Mitigating Context Interference for Reliable and Efficient Search Agents

Anonymous ACL submission

Abstract

Recent research empowers Large Language Models (LLMs) as multi-turn search agents to iteratively retrieve and generate outputs until complex tasks are solved. However, the contexts of multi-turn search agents are lengthy and complex. For example, the retrieved set of documents in each turn would inevitably introduce irrelevant information that distracts LLMs, referring to *context interference*, potentially hindering the reliability and efficiency of search agents. Therefore, we conduct a systematic study on context interference in multi-turn search agents, focusing on investigating i) which parts of the context of search agents will contribute to the context interference, ii) how to refine the contexts of search agents to mitigate the interference, and iii) can incorporating context refinement into search agent training yield further improvements. We reveal that interference primarily arises from the latest retrieved documents. Based on the explored findings, we then introduce a distill-based context refiner to dynamically mitigate context interference for multi-turn search agents. Finally, we validate that incorporating context refinement into RL training pipelines of search agents can significantly enhance both reliability and efficiency. This study highlights the importance of mitigating context interference of search agents, inspiring a novel paradigm of "refine context and then generate" for AI agents.

1 Introduction

Large Language Models (LLMs) have demonstrated strong performance in tackling complex tasks using their pretrained knowledge (OpenAI, 2025; DeepSeek-AI, 2025). Recent work has further empowered them to invoke search engines to retrieve external knowledge, essentially training them as multi-turn search agents that iterate on retrieval and generation until tasks are solved (Wang et al., 2025; Jia et al., 2025; Huang et al., 2025).

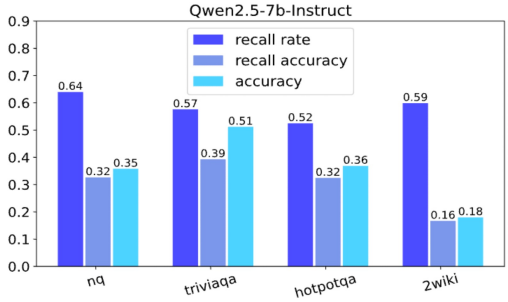

Figure 1: Demonstration of how context interference affects the performance of LLM-based search agents. ("recall rate"=N_{r}/N denotes the proportion of questions for which the retrieved documents contain the correct answer (N_{r}) among all questions (N). "recall accuracy"=N_{rc}/N refers to the proportion of correctly answered questions (N_{rc}) in N_{r}, relative to all questions (N). "accuracy" represents the proportion of all correctly answered questions N_{c} out of the questions (N).)

However, the contexts of multi-turn search agents are lengthy and complex, encompassing the question, multi-round search queries, retrieved documents, and reasoning steps (Jin et al., 2025), which may include irrelevant information. For instance, the retriever always returns a set of documents to ensure coverage of the search query, which also introduces noisy or irrelevant documents into the context (Dong et al., 2025). This refers to *context interference*, which indicates "feeding too much irrelevant context may confuse LLMs to focus on wrong information (Coleman et al., 2023; Haseeb, 2025; Gupta et al., 2024; Jiang et al., 2025)." The context interference presented in each round may distract LLMs from irrelevant information and persistently degrade subsequent generation quality (Li et al., 2025; Laban et al., 2025), thereby hindering the efficiency and reliability of search agents. As in Figure 1, the gaps between "`recall rate`" and "`recall accuracy`" demonstrate that search agents always retrieve the documents containing the useful information but

Figure 2: Examples of the search agent with (a) contextual interference in the retrieved documents and (b) refined contexts with the most critical relevant information. The question derives from PopQA (Mallen et al., 2023), and we employ Qwen2.5-7b-Instruct (Qwen et al., 2025) as the foundation LLM of the search agent.

fail to generate the correct answer, highlighting the context interference effect of accurate knowledge expression of search agents.

Prior studies to mitigate context interference have predominantly centered on dialogue systems (Jacqmin et al., 2022) and retrieval-augmented generation (RAG) (Glass et al., 2022; Nguyen et al., 2025; Yu et al., 2024), while largely overlooking techniques on multi-turn search agent settings. Therefore, we systematically study the *context interference* issue on search agents in this work with three research questions (**RQ**):

i) *Which parts of contexts will contribute to context interference for multi-turn search agents?*

ii) *How to refine contexts of search agents to mitigate such interference?*

iii) *Can leveraging context refinement in RL training pipelines of search agents yield further performance improvements?*

In light of the above questions, we investigate the context interference effect on multi-turn search agents with respect to **reliability** and **efficiency** across a series of closed-book QA benchmarks. For **RQ i**, we compare the performances of multi-turn search agents with different inputs by masking specific parts of history contexts. Results identify that context interference primarily derives from the latest retrieved document of search agents and slightly arises from previous search queries and documents. For **RQ ii**, we first explore a series of context interference mitigation strategies based on the conclusion from **RQ i**. Based on the exploration, we propose to distill a context refinement dataset comprising both retrieved documents and refined texts, which contains the most critical information in documents to the search query. Then we train a context refiner using the dataset, which can mitigate context interference for multi-turn search agents as exemplified in Figure 2 and improve both efficiency and reliability. For **RQ iii**, we further incorporate the context refiner into the reinforcement learning (RL) training pipelines of search agents. Experiments demonstrate that training multi-turn search agents with dynamically refined context achieves significant performance improvements regarding both the reliability and efficiency over other training baselines.

The contributions of this work are as follows:

(1) This work first investigates the context interference issue on multi-turn search agents, highlighting the necessity of context refinement to improve both efficiency and reliability of search agents, inspiring a novel paradigm of "refine context and then generate" for AI agents.

(2) This work reveals that context interference primarily derives from the latest documents in multi-turn search agents, and therefore introduces a distill-based context refiner to dynamically eliminate context interference for search agents, which can be applicable to mitigate contextual interference in other search agent scenarios.

(3) This work further incorporates context refinement into RL training pipelines of search agents, which can further enhance both reliability and efficiency, providing insight into refining contexts during search agent training for future work.

2 Preliminary of Search Agent

To establish a theoretical foundation to analyze the context interference issue, we introduce the concepts of the search agent's internal/external knowledge, a Markov Decision Process (MDP), and practical settings of multi-turn search agents. Notation definitions of this work can be found in Appendix A. Related works are in Appendix B.

Internal/External Knowledge of Search Agent Previous works identify the concept of *Internal/External Knowledge* for LLM agent (Wang

et al., 2025; Jia et al., 2025), where internal knowledge $\mathcal{K}_I$ is learned from the pretrained corpus and encoded within the model parameters, and external knowledge $\mathcal{K}_E$ is accessed through external tools (e.g., search engines). For an LLM-based search agent $\mathcal{M}$ with parameters θ, the output $\boldsymbol{y}$ is jointly determined by its internal parametric knowledge $\mathcal{K}_I \in \theta$, the externally retrieved documents $\boldsymbol{d} \in \mathcal{K}_E$, and the input task $\boldsymbol{x}$ as $\boldsymbol{y} = \mathcal{M}_\theta(\boldsymbol{x}, \boldsymbol{d})$, where $\boldsymbol{x}$ generally refers to the query and sequentially concatenated history outputs.

As $\mathcal{K}_I$ embedded in parameters θ cannot be directly accessed, the manifestation of $\mathcal{K}_I$ is conditioned on the context of $\boldsymbol{x}$ and $\boldsymbol{d}$. The retriever generally returns a set of documents in $\boldsymbol{d} \in \mathcal{K}_E$ intended to comprehensively cover the search query, but this inevitably introduces redundancy and noise. Such extraneous content can in turn distort the utilization of both $\mathcal{K}_I$ and $\mathcal{K}_E$, and undermine the reliability of search agents. In Figure 1, "recall rate" denotes the questions calling $\mathcal{K}_E$ containing helpful information, while "recall accuracy" refers to actually solved questions using $\mathcal{K}_E$, and "accuracy" represents successfully answered questions through the combined use of $\mathcal{K}_I$ and $\mathcal{K}_E$. The gaps between "recall rate" and "recall accuracy" / "accuracy" exemplify the negative impact of context interference. Moreover, distractions from irrelevant information will incur extra retrievals and inference costs, degrading the efficiency of search agents.

