Augmenting LLM Reasoning with Dynamic Notes Writing for Complex QA

Anonymous ACL submission

Abstract

Iterative RAG for multi-hop question answering faces challenges with lengthy contexts and the buildup of irrelevant information. This hinders a model's capacity to process and reason over retrieved content and limits performance. While recent methods focus on compressing retrieved information, they are either restricted to single-round RAG, require finetuning or lack scalability in iterative RAG. To address these challenges, we propose *NotesWriting*, a method that generates concise and relevant notes from retrieved documents at each step, thereby reducing noise and retaining only essential information. This indirectly increases the *effective context length* of Large Language Models (LLMs), enabling them to reason and plan more effectively while processing larger volumes of input text. *NotesWriting* is framework agnostic and can be integrated with different iterative RAG methods. We demonstrate its effectiveness with three iterative RAG methods, across two models and four evaluation datasets. *NotesWriting* yields an average improvement of 15.6 percentage points overall, with minimal increase in output tokens.

1 Introduction

The retrieval augmented generation (RAG) paradigm has advanced open domain question answering (Zhang et al., 2022; Kamalloo et al., 2023) by incorporating external knowledge (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022; Shi et al., 2023b; Izacard et al., 2023), enabling Large Language Models (LLMs) (Hurst et al., 2024; Dubey et al., 2024) to refresh outdated parametric knowledge (Dhingra et al., 2022; Kasai et al., 2023; Vu et al., 2023) and mitigate hallucinations (Ji et al., 2023; Zhang et al., 2023).

However, for tasks like multi hop question answering (Yang et al., 2018; Zhu et al., 2024; Krishna et al., 2024) which requires reasoning over multiple documents, a single-round RAG based solely on the initial question often falls short, as it fails to capture all the necessary information. To overcome this, iterative RAG methods such as IR-CoT (Trivedi et al., 2022), FLARE (Jiang et al., 2023), and ReAcT (Yao et al., 2023) interleave retrieval and reasoning over multiple steps, progressively accumulating the evidence needed to answer complex queries.

Nevertheless, retrieved information can be noisy, and prior work has shown that excessive noise in the retrieved context can significantly degrade RAG performance (Petroni et al., 2020; Shi et al., 2023a; Zhang et al., 2024; Leng et al., 2024; Wu et al., 2024). This challenge is amplified in iterative retrieval settings, where new information must be retrieved at each reasoning step. Therefore, simply concatenating all retrieved documents at each step leads to several problems:

- **Context Overload:** Exceeding the LLM's context window limit (Krishna et al., 2024).
- **Computational Cost & Scalabiity:** Increasing processing time and resources (Yue et al., 2024).
- **Distraction:** Including irrelevant or redundant information that hinders the LLM's reasoning and planning ability (Yu et al., 2023; Chen et al., 2024; Xie et al., 2024; Aghzal et al., 2025).
- **Readability:** Excessively long reasoning traces created from multiple documents pose challenges for users to interpret precise reasoning. Redundant information can further affect readability.

To address these issues, we propose a simple yet effective and scalable method called *NotesWriting*. At each retrieval step, *NotesWriting* produces concise notes based upon the retrieved documents and the sub-question, thus providing only the essential information required at each step. This increases the *effective context length* as it does not overload the LLM context with irrelevant information, which helps the LLMs plan & reason better. Furthermore, *NotesWriting* is generic and can be coupled with any iterative RAG framework. While recent meth-

ods (Edge et al., 2024; Xu et al., 2023a; Kim et al., 2024) have explored summarizing retrieved content, they are often limited to single-round RAG or require synthetic data for fine-tuning. Jiang et al. (2025) extend this idea by summarizing retrieved documents at each step; however, this approach lacks scalability as it is still limited to three iterations at maximum and depends on multiple modules leading to multiple LLM calls in each iteration. This work makes the following key contributions:

- We propose *NotesWriting* to improve effective context length of iterative RAG. This reduces context overload, the number of reasoning steps, and redundancy.
- *NotesWriting* is plug-and-play, and can be coupled with any iterative RAG framework, benefiting planning and reasoning abilities by reducing tokens in context (thus indirectly increasing the effective context length per step) and reducing retention of irrelevant information.
- Our experiments across *three* iterative RAG baselines (IRCoT, FLARE & ReAct), *four* multi-hop QA datasets and *two* LLMs demonstrates that *NotesWriting* achieves 15.6 percentage points improvement by increasing the volume of ingested text with minimal increase in output tokens.
- *NotesWriting* with ReAct (ReNAct) achieves the highest performance, enabling better planning by guiding the model to generate more accurate search queries and retrieving correct documents as demonstrated in Section 6.

2 Background and Related Work

Single-Step vs. Iterative RAG. Traditional RAG often operates in a single step: retrieve relevant documents based on the initial query, then generate the final response conditioned on both the query and the retrieved context. While effective for simpler questions, this retrieve-then-read approach struggles for multi-hop QA, where the information needed evolves throughout the reasoning process. Iterative RAG addresses this limitation by interleaving retrieval and generation. The model can issue multiple queries, gather information incrementally, and refine its reasoning path based on newly retrieved evidence. This dynamic interaction between the LLM and the retriever is better suited for complex, multi-step reasoning.

Formulation of Iterative RAG. Let $\mathbf{x}$ be the user input question, and $\mathcal{D} = \{d_i\}_{i=1}^{|\mathcal{D}|}$ represent the external knowledge corpus (e.g., Wikipedia). An iterative RAG process aims to generate a sequence of reasoning steps or partial outputs $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n]$. We denote the language model as $\text{LM}(\cdot)$ and the retrieval function, which returns the top-k documents for a query $\mathbf{q}$, as $\text{ret}(\mathbf{q})$.

