picked from Table 4. F1 and Helmet correctness scores are reported.

Forced Answer Correctness	Model Judged "Can Answer"	Model Judged "Cannot Answer"	Total
Answer is Correct	149 (True Positives)	21 (False Positives)	170
Answer is Incorrect	130 (False Negatives)	71 (True Negatives)	201

Table 11: Sample distribution (#samples) of the answerability check in EXTAGENTS, comparing model judgement against accuracy when forced to answer, demonstrating effective safeguarding and confident hallucination.

2, according to Table 1). Specifically, Chain of Agents failed to achieve steady performance improvement when adding more input length with either the same or different chunk sizes. Generally, it degrades performance when inputting from 32k to 1024k, not able to handle overwhelming external knowledge. Also, Chain of Agents could not achieve better than directly inputting 128k tokens into the LLM, let alone EXTAGENTS with 1024k external knowledge input.

Robustness Analysis of the Answerability Check EXTAGENTS halts on Reasoning Agent deciding the query is answerable or hitting the maximum synchronization timestep T, and then outputting the answer (Section 4). To examine the robustness of the answerability check, we conducted a quantitative analysis of false positive answers versus appropriate refusals. We isolated this mechanism by removing the Knowledge-Accumulating Reasoning (KAR) strategy and providing exactly 4 chunks of information to the Reasoning Agent, whose role is to identify answerability and refuse insufficient queries. Using gpt-4o-mini, we evaluated 371 samples in HotpotQA to observe how the model's self-judged answerability aligns with its actual ability to answer correctly when forced. As shown in Table 11, results reveal: (1) **Effective Safeguarding** (False Positives): Only in 21/170 cases that model answers correctly, the model judged that it could not answer. This indicates the answerability check does not prevent correct answers from being outputted. (2) **Confident Hallucinations** (False Negatives): The model exhibited overconfidence in 130/201 cases, believing it had sufficient information to answer but ultimately providing an incorrect response. This highlights the inherent limitations of current LLMs regarding the self-assessment of their own knowledge bounds. It could be alleviated with stronger models, and we indeed observe stronger models have better end-to-end performance (Section 5.2).

Input Length (#Tokens)	T=1	T=2	T=5 (Default)	T=10
32k	.456	.454	.477	.480
128k	.485	.496	.500	.507
512k	.493	.494	.516	.521

Table 12: Impact of the maximum synchronization timesteps (T) in EXTAGENTS with gpt-4o-mini on HotpotQA (F1 Score). The chunk size is fixed to 32k tokens.

Detailed Ablations on Hyper-Parameters To evaluate the hyperparameter sensitivity of EXTAGENTS, we investigated the influence of the maximum synchronization timesteps (T) and the reasoning iteration scheduling strategy (S) on HotpotQA using gpt-4o-mini with a 32k-token chunk size. As detailed in Table 12, expanding the maximum syn-

Scheduling Strategy	F1 Score
Base-2 Logarithmic (Seq: $1, 2, 4, 8 \dots$)	.516
Base-4 Logarithmic (Seq: $1, 4, 16 \dots$)	.509
Proportional Fractional ($k = \lfloor N/2 \rfloor$)	.468
Exhaustive Accumulation ($k = N$)	.506
Small Window Constraint ($k = 4$)	.439

Table 13: Impact of reasoning iteration scheduling (S) in EXTAGENTS with gpt-4o-mini on HotpotQA. The input length is 512k tokens and the chunk size is 32k.

chronization timesteps consistently enhances F1 scores across various input context lengths, demonstrating the framework's ability to effectively leverage extended inference-time compute for complex queries. However, the performance improvements exhibit diminishing marginal utility beyond $T = 5$, validating our default configuration as an optimal equilibrium between reasoning accuracy and computational overhead. Furthermore, Table 13 illustrates the efficacy of different context accumulation schedules. Our default base-2 logarithmic scheduling (Seq: $1, 2, 4, 8 \dots$) achieves the highest score. Static strategies—such as a small window ($k = 4$) or a proportional window ($k = \lfloor N/2 \rfloor$)—deprive the Reasoning Agent of sufficient global context. Conversely, exhaustive accumulation degrades performance by reintroducing the information overload bottleneck characteristic of long-context processing, even with ranked orders.