Markov Decision Process of Agent The process of an LLM agent $\mathcal{M}_\theta$ with $\mathcal{K}_I \in \theta$ interacting with an environment $\boldsymbol{E}$ (referred to $\mathcal{K}_E$) to complete a task can be regarded as a Markov Decision Process: $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{R})$. Initially, a specific task $\boldsymbol{x}$ is provided as the initial environmental state. Assuming the interaction proceeds in N rounds and in round n ($n < N$), the LLM agent receives observation $\boldsymbol{o}_n \in \mathcal{O}$ and takes action $\boldsymbol{a}_n \in \mathcal{A}$. The state $\boldsymbol{s}_n$ at round n is the history context of all preceding concatenated sequences $\boldsymbol{s}_n = (\boldsymbol{x}, \boldsymbol{a}_0, \boldsymbol{o}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{n-1}, \boldsymbol{o}_n) \in \mathcal{S}$. $\mathcal{M}_\theta$ is responsible for deciding $\boldsymbol{a}_n$ based on $\boldsymbol{s}_n$: $\boldsymbol{a}_n \sim \mathcal{M}_\theta(\cdot|\boldsymbol{s}_n)$, and a retriever $\mathcal{E}$ interacts with $\boldsymbol{E}$ to determines the state transition $\mathcal{T}$. Upon task completion after N rounds, the trajectory is characterized as $\boldsymbol{\tau} = (\boldsymbol{x}, \boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{o}_1, \ldots, \boldsymbol{a}_{N-1}, \boldsymbol{o}_N, \boldsymbol{a}_N)$ where the final response $\boldsymbol{y}$ is parsed from $\boldsymbol{a}_N$. A final reward r is provided by the reward model $\mathcal{R}$ by comparing $\boldsymbol{y}$ and the reference $\hat{\boldsymbol{y}}$ as $r = \mathcal{R}(\boldsymbol{y}, \hat{\boldsymbol{y}})$.

Multi-turn Search Agent For multi-turn search agents, the i-th observation $\boldsymbol{o}_i$ is generally a list of retrieved Top-K documents $\boldsymbol{d}_i = [\boldsymbol{d}_{i,1}, \boldsymbol{d}_{i,2}, \ldots, \boldsymbol{d}_{i,K}]$ returned by the retriever $\mathcal{E}(\boldsymbol{q}_{i-1}|\boldsymbol{K}_b)$ where $\boldsymbol{q}_{i-1}$ is the search query generated in the previous round and $\boldsymbol{K}_b$ is the external knowledge base as $\mathcal{K}_E$. The action $\boldsymbol{a}_i$ includes a thinking step $\boldsymbol{t}_i$ and a search query $\boldsymbol{q}_i$. Following Jin et al. (2025), LLMs are instructed to encapsulate their search queries, retrieved documents, and final answer between specially designated tokens respectively. Both $\boldsymbol{q}_i$ and $\boldsymbol{d}_i$ are appended to the context in each turn. When generating $\boldsymbol{t}_i$, all the preceding sequences in $\boldsymbol{s}_i$ are fed into $\mathcal{M}_\theta$ and contribute to $\boldsymbol{a}_i$. However, not all previous documents and search queries are pertinent to the current thinking step, and the inclusion of irrelevant content can introduce context interference, impairing the LLMs' efficiency and reliability to accurately express knowledge.

3 Context Interference in Search Agent

In this section, we first detail the evaluation setting in Sec. 3.1. Then we analyze context interference effects in different parts of contexts of search agents in Sec. 3.2, and finally propose the context refiner to mitigate interference in Sec 3.3.

3.1 Evaluation Settings

Dataset Various closed-book QA datasets are employed to evaluate the performance of search agents, which necessitate extra retrieval to address, encompassing both single- and multi-hop scenarios. **Single-hop QA** includes: *Natural Questions (NQ)* (Kwiatkowski et al., 2019), *TriviaQA* (Joshi et al., 2017), and *PopQA* (Mallen et al., 2023). **Multi-hop QA** includes: *HotpotQA* (Yang et al., 2018), *2WikiMultiHopQA (2Wiki)* (Ho et al., 2020), *MuSiQue* (Trivedi et al., 2022), and *Bamboogle* (Press et al., 2023). Dataset details are presented in Appendix C.

Search Agent We employ two foundation LLMs $\mathcal{M}$ for search agents: **Qwen-2.5-7b-Instruct** and **Qwen-2.5-3b-Instruct** (Qwen et al., 2025). For retrieval, we leverage E5 (Wang et al., 2024) as the retriever $\mathcal{E}$ and 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge base $\boldsymbol{K}_b$ respectively. The number of retrieved passages K is set to 3 across all retrieval-based methods.

Evaluation Metrics We employ several metrics to evaluate both the **reliability** and **efficiency** of

Methods	Single-Hop QA			Multi-Hop QA				*Avg.*
	NQ	TriviaQA	PopQA	HotpotQA	2wiki	Musique	Bamboogle	
Qwen2.5-7b-Instruct								
Direct	17.7 / 0.0	44.2 / 0.0	15.5 / 0.0	17.9 / 0.0	**24.0** / 0.0	3.9 / 0.0	8.0 / 0.0	18.7 / 0.0
CoT	17.7 / 0.0	47.0 / 0.0	13.4 / 0.0	21.0 / 0.0	23.6 / 0.0	4.7 / 0.0	30.4 / 0.0	22.5 / 0.0
IRCoT	30.6 / 2.0	51.2 / 1.8	31.7 / 2.1	24.6 / 3.0	18.0 / 3.4	9.8 / 3.1	28.8 / 2.5	27.5 / 2.6
IRCoT-*o*	29.8 / **1.8**	51.3 / **1.7**	32.7 / **1.7**	25.3 / 2.7	22.7 / 3.1	11.2 / 2.9	34.4 / 1.9	29.6 / **2.3**
IRCoT-*oq*	32.8 / 2.0	**51.7** / 2.0	**33.0** / 2.0	**26.0** / 3.0	23.6 / **3.1**	**11.4** / 3.1	34.4 / **1.7**	**30.4** / 2.4
IRCoT-*oqt*	**32.9** / 2.6	51.5 / 2.7	32.7 / 2.8	25.4 / **2.6**	22.8 / 3.1	10.6 / **2.8**	**36.0** / 2.5	30.3 / 2.7
Qwen2.5-3b-Instruct								
Direct	9.7 / 0.0	25.7 / 0.0	7.7 / 0.0	13.5 / 0.0	17.1 / 0.0	1.7 / 0.0	3.2 / 0.0	11.2 / 0.0
CoT	12.4 / 0.0	35.4 / 0.0	9.4 / 0.0	15.5 / 0.0	16.4 / 0.0	2.6 / 0.0	**20.0** / 0.0	16.0 / 0.0
IRCoT	21.6 / **1.2**	45.2 / 1.1	29.6 / 1.1	24.1 / 1.6	23.5 / 1.9	6.8 / 1.6	19.2 / 1.4	24.3 / 1.4
IRCoT-*o*	22.4 / 1.2	47.0 / **1.1**	**31.8** / **1.1**	24.7 / **1.5**	21.3 / **1.7**	**8.0** / 1.5	19.4 / **1.2**	24.9 / **1.3**
IRCoT-*oq*	23.5 / 1.3	47.2 / 1.2	29.9 / 1.4	**24.9** / 1.6	**24.2** / 1.7	7.0 / 1.6	19.4 / 1.2	**25.2** / 1.4
IRCoT-*oqt*	**24.6** / 1.5	**47.7** / 1.3	30.5 / 1.5	24.5 / 1.8	22.5 / 1.7	5.1 / **1.5**	19.0 / 1.2	24.8 / 1.5

Table 1: Performance (EM/ART) of different inference methods for search agents across QA test sets, measured by Exact Match (EM, for reliability) and Average Retrieval Times (ART, for efficiency).

search agents. For reliability, we assess the correctness of the generated answer $\boldsymbol{y}$ with the reference $\hat{\boldsymbol{y}}$ using **Exact Match (EM)** (Song et al., 2025; Jin et al., 2025), a standard string-matching metric that checks whether $\boldsymbol{y} \equiv \hat{\boldsymbol{y}}$. Efficiency is evaluated regarding different aspects. Since the time cost of search agents is primarily determined by the number of retrieval operations, the **average retrieval times (ART)** is employed to intuitively evaluate the efficiency of the generation processes. In addition, the **average context length (Len.)** in multi-turn generations and the **average inference time per question (AIT)** (/seconds) are also employed to assist efficiency assessments for search agents.

3.2 Context Interference in Different Parts of Context of Search Agents

Background To figure out context interference effects of different parts in the contexts of search agents (as **RQ i**), we compare the performance of multi-turn search agents with different input contexts by masking specific segments (actions and observations in preceding rounds) of the history state. The history state $\boldsymbol{s}_i$ includes input question $\boldsymbol{x}$ and a series of actions $\boldsymbol{a}_{0:i-1}$ and observations $\boldsymbol{o}_{1:i}$. Specifically, $\boldsymbol{x}$ representing the initial state is fixed. $\boldsymbol{t}_i$ in $\boldsymbol{a}_i$ generally involve summarizing and reasoning from $\boldsymbol{o}_i$. $\boldsymbol{q}_i$ is to interact with $\boldsymbol{K}_b$ to get retrieved documents in $\boldsymbol{o}_{i+1}$. We present several inference methods for search agents as follows.