At each step $t \geq 1$, the typical process involves:

1. **Query Formulation:** A query $\mathbf{q}_t$ is generated based on the initial input $\mathbf{x}$ and the preceding steps $\mathbf{s}_{<t} = [\mathbf{s}_1, \ldots, \mathbf{s}_{t-1}]$. This is governed by a query formulation function $\mathcal{Q}(\cdot)$:
$$\mathbf{q}_t = \mathcal{Q}(\mathbf{x}, \mathbf{s}_{<t}) \quad (1)$$
For the first step, $\mathbf{s}_{<1} = \emptyset$, and often $\mathbf{q}_1 = \mathbf{x}$.
2. **Retrieval:** The retriever fetches the top-k relevant documents: $\mathcal{D}_{\mathbf{q}_t} = \text{ret}(\mathbf{q}_t)$.
3. **Generation:** The LM generates the next reasoning step $\mathbf{s}_t$ using the original input, previous steps, and the newly retrieved documents:
$$\mathbf{s}_t = \text{LM}([\mathcal{D}_{\mathbf{q}_t}, \mathbf{x}, \mathbf{s}_{<t}]) \quad (2)$$

This process continues until a final answer is generated, or a maximum number of steps is reached.

Advances in Iterative RAG. Several approaches have explored different strategies within this iterative framework: IRCoT (Interleaving Retrieval and Chain-of-Thought) (Trivedi et al., 2022) directly uses the last generated sentence of the Chain-of-Thought (CoT) (Wei et al., 2022) reasoning as the query for the next retrieval step. This tightly couples reasoning progress with information gathering. Formally, its query function is:
$$\mathbf{q}_t = \mathcal{Q}_{\text{IRCoT}}(\mathbf{x}, \mathbf{s}_{<t}) = \mathbf{s}_{t-1} \quad (3)$$
IRCoT demonstrates improved performance over single-step RAG across various QA datasets, showing robustness even with smaller models.

FLARE (Forward-Looking Active Retrieval Augmented Generation) (Jiang et al., 2023) builds on similar ideas but retrieves proactively. It anticipates future content needs by checking the confidence of generated tokens. Retrieval is triggered only when the model generates low-confidence tokens (often indicating a need for information), using the preceding sentence as the query. If confidence is high, retrieval is skipped. The query function is conditional:
$$\mathbf{q}_t = \begin{cases} \mathbf{s}_{t-1}, & \text{if } \text{conf}(\mathbf{s}_{t-1}) > \theta \\ \emptyset, & \text{otherwise} \end{cases} \quad (4)$$
While effective in long-form generation, its reliance on predicting future needs might face challenges if the context shifts unexpectedly.

ReAct (Reason and Action) (Yao et al., 2023) ex-

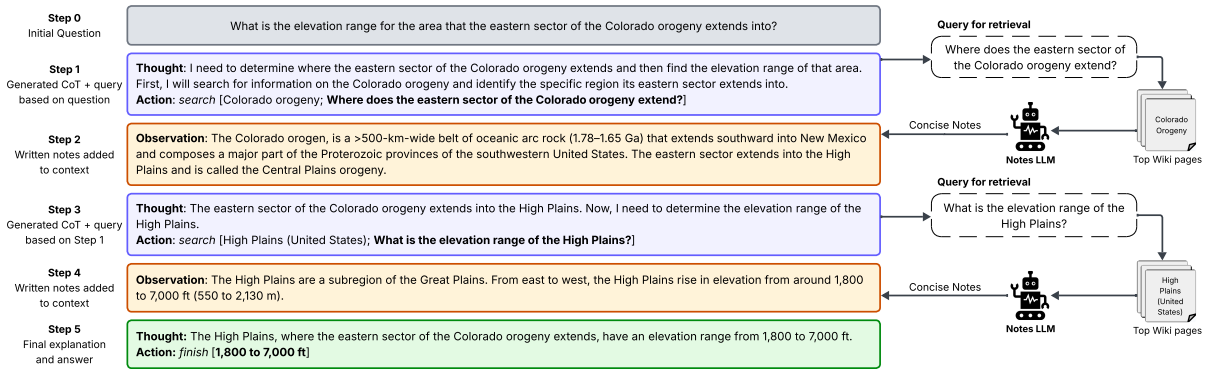

Figure 1: Overview of *NotesWriting* within an iterative RAG framework.

plicitly separates reasoning (`Thought`) from information gathering (`Action`), where the action often involves generating a specific search query.

$$\mathbf{q}_t = \mathcal{Q}_{\text{ReAct}}(\mathbf{x}, \mathbf{s}_{<t}) = \texttt{Action}_t \quad (5)$$