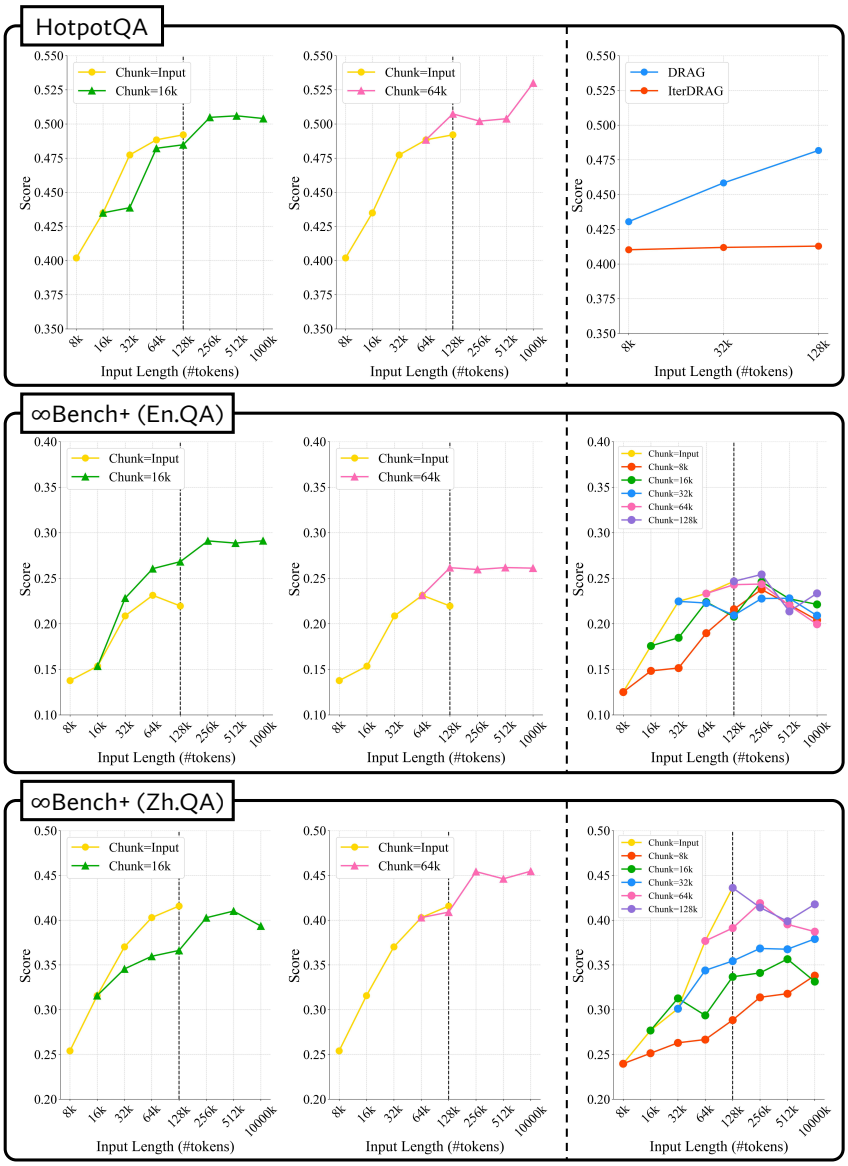

Figure 8: Detailed experimental results of scaling external knowledge input on multi-hop QA tasks, complementary to Figure 4 with the same subfigure arrangement.