Inference Methods We employ direct inference (**Direct**) and Chain-of-Thought (**CoT**) reasoning (Wei et al., 2022) as two retrieval-free baselines, which represent LLMs' internal knowledge $\mathcal{K}_I$ to answer questions. The prompt templates are in Appendix E. For retrieval-based settings of search agents, we employ Information Retrieval with CoT (**IRCoT**) (Trivedi et al., 2023a), which enables LLMs to actively call the retriever for questions beyond their knowledge scope after thinking.

To understand the effect of different parts of context, several variants based on **IRCoT** are developed. Since search agents typically decompose a complex question into a set of sub-questions (Jin et al., 2025; Song et al., 2025), the generated search queries and retrieved documents in a history state are generally mutually independent, exhibiting rarely sequential dependencies. Therefore, we can specifically mask different parts in the states as follows. 1) To investigate interference in previous documents, **IRCoT-w/*o*** (**w/o** $\boldsymbol{o}_{:-1}$) only incorporates the latest observation $\boldsymbol{o}_i$ of retrieved documents in context $[\boldsymbol{x}, \boldsymbol{t}_0, \boldsymbol{q}_1, \boldsymbol{t}_1, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_{i-1}, \boldsymbol{o}_i]$ when generating $\boldsymbol{a}_i$. 2) For interference in search queries, **IRCoT-w/*oq*** (**w/o** $\boldsymbol{o}_{:-1}, \boldsymbol{q}_{:-1}$) with context $[\boldsymbol{x}, \boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{i-1}, \boldsymbol{q}_{i-1}, \boldsymbol{o}_i]$ is also employed. 3) For previous thinking steps, we utilize **IRCoT-w/*oqt*** (**w/o** $\boldsymbol{o}_{:-1}, \boldsymbol{q}_{:-1}, \boldsymbol{t}_{:-1}$) relies exclusively on the latest thinking, search query, and observation as context $[\boldsymbol{x}, \boldsymbol{t}_{i-1}, \boldsymbol{q}_{i-1}, \boldsymbol{o}_i]$ when generating $\boldsymbol{a}_i$.

Analysis and Findings As presented in Table 1, benefiting from $\boldsymbol{K}_b$, all retrieval-based methods outperform retrieval-free baselines. Moreover, **IRCoT-*o*** outperforms **IRCoT** in both reliability and efficiency, suggesting that previous retrieved documents before round i contain context interference for generating $\boldsymbol{a}_i$. **IRCoT-*oq*** marginally outperforms **IRCoT-*o***, indicating that previous

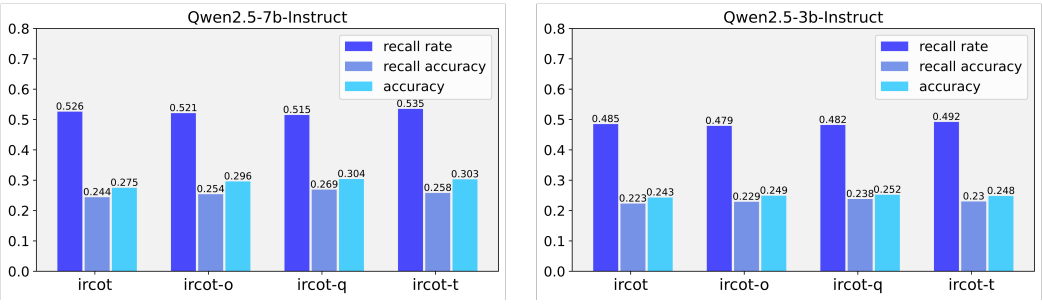

Figure 3: Demonstrations of context interference effects on four IRCoT variants of search agents. Results are averaged on all QA test sets; metrics follow the definitions in Figure 1.

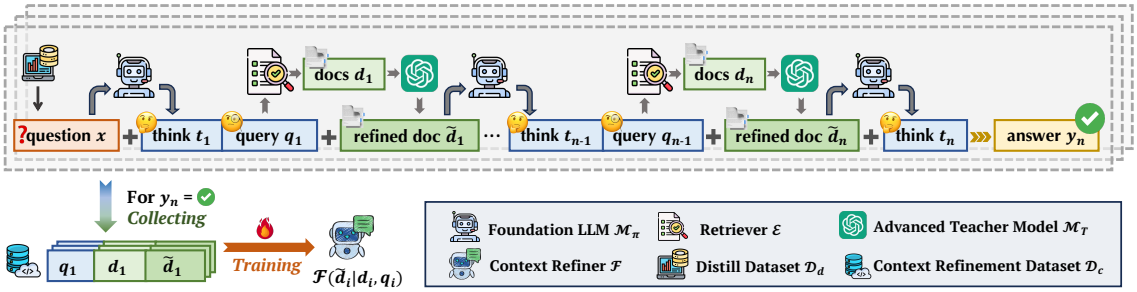

Figure 4: Training pipeline for the distill-based context refiner.

search queries $\boldsymbol{q}_{:-1}$ also carry slight interference, although removing them may incur little extra retrieval. **IRCoT-*oqt*** exhibits a slight drop in reliability and a notable decrease in efficiency compared with others, implying that previous thinking steps store key information for future steps. LLMs may repeat previous search after masking $\boldsymbol{t}_{:-1}$ and thus add retrieval costs. Consequently, **context interference when generating $\boldsymbol{a}_i$ may arise from both previous search queries and documents**.

However, as presented in Figure 3, although the above **IRCoT** variants gain slight improvements in accuracy, the gap between "recall rate" and "recall accuracy" remains considerable, indicating context interference in previous rounds is not the dominant factor. Since the latest thinking $\boldsymbol{t}_{i-1}$ and search query $\boldsymbol{q}_{i-1}$ mostly encapsulate summaries and reasoning of previous context, it can be inferred that **the latest observation $\boldsymbol{o}_i$ is subject to the primary cause of context interference**. To mitigate the interference, further context refinement methods for the latest observed documents are required on multi-turn search agents.

3.3 Context Refiner for Search Agent

Background Our preliminary findings and analysis in Sec. 3.2 demonstrate that we can marginally mitigate context interference by removing irrelevant information like previous documents and search queries in multi-turn search agents but not enough, which highlights the necessity of further capturing critical information and filtering noise in the latest retrieved documents (as **RQ ii**). More generally, we desire to develop a context refiner $\boldsymbol{\mathcal{F}}$ that, in each round i, refine the context to preserve the most relevant information $\tilde{\boldsymbol{d}}_i = \boldsymbol{\mathcal{F}}(\boldsymbol{q}_{i-1}, \boldsymbol{d}_i)$ to search query $\boldsymbol{q}_{i-1}$ from the latest retrieved documents $\boldsymbol{d}_i$. $\tilde{\boldsymbol{d}}_i$ is served as the i-th observation $\boldsymbol{o}_i$ and then appended to $\boldsymbol{s}_i$ to generate $\boldsymbol{a}_i$.

Prior works on mitigating context interference are demonstrated in Appendix B.2. However, directly designing precise schemes to capture key information from numerous documents is challenging (Glass et al., 2022). A compression model may focus on summarizing global contents, potentially leading to information loss or the introduction of extraneous knowledge (Li et al., 2025). Relatively small LLMs exhibit limited capability for key information extraction (in Table 2). Therefore, we propose to distill a dataset for context refinement from advanced LLMs, enabling relatively weak models to refine context to mitigate interference.