Context Management in Iterative RAG. Despite advancements in iterative RAG, a core challenge persists: managing the retrieved context effectively across iterations. Even with long context LLMs, studies have found that complex tasks which require compositional reasoning like multi-hop QA are solved better with retrieval (Xu et al., 2023b; Lee et al., 2024a). However, long context LLMs have been shown to face issues in handling information within the long context (needle-in-the-haystack issues) (Kamradt, 2023; Hsieh et al., 2024) which limit performance even when combined with RAG (Jiang et al., 2024). Thus, addressing context management requires mechanisms to condense, filter, or summarize the retrieved information. Several such approaches have been explored in recent research. RECOMP (Xu et al., 2023a) compresses retrieved documents using extractive or abstractive summarization before passing them to the main LLM in a single-turn RAG setting. This helps with query-relevant compression, but does not directly handle iterative context accumulation. Chain-of-Note (CON) (Yu et al., 2023) generates sequential notes during training to assess retrieved document relevance and reliability. This improves robustness against noise, but lacks explicit planning or iterative refinement at inference time. PlanRAG (Lee et al., 2024b) proposes a two-stage approach, generating a decision plan and then executing retrieval operations by adding Plan and Re-plan steps to ReAct. SmartRAG (Gao et al., 2024) similarly includes a policy network which decides whether to retrieve, and a retriever, which are jointly optimized to reduce retrieval while improving performance. However, retrieved documents while more relevant can still accumulate in context, affecting performance. Self-RAG (Asai et al., 2023) uses reflection tokens to self-reflect on the retrieved documents as a means to reduce the number of documents included in the context. Self-reflection is achieved by fine-tuning the model, which improves factual accuracy and citation integrity when benchmarked against existing models such as ChatGPT and Llama2-Chat. However, the requirement of fine-tuning could be costly, and require updates over time.

More recently, modular approaches have been suggested for iterative retrieval and summarization. Infogent (Reddy et al., 2024) proposes two modules, an Aggregator whose textual feedback guides further retrieval from a Navigator-Extractor. The Extractor extracts readable relevant content from the Navigator's web-based API access and forwards it to the Aggregator for evaluation. However, context management remains an important issue. ReSP (Retrieve, Summarize, Plan) (Jiang et al., 2025) uses query-focused summarization in multi-hop QA, maintaining global and local evidence summaries across iterations to prevent context overload. It involves multiple LLM calls per iteration focusing on planning for the next step with sub-questions, summarizing retrieved documents, generating the next sub-questions, and judging for whether there is sufficient information to answer the question. While specialized modules for each stage could boost performance further, this approach faces several drawbacks - like the possibility of cascading failures if any module fails during an iteration, and multiple LLM calls which can further increase latency and information repetition with local and global evidence.

Our proposed *NotesWriting* method overcomes

the aforementioned challenges by focusing on flexibly generating concise and relevant notes from retrieved documents at each iterative step. This addresses the critical need for noise reduction and context length enhancement, thereby allowing LLMs to reason and plan more effectively in complex multi-hop scenarios.

3 Method

To address the challenges of context overload and information noise in iterative RAG, particularly for multi-hop QA, we introduce ***NotesWriting***, a method for generating concise, query-relevant notes from retrieved documents at each step. Instead of feeding raw retrieved documents to the main LM, *NotesWriting* first processes them to extract key information, thereby reducing context length and filtering irrelevant content.

3.1 *NotesWriting*: Iterative Note Extraction

The core idea is to use a dedicated, smaller language model (LM_{notes}) to act as a note-taker. At each iteration t, after retrieving the top-k documents $\mathcal{D}_{\mathbf{q}_t} = \{d_1, d_2, \ldots, d_k\}$ based on the query $\mathbf{q}_t$, *NotesWriting* performs the following:

1. **Note Extraction:** For each retrieved document d_i, LM_{notes} is prompted (using prompt $\mathcal{P}_{\text{notes}}$, see Appendix A.6) to extract concise notes r_i relevant to the current query $\mathbf{q}_t$:
$$r_i = \text{LM}_{\text{notes}}(\mathbf{q}_t, d_i) \tag{6}$$
2. **Note Aggregation:** The extracted notes from all k documents are aggregated as $\mathbf{O}_t$:
$$\mathbf{O}_t = \bigcup_{i=1}^{k} r_i \tag{7}$$

This process replaces the direct feeding of potentially long and noisy documents $\mathcal{D}_{\mathbf{q}_t}$ with the much shorter and focused notes $\mathbf{O}_t$.

3.2 ReNAct: ReAct with *NotesWriting*

While *NotesWriting* is a generic module that can be integrated with different iterative RAG methods, results in Section 5 demonstrates that it works best with the ReAct framework (Yao et al., 2023). Therefore, we propose leveraging the ReAct framework as a suitable base for our approach. ReAct's structure explicitly separates reasoning (`Thought`) from information gathering (`Action`), where the action often involves generating a specific search query. This explicit query generation aligns naturally with the goal of targeted retrieval followed by focused note-taking.

We combine ReAct with *NotesWriting* with the process at step t as follows:

1. LM generates `Thought` step outlining reasoning, along with an `Action` step, typically containing a search query. This query becomes $\mathbf{q}_t$:
$$\begin{aligned}\mathbf{q}_t =& \mathcal{Q}_{\text{ReAct}}(\mathbf{x}, \mathbf{s}_{<t}) = \\ & \text{SearchQueryFrom}(\texttt{Action}_t)\end{aligned} \tag{8}$$
Retrieval is performed using $\mathbf{q}_t$ to get $\mathcal{D}_{\mathbf{q}_t}$.
2. *NotesWriting* processes $\mathcal{D}_{\mathbf{q}_t}$ using LM_{notes} to generate aggregated notes, presented as the observation $\mathbf{O}_t$.
3. The main LM receives $\mathbf{O}_t$ and uses it along with $\mathbf{x}$ and $\mathbf{s}_{<t}$ to generate the next `Thought` and `Action` pair:
$$\begin{aligned}\mathbf{s}_t(\text{next Thought+Action}) = \\ \text{LM}([\text{Observation: } \mathbf{O}_t, \mathbf{x}, \mathbf{s}_{<t}])\end{aligned} \tag{9}$$

Iterations continue until the model generates a final answer within $\mathbf{s}_t$ or reaches a maximum number of iterations T, after which a final answer is synthesized based on the full history $\mathbf{s}$ and the collected notes $\{\mathbf{O}_t\}_{t=1}^{T}$. This approach (illustrated in Figure 1) aims to combine the structured reasoning of ReAct with the context management benefits of *NotesWriting*, leading to a more robust and efficient iterative RAG system for complex QA.