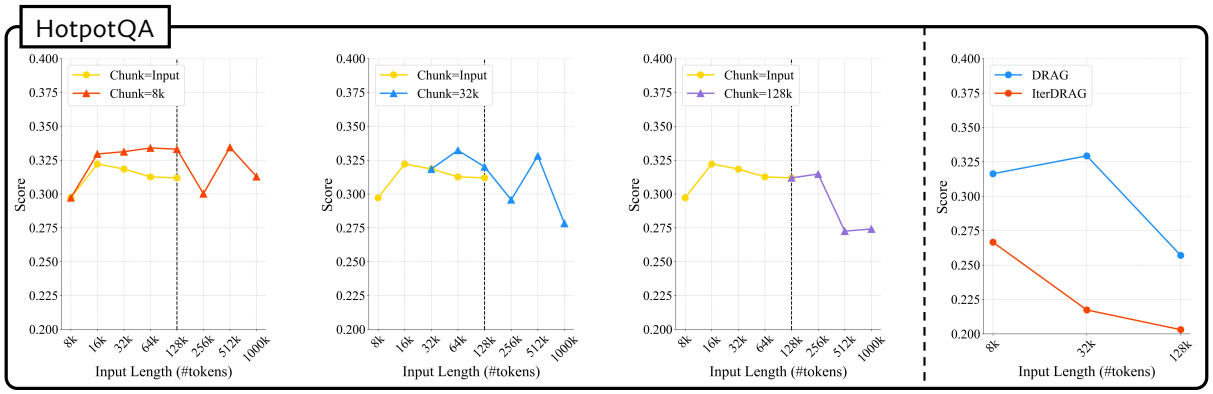

Figure 9: Results of ExtAgents with Llama-3.2-3B-Instruct on HotpotQA benchmark.

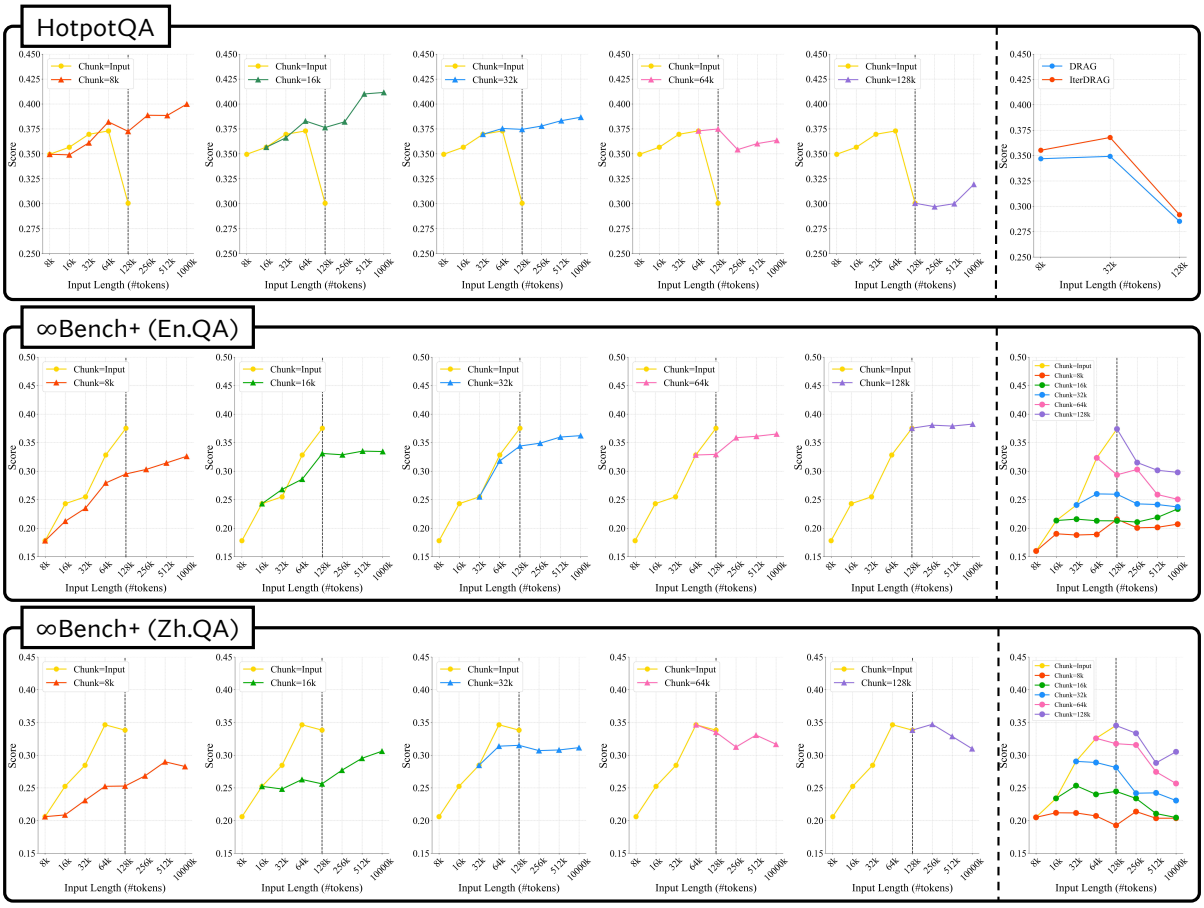

Figure 10: Experiment of scaling external knowledge input on multi-hop QA tasks. We plot gpt-4o-mini results for En.QA and Llama-3.1-8B-Instruct results for other tasks. Other arrangement of subfigures is the same as Figure 4.