Context Refiner Given a distill dataset $\boldsymbol{\mathcal{D}}_d = \{\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i\}_{i=1}^N$, a foundation LLM $\boldsymbol{\mathcal{M}}_\pi$, an advanced teacher LLM $\boldsymbol{\mathcal{M}}_T$, a retriever $\boldsymbol{\mathcal{E}}$, and a knowledge base $\boldsymbol{K}_b$, we infer each query $\boldsymbol{x}$ on $\boldsymbol{\mathcal{M}}_T$ using IRCoT (Trivedi et al., 2023b). In i-th round

Methods	Single-Hop QA			Multi-Hop QA				*Avg.*
	NQ	TriviaQA	PopQA	HotpotQA	2wiki	Musique	Bamboogle	
Qwen2.5-7b-Instruct								
IRCoT	30.6 / 2.0	51.2 / 1.8	31.7 / 2.1	24.6 / 3.0	18.0 / 3.4	**9.8** / 3.1	28.8 / 2.5	27.5 / 2.6
GPT-Compress	32.5 / **0.9**	55.5 / 1.0	34.8 / 1.1	29.8 / 1.6	25.0 / 1.5	9.6 / 1.3	26.4 / 1.1	30.5 / **1.2**
GPT-Refine	**34.6** / 1.1	**57.5 / 0.9**	**37.0** / 1.0	**33.0 / 1.3**	**27.6** / 1.8	9.3 / 1.4	**33.6** / 1.0	**33.2** / 1.2
Self-Refine	32.5 / 1.5	53.3 / 0.9	33.0 / **1.0**	27.0 / 1.4	21.6 / **1.5**	8.4 / **1.2**	28.8 / **1.0**	29.2 / 1.2
Context Refiner	34.0 / 1.2	56.9 / 0.9	36.4 / 1.0	32.0 / 1.4	27.2 / 1.6	8.6 / 1.2	30.4 / 1.0	32.2 / 1.2
Qwen2.5-3b-Instruct								
IRCoT	21.6 / 1.2	45.2 / 1.1	29.6 / 1.1	24.1 / 1.6	**23.5** / 1.9	6.8 / 1.6	19.2 / 1.4	24.3 / 1.4
GPT-Compress	31.3 / **0.9**	51.5 / 0.9	29.9 / 1.0	24.1 / 1.1	22.7 / **1.2**	7.2 / 1.0	18.4 / 0.9	26.4 / 1.0
GPT-Refine	**32.6** / 1.0	**54.5 / 0.8**	**35.6 / 0.9**	**24.4 / 1.0**	22.5 / 1.3	**7.5 / 0.9**	**24.0 / 0.9**	**28.7 / 1.0**
Self-Refine	23.7 / 1.1	47.6 / 0.9	30.8 / 0.9	23.1 / 1.0	21.4 / 1.2	5.0 / 1.1	20.0 / 0.9	24.5 / 1.0
Context Refiner	30.7 / 1.0	50.8 / 0.9	32.0 / 0.9	24.1 / 1.0	22.2 / 1.2	7.2 / 1.0	19.4 / 0.9	26.6 / 1.0

Table 2: Performance (EM/ART) of context refinement methods for mitigating interference, measured by Exact Match (EM) and Average Retrieval Times (ART) across QA test sets.

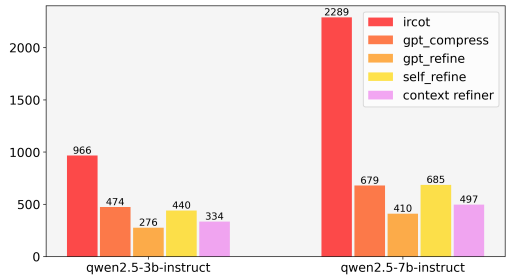

Figure 5: Averaged context lengths of search agents across all QA test sets, comparing IRCoT (baseline), three context refinement methods, and the proposed context refiner for efficiency assessments.

of inference for $\boldsymbol{x}$, the retriever return documents $\boldsymbol{d}_i = \mathcal{E}(\boldsymbol{q}_{i-1}|\boldsymbol{K}_b)$. Then we instruct the teacher model $\mathcal{M}_T$ to specifically extract only critical information related to $\boldsymbol{q}_{i-1}$ from $\boldsymbol{d}_i$. Extracted information $\tilde{\boldsymbol{d}}_i = \mathcal{M}_T(\boldsymbol{q}_{i-1}, \boldsymbol{d}_i)$ is then appended into context $\boldsymbol{s}_i$ to generate the next-step action $\boldsymbol{a}_i$. For each correct trajectory with $\boldsymbol{y} \equiv \tilde{\boldsymbol{y}}$, we retain all step-wise pairs of extracted data $(\tilde{\boldsymbol{d}}_i, <\boldsymbol{d}_i, \boldsymbol{q}_{i-1}>)$ and employ an entailment model to verify that $\tilde{\boldsymbol{d}}_i$ is entirely encompassed within $\boldsymbol{d}_i$ and does not introduce extra knowledge. Finally, all qualified data points are re-formatted and incorporated into the context refinement dataset $\mathcal{D}_\text{c} = \{\tilde{\boldsymbol{d}}_j, \boldsymbol{d}_i, \boldsymbol{q}_i\}_{i=1}^{M}$.

Given $\mathcal{D}_\text{c}$, we train the model $\mathcal{M}_\pi$ to enable its ability of context refinement using supervised fine-tuning (SFT) as follows.

$$\pi^* = \arg\min_{\pi} \mathcal{L}_\pi^{\text{SFT}} \tag{1}$$

$$\mathcal{L}_\pi^{\text{SFT}} = -\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{(\tilde{\boldsymbol{d}}_i, \boldsymbol{d}_i, \boldsymbol{q}_i)\sim\mathcal{D}_\text{c}} \mathcal{L}_\pi^{(i)} \tag{2}$$

$$\mathcal{L}_\pi^{(i)} = \log \mathcal{M}_\pi(\tilde{\boldsymbol{d}}_i \mid \boldsymbol{d}_i, \boldsymbol{q}_i) \tag{3}$$

$\mathcal{M}_\pi$ is trained to generate refined documents $\tilde{\boldsymbol{d}}_i$, yielding a context refiner $\mathcal{F} = \mathcal{M}_{\pi^*}$ for dynamic context refinement in multi-turn search agents.

Context Refinement Methods We employ several existing context refinement methods as comparisons: 1) We utilize the compression method by introducing GPT-4 (OpenAI, 2023) to summarize the previous contexts (**GPT-Compress**); 2) We employ GPT-4 to dynamically refine the latest Top-K retrieved documents $\boldsymbol{d}_i$ in each round based on the search query $\boldsymbol{q}_i$ (**GPT-Refine**). 3) We also employ the foundation LLM itself $\mathcal{M}_\pi$ to dynamically refine the latest documents $\boldsymbol{d}_i$ using $\boldsymbol{q}_i$ (**Self-Refine**).

Results and Analysis As in Table 2 and Figure 5, **IRCoT** baseline is also presented for intuitive comparisons of context interference mitigation. **GPT-Refine** consistently outperforms **GPT-Compression** in terms of reliability, while being slightly inferior in search times and context length, indicating that extracting and preserving search query-relevant key information is more effective in mitigating contextual interference than general summarization. These two baselines can effectively reduce the retrieval wastes of search agents on irrelevant information in context over **IRCoT**. **Self-Refine**, by contrast, does not yield performance improvements and even fails to reduce context length, which can be attributed to the lack of information extraction capability of the foundation model $\mathcal{M}_\pi$. In comparison, **Context Refiner** trained on the context refinement dataset achieves close performance to **GPT-Refine**, suggesting that even relatively weak LLMs can also acquire the ability to refine context through fine-tuning, without relying on external models for search agents.

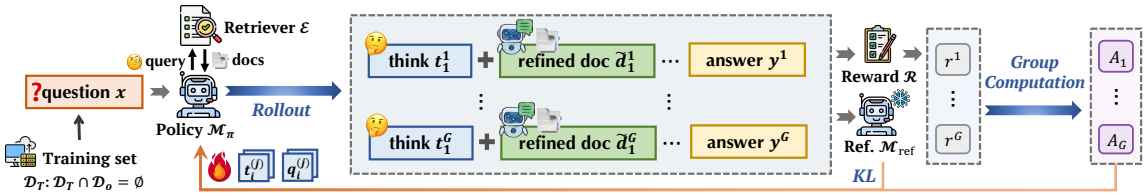

Figure 6: Demonstration of our proposed CRRL method.

4 Context Refinement for Search Agent Training

To further explore the potential of mitigating context interference in search agent training pipelines (as **RQ iii**), we extend our context refiner to the RL training pipeline of search agents, proposing a novel **C**ontext-**R**efined **R**einforcement **L**earning (CRRL) framework to dynamically refine context during training rollouts, reducing context interference in trajectory quality as follows.

4.1 Context-Refined Reinforcement Learning

Reinforcement learning (RL) (Kaelbling et al., 1996) has emerged as a paradigm for search agent training (Song et al., 2025; Jin et al., 2025; Chen et al., 2025) using PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). During rollouts, these RL algorithms enable LLMs to iteratively interact with search engines and append the retrieved documents to contexts, potentially introducing interference in RL training and resulting in suboptimal performance of RL. Nonetheless, current RL methods overlook the impact of context interference of rollouts. Derived from Sec. 3, we have figured out that removing previous documents and search queries, as well as leveraging context refiner for the latest documents in contexts, can eliminate context interference for multi-turn search agents, which can also improve the quality of rollouts. Therefore, we propose the CRRL algorithm.

Current RL pipelines for search agents are mainly based on Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO performs multiple rollouts per task and calculates the relative reward within the group as the advantage, which is more lightweight without the value model and demonstrates comparable performance with PPO (Jin et al., 2025). Hence, this paper adopts GRPO as the default RL algorithm, and the proposed CRRL is also based on GRPO.