3.3 *NotesWriting*: A Plug-and-Play Module for Iterative RAG

NotesWriting is designed as a complementary module that can be integrated into various iterative RAG frameworks. It modifies the generation step (Eq. 5) while keeping the specific query formulation $\mathcal{Q}$ of the base framework. We demonstrate this integration with two SOTA iterative RAG frameworks: IRCoT and FLARE.

IRCoT: Query remains the last generated sentence ($\mathbf{q}_t = \mathbf{s}_{t-1}$). Generation step becomes:
$$\begin{aligned}\mathbf{s}_t =& \text{LM}([\mathbf{O}_t, \mathbf{x}, \mathbf{s}_{<t}]) \text{ where} \\ & \mathbf{O}_t \text{ is derived from } \mathcal{D}_{\mathbf{q}_t} = \text{ret}(\mathbf{s}_{t-1})\end{aligned} \tag{10}$$

FLARE: Query formulation remains conditional based on confidence θ ($\mathbf{q}_t = \mathcal{Q}_{\text{FLARE}}(\mathbf{x}, \mathbf{s}_{<t})$). If retrieval occurs ($\mathbf{q}_t \neq \emptyset$), the generation step uses the extracted notes:
$$\begin{aligned}\mathbf{s}_t =& \text{LM}([\mathbf{O}_t, \mathbf{x}, \mathbf{s}_{<t}]) \\ & \text{where } \mathbf{O}_t \text{ is derived from } \mathcal{D}_{\mathbf{q}_t}\end{aligned} \tag{11}$$

If retrieval is skipped ($\mathbf{q}_t = \emptyset$), $\mathbf{O}_t = \emptyset$ and generation proceeds without new retrieved context.

Model	Dataset	Over Limit (n / %)
GPT-4o-mini (128k)	Frames	463 / 549 (84.3%)
	FanoutQA	244 / 310 (78.7%)
LLaMA 3.1 70B (64K)	Frames	488 / 549 (88.8 %)
	FanoutQA	255 / 310 (82.2%)

Table 1: Number of questions exceeding LLMs context with the top-5 Wikipedia pages (markdown format) being inserted into the LLM context at each step.

4 Experiments

4.1 Datasets

(1) **FanoutQA** (Zhu et al., 2024) focuses on "fanout" multi-hop, multi-document complex questions that require gathering information about a large set of entities. We report results on the dev set containing 310 questions.
(2) **FRAMES** (Krishna et al., 2024) a challenging multi-hop QA dataset requiring 2–15 hops to answer questions. We exclude questions requiring tabular reasoning and evaluate on 549 examples.
(3) **HotpotQA** (Yang et al., 2018) a popular multi-hop QA dataset that requires reasoning over 2-3 Wikipedia articles. reasoning. We report results on 500 examples from the dev set.
(4) **MultiHop-RAG** (Tang and Yang, 2024) a non-wikipedia based benchmark that involves retrieval over recent news articles. It has $\sim$ 600 news articles. For each question, we used BM25 to get the top five news articles in each iteration.

Evaluation metrics. We report the **F1 score** between predicted and ground truth answer and following (Krishna et al., 2024) we also use **GPT4-as-Judge score** with prompt in the appendix 6. We also measure the ***effective context length*** by reporting the average number of input & output tokens processed by the main LLM and notes writing LM_{notes} across all steps/iterations. Finally, we look at the **average number of steps** that is defined as the number of search queries that the main LLM needs to answer the question.

4.2 Models

We experiment with two LLMs, representing closed & open weights, `GPT-4o-mini`[1] and `Llama 3.1-70-Instruct` (Dubey et al., 2024). We set the temperature to 0.7 and use the same LLM for generating reasoning step and *NotesWriting* (i.e LM = LM_{notes}). `Llama 3.1-70-Instruct` was hosted using vLLM (Kwon et al., 2023) across 8 A100-80GB GPUs, supporting a maximum context length of $64K$. `GPT-4o-mini`, which has a context length of $128K$.

4.3 *NotesWriting* Implementation Details

For *NotesWriting*, we utilize the Wikipedia API to fetch the top 5 relevant pages based on the query $\mathbf{q}_t$. Each retrieved Wikipedia page is converted to Markdown format using `markdownify`[2] before being processed by LM_{notes}.
Initial experiments revealed that feeding the full content of the top-5 retrieved Wikipedia pages directly into the main LM (as standard iterative RAG baselines do) frequently caused context length errors, especially on challenging benchmarks like Fanout-QA and FRAMES. Table 1 shows that approximately 80% of the questions are unanswerable as the context builds up and exceeds the context window. This observation, corroborated by the large average token counts reported for baselines Tables 2, 8, and 9 (which often exceed typical LLM context limits, necessitating adjustments. Therefore, for a fair comparison, the baseline methods (IRCoT, FLARE, ReAct without *NotesWriting*) were re-implemented using a chunked document setup as detailed in Section 4.4, while *NotesWriting* operates on the full retrieved pages.