C.3 Prompt Templates

Prompt Templates for HotpotQA

Knowledge synchronization: First iteration

We are working on long-text question answering, and you are responsible for one chunk. Read the following chunk and extract as much information as possible related to the question. Ensure your extracted information provides clear context and is logically complete. If no information, just output "NO INFORMATION".

Your chunk:
{Retrieved documents}

Question:{question}

Knowledge synchronization: Other iterations

We are working on long-text question answering, and you are responsible for one chunk. This is the {iteration} round of Q&A. And we have the previously extracted information from all chunks in the previous round. Based on the previously extracted information and question, extract new information from the chunk. Do not repeat the previously extracted information. If no new information, just output "NO INFORMATION".

Your chunk:
{Retrieved documents}

Previously extracted information:
{Previously extracted information}

Question: {question}

Knowledge-accumulating reasoning: No need to terminate

We have the following extracted information from different chunks of the text:

{Extracted information}

Based on the extracted information, decide whether you can confidently answer the question. If you can, combine and reduce this information into a final answer, as short as possible, word or phrase. If you cannot, just output "NO ANSWER".

Question: {question}

Knowledge-accumulating reasoning: Need to terminate

We have the following extracted information from different chunks of the text:

{Extracted information}

Based on the extracted information, combine and reduce this information into a final answer, as short as possible, word or phrase.

Question: {question}

Prompt Templates for En.QA in ∞Bench+

Knowledge synchronization: First iteration

Read the following article and extract as much information as possible related to the question.

{Context}

Question: {question}

Knowledge synchronization: Other iterations

We are working on long-text question answering, and you are responsible for one chunk. This is the {iteration} round of Q&A. And we have the previously extracted information from all chunks in the previous round. Based on the previously extracted information and question, extract new information from the chunk. Do not repeat the previously extracted information.

Your chunk:
{Context}

Previously extracted information:
{Extracted information}

Question: {question}

Knowledge synchronization: Ranking information

Based on the extracted information and question, provide a score (0-100) for how useful the extracted information is for answering this question.

Extracted information: {extracted information}

Question: {question}

Please follow this format:

Score: (0-100)

Knowledge-accumulating reasoning: No need to terminate

We have the following extracted information from different chunks of the text:

{Extracted information}

Based on the extracted information, decide whether you can confidently answer the question. If you can, combine and reduce this information into a final answer, as short as possible, word or phrase. If you cannot, just output "NO ANSWER".

Question: {question}

Knowledge-accumulating reasoning: Need to terminate

We have the following extracted information from different chunks of the text:

{Extracted information}

Based on the extracted information, combine and reduce this information into a final answer, as short as possible, word or phrase.

Question: {question}

Prompt Template for Zh.QA in ∞Bench+

Knowledge synchronization: First iteration

请阅读以下文章并尽可能提取与问题相关的信息。

{Context}

问题：{question}

Knowledge synchronization: Other iterations

我们正在进行长文本问答任务，你负责处理其中一个文本块。这是第{iteration}轮问答。我们在之前几轮已经对所有文本块中提取了信息。请基于先前提取的信息和问题，从当前文本块中提取新信息。不要重复已提取的信息。

你的文本块：
{Context}

先前提取的信息：
{Extracted information}

问题：{question}

Knowledge synchronization: Ranking information

根据提取的信息和问题，给出一个分数（0-100），评估提取的信息对回答该问题的有用程度。

提取的信息：{extracted information}

问题：{question}

请遵循以下格式：

Score: (0-100)

Knowledge-accumulating reasoning: No need to terminate

我们有以下从不同文本块中提取的信息：

{Extracted information}

根据提取的信息，请判断是否能确定地回答该问题。如果能，将这些信息合并并简化为最终答案。请尽量简短地回答，只使用一个或多个词语。如果不能，直接输出"NO ANSWER".

问题：{question}

Knowledge-accumulating reasoning: Need to terminate

我们有以下从不同文本块中提取的信息：

{Extracted information}

根据提取的信息，将这些信息合并并简化为最终答案。请尽量简短地回答，只使用一个或多个词语。

问题：{question}