To mitigate context interference of GRPO rollouts and obtain high-quality trajectories, our CRRL dynamically refines the input context during rollout of multi-turn search agents with the context refiner $\mathcal{F}$. As shown in Figure 6, during rollout, the trajectories of CRRL only contain thinking steps and refined documents, which indicates that the action $\boldsymbol{a}_i^j = [\boldsymbol{t}_i^j, \boldsymbol{q}_i^j]$ is produced with the context $\boldsymbol{s}_i^j = [\boldsymbol{x}, \boldsymbol{t}_{0:i-1}^j, \boldsymbol{q}_{i-1}^j, \tilde{\boldsymbol{d}}_i^j]$ where $\tilde{\boldsymbol{d}}_i^j = \mathcal{F}(\boldsymbol{d}_i^j), \boldsymbol{d}_i^j = \mathcal{E}(\boldsymbol{q}_{i-1}^j)$. RL pipelines for search agents explicitly incorporate retrieval interleaved reasoning, and the token-level losses are only computed over the rollouts of LLM-generated tokens, including both search queries and thinking steps, where loss masking is introduced for retrieved tokens, ensuring the stabilization of training while preserving the ability to adaptively retrieve. When optimize optimize the policy $\mathcal{M}_\pi$ on $\boldsymbol{a}_i = [\boldsymbol{t}_i, \boldsymbol{q}_i]$, the CRRL algorithm can be represented as follows:

$$\mathcal{L}_{\mathcal{M}_\pi}^{\text{CRRL}} = -\mathbb{E}_{\{\boldsymbol{a}_i^j\}_{j=1}^G \sim \mathcal{M}_{\text{ref}}(\cdot|\boldsymbol{s}_i^j)} \left[\mathcal{G}_\pi - \beta \text{KL}\right] \tag{4}$$

$$\mathcal{G}_\pi = \frac{1}{G} \sum_{j=1}^{G} \frac{1}{\sum_{k=1}^{N-1} |a_{i,k}^j|} \left[\mathcal{R}_\pi^j\right] \tag{5}$$

$$\mathcal{R}_\pi^j = \sum_{k}^{N-1} \min\left(r_\pi^j A_j, \text{clip}\left(r_\pi^j, 1-\epsilon, 1+\epsilon\right) A_j\right) \tag{6}$$

$$r_\pi^j = \frac{\mathcal{M}_\pi(a_{i,k}^j | \boldsymbol{x}, \boldsymbol{a}_{i,<k}^j, \boldsymbol{s}_i^j)}{\mathcal{M}_{\text{ref}}(a_{i,k}^j | \boldsymbol{x}, \boldsymbol{a}_{i,<k}^j, \boldsymbol{s}_i^j)} \tag{7}$$

where $\mathcal{M}_{\text{ref}}$ represents reference model. The term ϵ is a clipping ratio. β is the coefficient for the KL divergence. The advantage estimate is computed on the group-relative rewards of trajectory $\boldsymbol{\tau}^j$ as

$$A_j = \frac{r^j - \mu^j}{\sigma^j} \tag{8}$$

where $r^j = \mathcal{R}(\boldsymbol{y}^j)$ is the final reward, and μ^j and σ^j denote the mean and standard deviation of the rewards within the group.

4.2 Training Setup

We sample 40k data from the training sets of NQ and HotpotQA and merge them into $\mathcal{D}_t$ where

Methods	Single-Hop QA			Multi-Hop QA				*Avg.*
	NQ	TriviaQA	PopQA	HotpotQA	2wiki	Musique	Bamboogle	
Qwen2.5-7b-Instruct								
IRCoT	30.6 / 2.0	51.2 / 1.8	31.7 / 2.1	24.6 / 3.0	18.0 / 3.4	9.8 / 3.1	28.8 / 2.5	27.5 / 2.6
SFT	26.7 / 0.0	40.5 / 0.0	15.3 / 0.0	22.4 / 0.0	20.5 / 0.0	6.3 / 0.0	11.2 / 0.0	20.4 / 0.0
R1	24.7 / 0.0	55.4 / 0.0	20.0 / 0.0	19.4 / 0.0	17.5 / 0.0	7.8 / 0.0	22.4 / 0.0	23.9 / 0.0
RFT	34.4 / 2.2	57.5 / 2.0	36.3 / 2.4	31.8 / 3.0	26.5 / 3.6	11.5 / 3.2	33.6 / 2.7	33.1 / 2.7
Search-GRPO	35.5 / 1.7	58.4 / 1.4	38.5 / 1.8	35.6 / 2.7	27.0 / 3.0	11.5 / 2.3	36.0 / 1.8	34.6 / 2.1
CRRL	**38.1 / 1.4**	**60.1 / 1.2**	**39.4 / 1.6**	**38.2 / 2.0**	**30.5 / 2.5**	**13.2 / 1.9**	**36.8 / 1.5**	**36.6 / 1.7**
Qwen2.5-3b-Instruct								
IRCoT	21.6 / 1.2	45.2 / 1.1	29.6 / 1.1	24.1 / 1.6	23.5 / 1.9	6.8 / 1.6	19.2 / 1.4	24.3 / 1.4
SFT	23.8 / 0.0	36.2 / 0.0	11.4 / 0.0	18.6 / 0.0	22.8 / 0.0	4.8 / 0.0	8.0 / 0.0	17.9 / 0.0
R1	19.6 / 0.0	42.9 / 0.0	15.2 / 0.0	18.5 / 0.0	26.5 / 0.0	8.0 / 0.0	19.2 / 0.0	21.4 / 0.0
RFT	29.4 / 1.3	47.5 / 1.2	32.5 / 1.1	26.0 / 1.8	24.4 / 2.2	6.4 / 1.8	21.0 / 1.8	26.7 / 1.6
Search-GRPO	33.7 / 1.4	52.1 / 1.1	34.2 / 1.1	27.1 / 1.6	26.5 / 1.6	11.8 / 1.7	**23.2** / 1.5	29.8 / 1.4
CRRL	**34.7 / 1.1**	**54.5 / 1.1**	**36.8 / 1.0**	**29.0 / 1.4**	**30.0 / 1.5**	**12.4 / 1.6**	**23.2 / 1.4**	**31.5 / 1.3**

Table 3: Performance results of EM/ART across QA test sets on several baselines as well as our proposed CRRL method, measured by Exact Match (EM) and Average Retrieval Times (ART).

$\mathcal{D}_t \cap \mathcal{D}_d = \varnothing$. Evaluation is conducted on seven QA datasets to assess both in-domain and out-of-domain performance. The GRPO implementation is based on Verl (Sheng et al., 2025). More implementation details can be found in Appendix D.

We employ two types of baselines as comparisons. 1) **Retrieval-free Fine-Tuning**: We train LLMs using both supervised fine-tuning **(SFT)** and **(GRPO)**-based RL (Shao et al., 2024) methods which only contain reasoning and answer steps. 2) **Retrieval-based Fine-Tuning**: To obtain the trajectories with the retriever, we utilize rejection sampling to generate several candidate responses by interacting with the search engine for each question from the training set $\mathcal{D}_t$. Then we collect paths including correct answers for rejection fine-tuning **(RFT)** (Yuan et al., 2023). We also employ the RL pipeline for multi-turn search agents in **Search-GRPO** following (Jin et al., 2025), which optimizes LLM rollouts to autonomously call the retriever when they lack relevant knowledge.

4.3 Results and Analysis

LLMs	Methods	ART	Len.	AIT
Qwen2.5-7b	**IRCoT**	2.6	2.3k	22.4s
	Search-GRPO	2.1	0.9	17.7s
	CRRL	1.6	0.7k	16.9s
Qwen2.5-3b	**IRCoT**	1.4	0.9k	7.5s
	Search-GRPO	1.4	0.7k	4.5s
	CRRL	1.2	0.5k	4.2s

Table 4: Efficiency comparisons of baselines and our proposed CRRL during inference on test sets of average retrieval times (ART), average context length (Len.), and average inference time per question (AIT).

In Table 3, experimental results demonstrate that our proposed CRRL method outperforms other baselines in EM scores, suggesting that incorporating context refinement into the RL training pipeline of search agents can effectively mitigate context interference and improve the reliability of generations. In Table 4, during the inference phase, our proposed CRRL achieves a significant reduction in both ART and Len. compared to IRCoT and Search-GRPO, which compensate for context refiner overhead arising from the extra inference costs, and result in lower AIT, suggesting that introducing a context refiner during RL training to relax contextual interference can finally improve the efficiency of the search agent in inference.