4.4 Baseline Implementation Details

ReAct — As simply passing the top retrieved documents at each step to LLM causes context length exceeded (refer Table 1), we re-implement the original baseline (Yao et al., 2023) which allows the LLM to `search` that retrieves the first paragraph of the top 5 Wikipedia pages, `select` that allows ReAct to select relevant page for first 10 passages and `lookup` that returns paragraphs containing that specific string in the selected page.
IRCoT & FLARE — These were originally designed for older completion-based models such as `text-davinci-003` which relied heavily on internal parametric knowledge to generate CoTs. However, such a design is not directly applicable to questions requiring step by step planning and up-to-date knowledge. To address this, we adapt the few-shot prompting strategy to be compatible with chat models, enabling them not only to generate CoTs but also to explicitly plan ahead (see Appendix A.6). Moreover, the original baselines used

[1] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[2] https://pypi.org/project/markdownify/

Model	Setting	Benchmark	F1 (%)	GPT-4 Score (%)	Avg steps	Main Tokens		Notes Tokens	
						Input	Output	Input	Output
GPT-4o-mini	ReAct	Fanout-QA	28.6	12.9	10.4	116K	916	-	-
		Frames	8.7	31.1	8.36	67K	707	-	-
		Hotpot-QA	42.2	56.4	3.33	26K	**319**	-	-
		MultiHop-RAG	**58.0**	64.2	**5.6**	188K	**278**	-	-
	ReAct + *NotesWriting* (ReNAct)	Fanout-QA	**50.0**	**28.0**	**7.8**	**17K**	**598**	359K	675
		Frames	**46.8**	**52.3**	**6.51**	**16K**	**543**	277K	607
		Hotpot-QA	**51.0**	**64.0**	**3.2**	**9K**	326	130K	321
		MultiHop-RAG	**58.0**	**70.6**	6.0	**46K**	368	68K	390
LLaMA-3.1-70B	ReAct	Fanout-QA	13.5	8.7	7.5	113K	506	-	-
		Frames	21.7	26.8	6.83	85K	433	-	-
		Hotpot-QA	43.7	52.6	4.3	49K	289	-	-
		MultiHop-RAG	53.6	61.4	**5.24**	180K	295	-	-
	ReAct+ *NotesWriting* (ReNAct)	Fanout-QA	**43.0**	**26.1**	**5.80**	**15K**	**485**	265K	1116
		Frames	**49.0**	**57.6**	**4.81**	**13K**	**412**	193K	717
		Hotpot-QA	**55.5**	**67.4**	**3.34**	**8K**	**274**	109K	391
		MultiHop-RAG	**63.5**	**73.0**	5.9	**47K**	**262**	76K	425

Table 2: ReAct and *NotesWriting* results for GPT-4o-mini and LLaMA. Main tokens represent the total number of input & output tokens for the main LLM across all steps (average on all questions). Similarly, notes tokens represent the total number of input & output tokens across all steps by the notes writing LLM (averaged on all questions). Token counts are rounded to the nearest thousand.

BM25 from an older Wikipedia dump. However, in initial experiments we observed that the older dump is outdated for latest datasets. Therefore, we used a recent dump `20231101.en`[3] and dense passage retrieval with `ef-base-v2` embeddings (Wang et al., 2022). We set the selective retrieval parameter θ to 0.8 for all our experiments.[4]

5 Results

Enhanced performance with *NotesWriting*. From Table 2, 3 and 4 in comparison to the respective baselines, *NotesWriting* shows significant improvements across all models and benchmarks. Specifically from Table 2, on complex long-form multihop-QA datasets like *FRAMES* and *Fanout-QA*, on average ReNAct achieves an absolute improvement of 29.1 points in F1 score and 21.1 points in GPT-4 score. On relatively easier datasets such as *Hotpot-QA* and *MultiHop-RAG*, ReNAcT yields absolute improvements of 10.3 and 5.0 points, respectively. The strong results compared to the baseline demonstrate that the LLM is receiving correct and relevant information at each step with *NotesWriting*.

From Tables 3 and 4 on challenging datasets *NotesWriting* coupled with each of IRCoT and FLARE leads to 14.4 and 10.5 points improvement on F1 and GPT-4 score. Similarly on Hotpot-QA and MultiHop-RAG we find 7.0 and 10.8 points improvement on F1 and GPT-4 score respectively.

Model	Setting	Benchmark	F1 (%)	GPT-4 (%)
GPT-4o-mini	IRCoT	FanoutQA	33.6	15.2
		Frames	24.0	22.0
		HotpotQA	36.4	42.8
		M-RAG	23.7	48.0
	IRCoT +*Notes Writing*	FanoutQA	**41.9**	**21.3**
		Frames	**43.9**	**42.3**
		HotpotQA	**46.2**	**53.8**
		M-RAG	**36.0**	**65.6**
LLaMA-3.1-70B	IRCoT	FanoutQA	21.0	8.4
		Frames	19.4	21.1
		HotpotQA	31.5	38.2
		M-RAG	35.4	64.8
	IRCoT +*Notes Writing*	FanoutQA	**36.0**	**22.9**
		Frames	**26.9**	**33.3**
		HotpotQA	**36.5**	**53.0**
		M-RAG	**38.0**	**64.8**

Table 3: IRCoT performance for GPT-4o-mini and LLaMA-3.1-70B M-RAG represents MultiHop-RAG.

Increased effective context length. Tables 2, 8 and 9 show the average number of input and output tokens across all steps for the baseline and *NotesWriting*. The total number of tokens processed by the system (sum of input tokens across main and notes-writing LLMs) increases, allowing the model to reason over more retrieved content. However, it is important to note that this information cannot be naively appended to the main

[3] https://huggingface.co/datasets/wikimedia/wikipedia

[4] We also compare *NotesWriting* with Infogent (Reddy et al., 2024) with details and results in Appendix A.1

LLM's context (summing columns 7 and 9 would exceed the context window). This demonstrates that *NotesWriting* enables scalable use of large retrieval context by delegating information management to a specialized LLM.