5 Conclusion

This work investigates the context interference issue on search agents across a variety of QA benchmarks. We first demonstrate that context interference, largely stemming from the latest retrieved documents, poses a key challenge to multi-turn search agents. Then, we present a distill-based context refiner to dynamically mitigate interference in multi-turn search agents and thus significantly boost both reliability and efficiency. Furthermore, we introduce context refinement into the RL training pipelines of search agents, which can further yield performance improvements. These findings highlight the importance of context refinement to mitigate context interference to construct reliable and efficient search agents, paving the way for a new paradigm of dynamic "refine context and then generate" for AI agents in future work.

Limitations

The limitations and future work of this study are listed as follows:

Task Settings This study mainly focuses on mitigating the context interference issue on search agent tasks. However, similar problems also arise in other agent settings, such as **tool use** and **planning**. The principal factors of context interference in various tasks may differ, necessitating specific mitigation strategies to further improve reliability and efficiency of AI agents.

Paradigm Design The context refiner proposed in this study is implemented as an auxiliary module to the search agent, rather than being integrated into the training pipelines of agents. In future work, we plan to develop a dedicated training algorithm that internalizes the context refinement capability within agents themselves. This may inspire a new paradigm of "receiving observation → refining context → generating action" for agents, which can dynamically eliminate the context interference, achieving more efficient and reliable AI agents.

Acknowledgments

References

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

Eric Nuertey Coleman, Julio Hurtado, and Vincenzo Lomonaco. 2023. In-context interference in chat-based large language models. *Preprint*, arXiv:2309.12727.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Ji-Rong Wen, and Zhicheng Dou. 2025. Understand what llm needs: Dual preference alignment for retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 4206–4225, New York, NY, USA. Association for Computing Machinery.

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. *Preprint*, arXiv:2312.10997.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Akash Gupta, Ivaxi Sheth, Vyas Raina, Mark Gales, and Mario Fritz. 2024. LLM task interference: An initial study on the impact of task-switch in conversational history. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14633–14652, Miami, Florida, USA. Association for Computational Linguistics.

Muhammad Haseeb. 2025. Context engineering for multi-agent llm code assistants using elicit, notebooklm, chatgpt, and claude code. *Preprint*, arXiv:2508.08322.

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

Ziyang Huang, Xiaowei Yuan, Yiming Ju, Jun Zhao, and Kang Liu. 2025. Reinforced internal-external knowledge synergistic reasoning for efficient adaptive search agent. *Preprint*, arXiv:2505.07596.

Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. 2022. "do you follow me?": A survey of recent approaches in dialogue state tracking. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–350, Edinburgh, UK. Association for Computational Linguistics.

Xinda Jia, Jinpeng Li, Zezhong Wang, Jingjing Li, Xingshan Zeng, Yasheng Wang, Weinan Zhang, Yong Yu, and Weiwen Liu. 2025. Fast, slow, and tool-augmented thinking for llms: A review. *Preprint*, arXiv:2508.12265.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Ming Jiang, Tingting Huang, Biao Guo, Yao Lu, and Feng Zhang. 2025. Enhancing robustness in large language models: Prompting for mitigating the impact of irrelevant information. *Preprint*, arXiv:2408.10615.

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. *Preprint*, arXiv:2503.09516.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *Preprint*, arXiv:2505.06120.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 374–382, New York, NY, USA. Association for Computing Machinery.

Yubo Li, Xiaobin Shen, Xinyu Yao, Xueying Ding, Yidi Miao, Ramayya Krishnan, and Rema Padman. 2025. Beyond single-turn: A survey on multi-turn interactions with large language models. *Preprint*, arXiv:2504.04717.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Thang Nguyen, Peter Chin, and Yu-Wing Tai. 2025. Marag: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning. *Preprint*, arXiv:2505.20096.

OpenAI. 2023. Gpt-4. Https://openai.com/index/gpt-4-research/.

OpenAI. 2025. Introducing gpt-5. Https://openai.com/index/introducing-gpt-5/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, and 1 others. 2023. A study of generative large language model for medical research and healthcare. *NPJ digital medicine*, 6(1):210.

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Meghana Rajeev, Rajkumar Ramamurthy, Prapti Trivedi, Vikas Yadav, Oluwanifemi Bamgbose, Sathwik Tejaswi Madhusudan, James Zou, and Nazneen Rajani. 2025. Cats confuse reasoning llm: Query agnostic adversarial triggers for reasoning models. *Preprint*, arXiv:2503.01781.

Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part III*, page 303–313, Berlin, Heidelberg. Springer-Verlag.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023.

Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Runxin Xu Qihao Zhu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.

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. *Preprint*, arXiv:2503.05592.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023a. 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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023b. 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.

Hongru Wang, Cheng Qian, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, and Kam-Fai Wong. 2025. Toward a theory of agents as tool-use decision-makers. *Preprint*, arXiv:2506.00886.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qiming Xie, Zengzhi Wang, Yi Feng, and Rui Xia. 2024. Ask again, then fail: Large language models' vacillations in judgment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10709–10745, Bangkok, Thailand. Association for Computational Linguistics.

Guangzhi Xiong, Qiao Jin, Xiao Wang, Yin Fang, Haolin Liu, Yifan Yang, Fangyuan Chen, Zhixing Song, Dengyu Wang, Minjia Zhang, Zhiyong Lu, and Aidong Zhang. 2025. Rag-gym: Systematic optimization of language agents for retrieval-augmented generation. *Preprint*, arXiv:2502.13957.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*, page no, Kigali Rwanda. The Eleventh International Conference on Learning Representations.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. 2025. A survey on recent advances in llm-based multi-turn dialogue systems. *Preprint*, arXiv:2402.18013.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Preprint*, arXiv:2407.02485.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *Preprint*, arXiv:2308.01825.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

A Protocols

The definitions of the notations are summarized in Table 6.

Notation	Description
Internal/External Knowledge of Search Agent (Sec. 2)	
$\mathcal{M}_\theta$	LLM agent with parameter θ.
$\mathcal{K}_I$	Internal parametric knowledge of LLM.
$\mathcal{K}_E$	External knowledge of retrieved documents.
Markov Decision Process of Agent (Sec. 2)	
$\mathcal{S}$	Set of states where i-th state $\boldsymbol{s}_i \in \mathcal{S}$.
$\mathcal{A}$	Set of actions where i-th action $\boldsymbol{a}_i \in \mathcal{A}$.
$\mathcal{O}$	Set of observations where i-th observation $\boldsymbol{o}_i \in \mathcal{O}$.
$\mathcal{T}$	Set of state transition functions where $T(\boldsymbol{s}_{n+1}\|\boldsymbol{s}_n, \boldsymbol{a}_n) \in \mathcal{T}$.
$\mathcal{R}$	Reward model where $r = \mathcal{R}(\cdot)$.
$\mathcal{E}$	External retriever or search engine.
$\boldsymbol{x}$	Input query or task.
$\boldsymbol{s}_i$	i-th state including query, and previous actions and observations.
$\boldsymbol{\tau}$	Completed trajectory to the query by LLM.
$\boldsymbol{y}$	LLM generated final answer of $\boldsymbol{\tau}$.
$\hat{\boldsymbol{y}}$	Ground-truth reference answer to query $\boldsymbol{x}$.
Multi-turn Search Agent (Sec. 2)	
$\boldsymbol{K}_b$	External knowledge base.
$\boldsymbol{d}_i$	Retrieved Top-K documents $[\boldsymbol{d}_{i,1}, \ldots, \boldsymbol{d}_{i,K}]$ at round i.
$\boldsymbol{q}_i$	Search query of $\boldsymbol{a}_i$ in round i.
$\boldsymbol{t}_i$	Thinking step of $\boldsymbol{a}_i$ in round i.
Context Refiner for Search Agent (Sec. 3.3)	
$\mathcal{F}$	Context refiner.
$\tilde{\boldsymbol{d}}_i$	Refined documents with critical information extracted from $\boldsymbol{d}_i$ to $\boldsymbol{q}_{i-1}$.
$\mathcal{D}_d$	Distill dataset.
$\mathcal{D}_c$	Context refinement dataset.
$\mathcal{M}_\pi$	Foundation LLM.
$\mathcal{M}_T$	Advanced teacher model.
$\mathcal{L}_\pi^{\text{SFT}}$	Loss function to train context refiner using SFT.
Context Refinement for Search Agent Training (Sec. 4)	
$\mathcal{L}_\pi^{\text{CRRL}}$	Loss function to train policy model based on GRPO.
$\mathcal{M}_\pi$	Policy model.
$\mathcal{M}_{\text{ref}}$	Reference model.
G	Number of rollouts in one group.
ϵ	Clipping ratio.
β	Coefficient for the KL divergence.
A_j	Advantage estimate based on the group-relative rewards of the trajectory.
$\mathcal{D}_t$	Training dataset for GRPO.

Table 5: Summarized notations in this work.

B Related Work

B.1 LLM-based Search Agent

Although LLMs exhibit impressive capabilities, they often lack updated or domain-specific knowledge (Peng et al., 2023; Li et al., 2023), which undermines LLMs' reliability. Therefore, search engines (Zhao et al., 2024) are widely integrated to provide external evidence. A common paradigm is retrieval-augmented generation (RAG) (Gao et al.,

2024; Lewis et al., 2020), in which a search engine retrieves documents relevant to the search query and feeds them into LLMs. More recent work treats search engines as interactive tools (Schick et al., 2023), prompting or fine-tuning LLMs to act as search agents (Xiong et al., 2025). Approaches such as IRCoT (Trivedi et al., 2023b) and ReAct (Yao et al., 2023) employ prompting to interleave reasoning with iterative search calls. Search-R1 (Jin et al., 2025) optimizes LLMs to produce high-quality trajectories through real-time, multi-turn search interactions using RL. Despite these advances, prior studies have largely focused on eliciting LLMs to output accurate knowledge in reasoning paths despite given lengthy contexts with noise, while overlooking the contextual interference introduced by multi-turn search interactions, which may cascade across subsequent actions, degrading the reliability and efficiency of the final answers for search agents.

B.2 Contextual Interference

LLMs are highly sensitive to input contexts, leaving them vulnerable to noise or irrelevant content that can degrade output quality (Xie et al., 2024; Razavi et al., 2025). Existing strategies to mitigate this issue fall into three main categories. **(1) Key Information Extraction** identifies and preserves the most critical or relevant information in contents as in dialogue state tracking (Jacqmin et al., 2022) or RAG reranking (Glass et al., 2022; Nguyen et al., 2025; Yu et al., 2024); **(2) Compression method** condenses lengthy input sequences into summaries or latent states to weaken noise (Jiang et al., 2023; Yi et al., 2025; Li et al., 2025). **(3) Prompt-based method** explicitly instructs LLMs to disregard irrelevant content (Rajeev et al., 2025). However, Key Information Extraction relies on task-specific schema, Compression may distort key information and incur extra compression cost, and Prompt-based methods are sensitive to prompt design. More importantly, these issues are exacerbated in multi-turn search-agent scenarios, where the large volume of retrieved documents after several turns introduces more noise and irrelevant information.

B.3 Reinforcement Learning for Agent

Reinforcement Learning (RL) (Kaelbling et al., 1996) has emerged as a paradigm for LLM post-training or alignment (Ouyang et al., 2022). A variety of RL algorithms have been introduced, like PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). With specific environments and reward designs, LLMs can evolve as autonomous agents capable of adaptive decision-making and interactions with the environment. One representative application is Search Agent (Song et al., 2025; Jin et al., 2025; Chen et al., 2025), which interacts with search engines to iteratively gather external knowledge into its reasoning, and thereby performs knowledge-intensive tasks more effectively. However, current RL research focuses on deriving the optimal action from rollout trajectories while overlooking the potential influence of context interference within trajectories—particularly in search agents—thereby constraining the achievable performance of RL algorithms.

C Dataset Details

Experiments are conducted to evaluate the performance of search agents on various closed-book QA datasets, which necessitate extra retrieval to address, encompassing both single- and multi-hop scenarios. **Single-hop QA** includes: 1) **Natural Questions (NQ)** (Kwiatkowski et al., 2019), which is constructed by Google Search queries along with annotated short answers; 2) **TriviaQA** (Joshi et al., 2017), which contains closed-book trivia QA pairs to gauge models' factual knowledge; and 3) **PopQA** (Mallen et al., 2023), which consists of entity-centric QA pairs converted from a knowledge tuple retrieved in Wikidata. **Multi-hop QA** includes: 1) **HotpotQA** (Yang et al., 2018), the first large-scale dataset requiring reasoning across multiple Wikipedia paragraphs; 2) **2WikiMultiHopQA (2Wiki)** (Ho et al., 2020), which provides evidence information containing a reasoning path for multi-hop questions; 3) **MuSiQue** (Trivedi et al., 2022), which features more difficult 2-4 hop questions; and 4) **Bamboogle** (Press et al., 2023), which is made up only of complex questions that Google answers incorrectly. Dataset statistics of seven test sets are presented in Table 6.

D Training Details

For GRPO training, we set the policy LLM learning rate to 1e-6 and sample 4 responses per prompt, following the GRPO implementation in Verl (Sheng et al., 2025). The batch size is set at 32, with a mini-batch size of 8 and a micro-batch size of 4. The maximum input sequence length and generation length are set to 2048 and 500 respectively.

Dataset	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MiSique	Bamboogle
# Ques.	3610	11313	14267	7405	12576	2417	125

Table 6: Data statistics of questions in seven test sets.

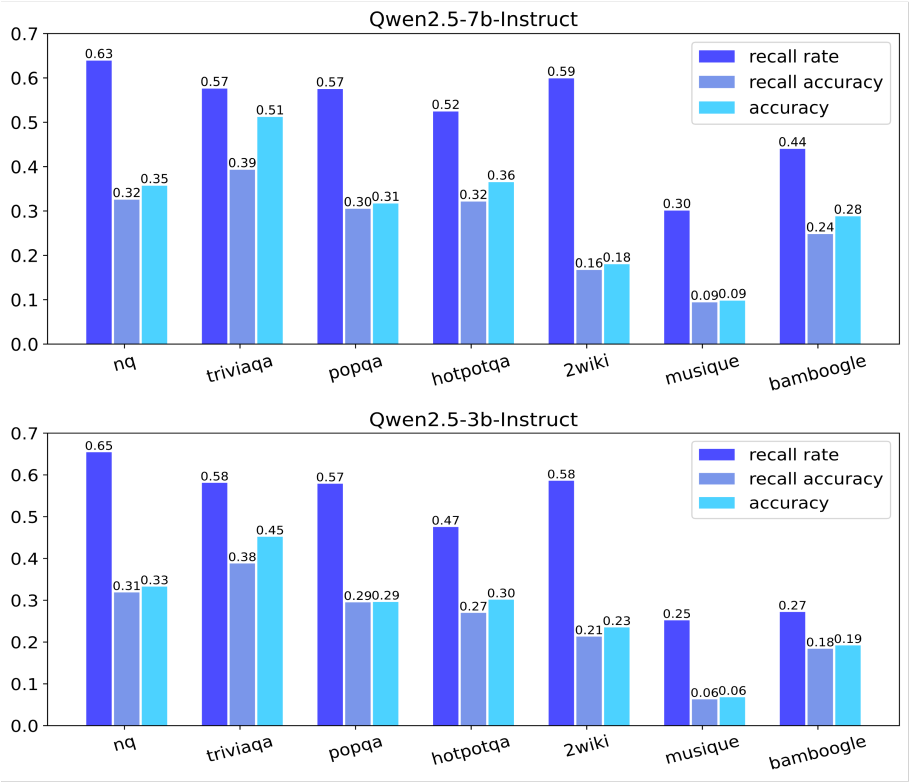

Figure 7: Demonstration of how contextual interference affects the performance of LLM search agents. "recall rate"=N_{r}/N denotes the proportion of questions for which the retrieved documents contain the correct answer (N_{r}) among all questions (N). "recall accuracy"=N_{rc}/N refers to the proportion of correctly answered questions (N_{rc}) in N_{r}, relative to all questions (N). "accuracy" represents the proportion of all correctly answered questions N_{c} out of the questions (N).

We enable gradient checkpointing and use Fully Sharded Data Parallel (FSDP) with CPU offloading. For efficient LLM rollouts, we adopt vLLM (Kwon et al., 2023) with a tensor parallel size of 1 and GPU memory utilization ratio of 0.7. The rollout sampling temperature is set to 0.7 and the top-p value to 1.0. The KL divergence regularization coefficient β and clip ratio ϵ are set to 0.001 and 0.2. The maximum action budget B is set to 8. In cases where training diverges, we evaluate at the most recent stable checkpoint according to the training reward curve; otherwise, the final checkpoint is used for evaluation.