With ReAct (Table 2), the number of tokens for the main LLM reduce significantly from baseline to ReNAct across all benchmarks — by 77K tokens for GPT-4o-mini and 86K tokens for LLaMA-3.1-70B on average. This demonstrates concise notes being added at each retrieval step. Similarly, output tokens decrease across benchmarks, with an average reduction of 96 tokens for GPT-4o-mini and 53 tokens for LLaMA-3.1-70B. The same trend is observed with IRCoT (Table 8) and FLARE (Table 9) where the main LLM input tokens reduces by at least 4x and 1.5x for GPT-4o-mini and LLaMA-3.1-70B, with output tokens being almost comparable.

Reduced average steps. ReNAct reduces the average number of steps across all the benchmarks *MultiHop-RAG* (Table 2). For *Frames* and *Fanout-QA*, the reduction is 2.23 & 1.86 for GPT-4o-mini and LLaMA-3.1-70B respectively. The reduction is smaller but still present for *Hotpot-QA* with an average drop of 0.13 and 0.96 steps respectively. We further analyze the reduction in redundant queries and correlation with ground truth steps in Section 6.

***NotesWriting* is cost effective.** Tables 2, 8, and 9 show that with *NotesWriting* the combined output tokens by the main & note taking LLM are on an average 2-3x more than the baselines. However, this tradeoff is justified by the performance improvement and the output tokens being significantly less (about 100x) than effective number of input tokens. As the output tokens are the major contributing factor to cost [5] & latency, *NotesWriting* is a much more cost + compute + performance effective approach.

6 Analysis

Reasoning Quality Analysis. We evaluate the reasoning chains generated by ReAct and ReNAct using GPT-4o as a judge across three axes, (1) Efficiency — to measure redundant searches and how well each step contributes to the final answer, (2) Redundancy — to assess repeated search queries, or unnecessary repetition or duplication of steps (3) Coherence — to check if the chain is comprehensible, logically connected, and free from unnecessary complexity or ambiguity. The evaluation prompt is in Appendix 7. Figure 2 shows the results. ReNAct is better across all three axes than ReAct across on all models and datasets. Specifically, on Frames and FanoutQA across both models efficiency, redundancy and coherence improve by at least 1.5x. On HotpotQA, the improvement is 1.2x.

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

Model	Setting	Benchmark	F1 (%)	GPT-4 (%)
GPT-4o-mini	FLARE	Fanout-QA	35.1	14.2
		Frames	26.3	23.7
		HotpotQA	34.8	39.0
		M-RAG	28.9	65.7
	FLARE +*Notes Writing*	Fanout-QA	**42.3**	**22.2**
		Frames	**27.7**	**29.8**
		HotpotQA	**34.5**	**45.8**
		M-RAG	**30.2**	**66.6**
LLaMA-3.1-70B	FLARE	FanoutQA	23.0	11.4
		Frames	16.4	18.6
		Hotpot-QA	24.7	31.2
		M-RAG	36.1	67.0
	FLARE +*Notes Writing*	FanoutQA	**35.8**	**24.2**
		Frames	**20.0**	**25.3**
		HotpotQA	**34.0**	**47.0**
		M-RAG	**30.5**	**66.4**

Table 4: FLARE performance for GPT-4o-mini and LLaMA-3.1-70B. M-RAG represents MultiHop-RAG.

Model	Dataset	ReNAct	ReAct
GPT-4o-mini	Fanout	**22.83**	36.84
	Frames	**23.18**	29.89
	HotpotQA	**30.56**	46.02
LLaMA	Fanout	49.44	**35.71**
	Frames	**28.22**	30.94
	HotpotQA	**31.33**	32.82

Table 5: % of correct questions having search steps less than number of ground truth Wikipedia pages.

Search Steps Comparison. Figure 3 shows the comparison of the number of ground truth steps, ReAct and ReNAct search steps for each question in each dataset across both models. The dashed lines for each method represents the in-correct answers and the sold line represents correct ones. The x-axis is the index of the question in the dataset sorted by the number of ground-truth search steps. From the Figure 3, it can be observed that ReNAct (solid blue line) is much closer to the ground truth steps with ReAct (solid red line) being relatively far demonstrating the effectiveness of *NotesWriting* in coming up with correct stepwise plan and search query for retrieval. Figure 3 also shows that the in-correct questions (dashed red & blue line) have a higher number of steps that shows that it fails

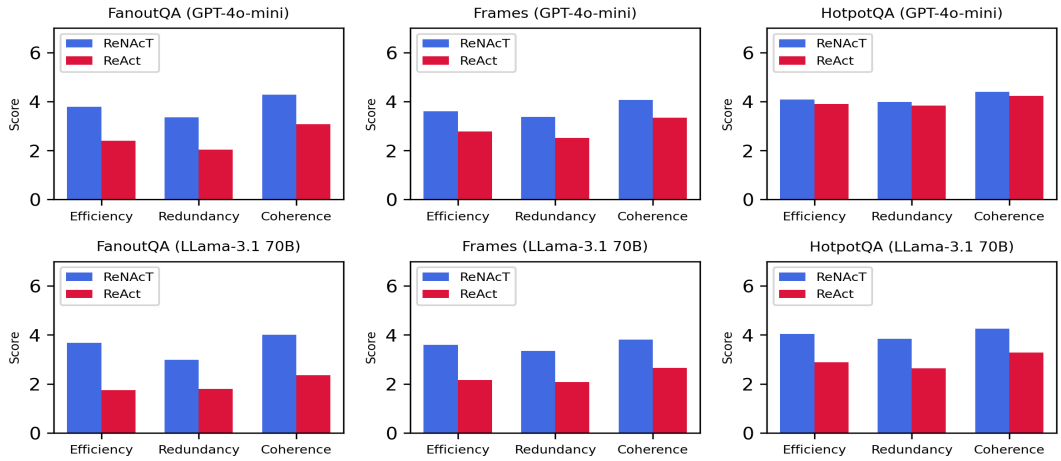

Figure 2: Quality evaluation of ReAct and ReNAct reasoning chain.

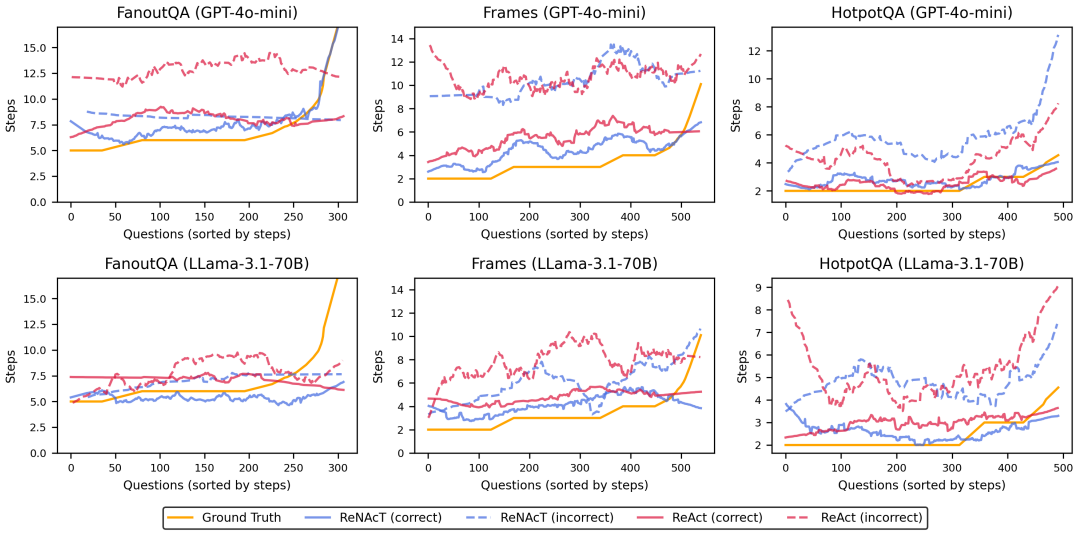

Figure 3: Steps (smoothed) by ReNAct, ReAct vs the ground truth steps for GPT-4o-mini and LLama-3.1-70B.

after many re-tries. The difference among ReNAct ReAct and ground truth steps is more significant in challenging datasets like Frames and Fanout-QA as opposed to HotpotQA.

Correct Answers with Fewer Searches than Ground Truth. Figure 3 shows cases where ReNAct and ReAct (solid blue & red lines) are below ground truth steps. Table 5 reports percentages of cases for the number of correctly answered questions which took less searches than the number of ground truth Wikipedia pages required to answer the question correctly.

7 Conclusion

We presented *NotesWriting*, a plug-and-play module that improves *effective context length* in iterative RAG by accumulating only the most relevant information at each reasoning step. Experiments on *three* RAG baselines (IRCoT, FLARE, and ReAct), *four* multi-hop QA datasets, and *two* LLMs show that *NotesWriting* improves performance by up to 15.6 points, while also reducing context overload, the number of reasoning steps, and redundancy. In the ReAct setting, *NotesWriting* enables better planning by guiding the model to generate more accurate search queries and retrieve the correct documents. Moreover, *NotesWriting* consistently improves coherence and efficiency of planning and search across models in ReAct. Therefore, we suggest ReNAct as an effective iterative RAG framework. Our results show that ReNAct (ReAct + *NotesWriting*) makes iterative RAG more scalable and precise.

Limitations and Societal Impact

Our approach has several limitations. First, our experiments are limited to the two models we experiment with, which could be extended to newer smaller open-source models. Second, we limit online searches to the Wikipedia API[6], which only supports searching for text matching Wiki pages; and third, Wiki pages change often and this could lead to a mismatch with static benchmarks' ground truth. While these could affect performance, we ensure that the same setup is also followed in all baselines we experiment with, to keep evaluation comparable while reducing the need to utilize paid search APIs. Third, with retrievals based on iterative notes writing, there is a possibility of conflicting information being received (Table 18). It is possible that the model starts hallucinating facts, and this remains a weakness at large. Lastly, we impose a maximum iteration limit to ensure computational efficiency, which could also impact performance. Further explorations towards improving on weaknesses remain future work.

Potential risks of our work include usage in scenarios where the requested retrieval information is toxic or harmful. While we cannot control how our method is used for prompting, we expect content moderation policies to help with reducing the impact of such queries. Moreover, hallucinations 19, 20 can affect the QA experience, although manual observation of the reasoning traces show that recovery can be better with *NotesWriting*.

We expect our work to significantly enhance the QA user experience, as focused information improves performance and reduced context lengths lower computational costs. We hope our *NotesWriting* method can contribute towards better task handling at large. We will make our code publicly available upon acceptance towards this goal.