E Prompt Template

Direct Inference Prompt

You are an excellent Question-Answering assistant. Please answer the following question based on your knowledge. You can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> North America </answer>. Question: {question}

CoT Prompt

You are an excellent Question-Answering assistant. Please answer the following question based on your knowledge. You must conduct reasoning inside <think> and </think> to think step by step first. You can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> North America </answer>. Question: {question}

IRCoT Prompt

You are an excellent Question-Answering assistant. Please answer the following question based on your knowledge. You must conduct reasoning inside <think> and </think> to think step by step first. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> and </search> and it will return the top searched results between <information> and </information>. You can search as many times as your want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> North America </answer>. Question: {question}

Question $\boldsymbol{x}$: Who is the mother of the father of George Washington?
ID 1: Generation using full documents with contextual interference.
Generation Round 1 $[\boldsymbol{a}_0, \boldsymbol{o}_1]$: <think> To find out who the mother of the father of George Washington is, I need to trace back the family lineage of George Washington. </think> <search> Who is the mother of the father of George Washington? </search> <information> **Doc 1**(Title: "Lawrence Washington (1659–1698)") Lawrence Washington (1659–1698) Lawrence Washington (1659 – February 1698), a colonial-era American who is principally remembered as the paternal grandfather of George Washington. He was the owner of a substantial Virginia plantation that he inherited from his father, John Washington, as the firstborn son under the law of primogeniture. Washington was sent to England to finish his education. In addition to being a landowner and planter, he was a lawyer, soldier, and a politician in colonial Virginia. Lawrence Washington was the eldest son of Lt. Col. John Washington (from Essex, England) and Anne Pope. He was born in 1659, on ... **Doc 2**(Title: "Augustine Washington") Augustine Washington Augustine Washington Sr. (November 12, 1694 – April 12, 1743) was the father of the first U.S. President George Washington. He belonged to the Colony of Virginia's landed gentry and was a planter and slaveholder. Augustine Washington was born in Westmoreland County, Virginia, on November 12, 1694 to Capt. Lawrence Washington, a militia captain and a member of the Virginia House of Burgesses, and Mildred Warner. His paternal grandparents were Lt. Col. John Washington (c. 1631–1677) and his first wife, Anne Pope. Augustine was only four years old when his father died. He inherited about on Bridges Creek ... **Doc 3**(Title: "Lawrence Washington (1602–1653)") Lawrence Washington (1602–1653) Rev. Lawrence Washington (1602 – 21 January 1653) was an English rector, and the great-great-grandfather of George Washington. Washington was born in 1602. He was the fifth son of Lawrence Washington (1565–1616) of Sulgrave Manor, Northamptonshire, son and heir of Robert Washington (1544–1619) esquire, of Sulgrave by his first wife Elizabeth Lyte, daughter and heiress of Walter Lyte of Radway, Warwickshire. His mother was Margaret Butler (d. 16 March 1651), the eldest daughter and co-heiress of William Butler, esquire, of Tyes Hall in Cuckfield, Sussex, and Margaret Greeke, the daughter of Thomas Greeke, gentleman, of Palsters, Lancashire. </information> **Generation Round 2** $[\boldsymbol{a}_1]$: <answer> Anne Popej </answer>
ID 2: Generation using our proposed.
Generation Round 1 $[\boldsymbol{a}_0, \boldsymbol{o}_1]$: <think> To solve this problem, I need to trace back the lineage of George Washington's family. George Washington was the first President of the United States. His father was named Augustine Washington. So, my next step would be to identify who Augustine Washington's mother was. </think> <search> Who was the mother of George Washington's father Augustine Washington? </search> <information> Augustine Washington, the father of George Washington, was born to Capt. Lawrence Washington and Mildred Warner. Therefore, the mother of Augustine Washington was Mildred Warner. </information> **Generation Round 2** $[\boldsymbol{a}_1]$: <answer> Mildred Warner </answer>

Table 7: Demonstrations of one generation of a search agent given original retrieved documents (**ID 1**) and extracted information (**ID 2**) respectively.

Question x: Hndrxx features a guest appearance by what Canadian singer?

ID 1: Generation using full documents with contextual interference.

Generation Round 1 $[a_0, o_1]$: <think> I need to find out which Canadian singer Hndrxx has featured in his song. </think> <search> Hndrxx featuring Canadian singer </search>
<information> **Doc 1**(Title: Hndrxx) Hndrxx Hndrxx (stylized in all caps and pronounced "Hendrix") is the sixth studio album by American rapper Future. It was released on February 24, 2017, by A1 Recordings, Epic Records, and Freebandz. It followed the release of Future's fifth eponymously titled album, "Future" (2017), by one week, and features guest appearances from Rihanna, The Weeknd, Chris Brown, and Nicki Minaj. "Hndrxx" is also eponymously named after Future's alter ego, Future Hendrix. "Hndrxx" was supported by four singles: "Selfish", "Pie", "Incredible", and "You da Baddest". The album received positive critical reviews and charted at number one, making Future the first artist ... **Doc 2**(Title: Hndrxx) featuring Trinidadian-American rapper Nicki Minaj, was released as the album's fourth single on July 28, 2017. The song was produced by Go Grizzly and Detail. "Hndrxx" received generally positive reviews from critics. At Metacritic, which assigns a normalized rating out of 100 to reviews from mainstream publications, the album received an average score of 78, based on eight reviews. Neil Z. Yeung of AllMusic said, "Released" just a week after his self-titled fifth effort, Future's "HNDRXX" provided an introspective and confessional complement to the more extroverted "Future". Ural Garrett of "HipHopDX" said, "HNDRXX" provides a view into a modern rock ... **Doc 3**(Title: Hndrxx) included on to the tracklist on streaming services. On July 28, 2017, another song was added to the streaming tracklist, "You da Baddest". "Selfish" featuring Barbadian singer Rihanna, was released as the album's lead single on February 28, 2017. The song was produced by Detail, Major Seven and Mantra. "Pie" featuring American singer Chris Brown, was released as the second single on June 25, 2017. The song was produced by Detail and D. A. Doman. The album's third single, "Incredible", was released to rhythmic contemporary radio on July 25, 2017. The song was produced by Dre Moon. "You da Baddest" </information>

Generation Round 2 $[a_1, o_2]$: <think> From the information provided, Hndrxx features a guest appearance by Rihanna, Chris Brown, and Nicki Minaj. However, the specific Canadian singer who appears in Hndrxx is not mentioned here. I need to search again for the exact information. </think> <search> Hndrxx Canadian singer guest appearance </search>
<information> **Doc 1**(Title: Hndrxx) Hndrxx Hndrxx (stylized in all caps and pronounced "Hendrix") is the sixth studio album by American rapper Future. It was released on February 24, 2017, by A1 Recordings, Epic Records, and Freebandz. It followed the release of Future's fifth eponymously titled album, "Future" (2017), by one week, and features guest appearances from Rihanna, The Weeknd, Chris Brown, and Nicki Minaj. "Hndrxx" is also eponymously named after Future's alter ego, Future Hendrix. "Hndrxx" was supported by four singles: "Selfish", "Pie", "Incredible", and "You da Baddest". The album received positive critical reviews and charted at number one, making Future the first artist ... **Doc 2**(Title: Hndrxx) included on to the tracklist on streaming services. On July 28, 2017, another song was added to the streaming tracklist, "You da Baddest". "Selfish" featuring Barbadian singer Rihanna, was released as the album's lead single on February 28, 2017. The song was produced by Detail, Major Seven and Mantra. "Pie" featuring American singer Chris Brown, was released as the second single on June 25, 2017. The song was produced by Detail and D. A. Doman. The album's third single, "Incredible", was released to rhythmic contemporary radio on July 25, 2017. The song was produced by Dre Moon. "You da Baddest" ... **Doc 3**(Title: Hndrxx) featuring Trinidadian-American rapper Nicki Minaj, was released as the album's fourth single on July 28, 2017. The song was produced by Go Grizzly and Detail. "Hndrxx" received generally positive reviews from critics. At Metacritic, which assigns a normalized rating out of 100 to reviews from mainstream publications, the album received an average score of 78, based on eight reviews. Neil Z. Yeung of AllMusic said, Released just a week after his self-titled fifth effort, Future's "HNDRXX" provided an introspective and confessional complement to the more extroverted "Future". Ural Garrett of "HipHopDX" said, "HNDRXX" provides a view into a modern rock </information>

Generation Round 3 $[a_2]$: <think> It seems that the search did not yield the specific information I was looking for. However, I remember from the initial searches that Hndrxx featured Rihanna, Chris Brown, and Nicki Minaj. Among these, Nicki Minaj is known to be a Canadian-American rapper. Therefore, the Canadian singer who appears on Hndrxx is likely Nicki Minaj. </think> <answer> Nicki Minaj </answer>

ID 2: Generation using our proposed.

Generation Round 1 $[a_0, o_1]$: <think> I need to know who Hndrxx is and which Canadian singer he featured with. I don't have this information, so I should search for it. </think> <search> Hndrxx Canadian singer guest appearance </search>
<information> The album Ḧndrxxb̈y Future features guest appearances from Rihanna, The Weeknd, Chris Brown, and Nicki Minaj. Chris Brown and Nicki Minaj are American singers. The Weeknd, a Canadian singer, is one of the guest artists on the album. </information>

Generation Round 2 $[a_1]$: <think> I found that the Canadian singer The Weeknd is a guest artist on Hndrxx's album. </think> <answer> The Weeknd </answer>

Table 8: Demonstrations of one generation of a search agent given original retrieved documents (**ID 1**) and extracted information (**ID 2**) respectively.