References

Mohamed Aghzal, Erion Plaku, Gregory J Stein, and Ziyu Yao. 2025. A survey on large language models for automated planning. *arXiv preprint arXiv:2502.12435*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Yanan Chen, Ali Pesaranghader, Tanmana Sadhu, and Dong Hoon Yi. 2024. Can we rely on llm agents to draft long-horizon plans? let's take travelplanner as an example. *arXiv preprint arXiv:2408.06318*.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

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

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Jingsheng Gao, Linxu Li, Weiyuan Li, Yuzhuo Fu, and Bin Dai. 2024. Smartrag: Jointly learn rag-related tasks from the environment feedback. *arXiv preprint arXiv:2410.18141*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

[6] https://www.mediawiki.org/wiki/API:Search

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2025. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. *Preprint*, arXiv:2407.13101.

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

Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *arXiv preprint arXiv:2305.06984*.

Greg Kamradt. 2023. Needle in a haystack-pressure testing llms. *Github Repository*, page 28.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. *arXiv preprint arXiv:2404.13081*.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2024. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *arXiv preprint arXiv:2409.12941*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

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

Myeonghwa Lee, Seonho An, and Min-Soo Kim. 2024b. Planrag: A plan-then-retrieval augmented generation for generative large language models as decision makers. *arXiv preprint arXiv:2406.12430*.

Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. 2024. Long context rag performance of large language models. *arXiv preprint arXiv:2411.03538*.

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

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*.

Revanth Gangi Reddy, Sagnik Mukherjee, Jeonghwan Kim, Zhenhailong Wang, Dilek Hakkani-Tur, and Heng Ji. 2024. Infogent: An agent-based framework for web information aggregation. *arXiv preprint arXiv:2410.19054*.

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

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. 2023. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

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

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? *arXiv preprint arXiv:2404.03302*.

Jian Xie, Kexun Zhang, Jiangjie Chen, Siyu Yuan, Kai Zhang, Yikai Zhang, Lei Li, and Yanghua Xiao. 2024. Revealing the barriers of language agents in planning. *arXiv preprint arXiv:2410.12409*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023a. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023b. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

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 *International Conference on Learning Representations (ICLR)*.

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

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2024. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*.

Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. 2022. A survey for efficient open domain question answering. *arXiv preprint arXiv:2211.07886*.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. In *First Conference on Language Modeling*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 Infogent Implementation Details

We use the official code provided by Infogent authors here (Apache 2.0. License) with the following modifications:

- Due to our limitations in accessing OpenAI, we modify the code to use AzureOpenAI.
- OpenAI embedding is replaced by sentence-transformers' `all-mpnet-base-v2`[7].
- Serper Google Search[8] is replaced by Wikipedia search API due to credit limitations and to use similar open knowledge tools as those used in our method, reducing the cost needed to conduct RAG experiments.

A.2 Results

Setting	Benchmark	F1 (%)	GPT-4 (%)
InfoAgent	FanoutQA	47.2	22.9
	Frames	28.0	29.9
NotesWriting	FanoutQA	**50.0**	**28.0**
	Frames	**46.8**	**52.3**

Table 6: Infoagent vs *NotesWriting* performance comparison on GPT-4o-mini.

A.3 Benchmarks

We evaluated four multi-hop QA datasets: (1) FanOutQA (Zhu et al., 2024), which features complex fanout questions, (2) FRAMES (Krishna et al., 2024), requiring reasoning over 2–15 articles, (3) MultiHop-RAG (Tang and Yang, 2024), which involves retrieval and reasoning over news articles, and (4) HotpotQA (Yang et al., 2018), which requires multi-article reasoning. For FanOutQA, we evaluated all 310 examples from the development set, while for FRAMES, we used 549 multiple-constraint-tagged questions. For MultiHop-RAG and HotpotQA, we assessed performance on 500 examples from the test and development splits, respectively. FanOutQA, HotpotQA and Wikipedia comes under CC BY-SA 4.0 (Creative Commons Attribution-ShareAlike 4.0 International License), FRAMES under Apache 2.0. license and MultiHop-RAG under ODC-By (Open Data Commons Attribution License).

[7] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[8] https://serper.dev/

A.4 Models

Models: We conduct experiments with two models, representing both closed and open weights: `GPT-4o-mini`[9] and `Llama 3.1-70-Instruct` (Dubey et al., 2024). The temperature is set to 0.7, and the same LLM is used for generating reasoning steps and *NotesWriting* (i.e., $\mathcal{M}_s = \mathcal{M}$). `Llama 3.1-70-Instruct` was hosted using vLLM (Kwon et al., 2023) across 8 A100-80GB GPUs, supporting a maximum context length of $64K$. With parallelization, evaluation runs took approximately 9–10 hours for MultiHop-RAG, HotpotQA, and FRAMES, and around 15 hours for FanOutQA. `GPT-4o-mini`, which has a context length of $128K$, completed evaluations in approximately 7 hours for FRAMES and FanOutQA, 2 hours for HotpotQA, and 27 minutes for MultiHop-RAG. The reported times include the full end-to-end process, accounting for rate limits, Wikipedia queries, and *NotesWriting*.

A.5 Standard deviation across runs

We ran the *NotesWriting* and ReNAct across all datasets and models three times to see the variance across different runs. We report the results in Table 7.

Model	Dataset	Avg F1	GPT-4 Score
GPT-4o-mini	Fanout	± 1.86	± 2.45
	Frames	± 1.10	± 2.35
Llama-3.1 70B	Fanout	± 3.79	± 1.54
	Frames	± 4.42	± 5.76

Table 7: Standard deviation across Frames & FanoutQA.

A.6 Examples comparing ReNAct with baselines

[9] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Model	Setting	Benchmark	F1 (%)	GPT-4	Main Tokens		Notes Tokens	
				Score (%)	Input	Output	Input	Output
GPT-4o-mini	Baseline	Fanout-QA	33.6	15.2	273K	385	-	-
		Frames	24.0	22.0	183K	312	-	-
		Hotpot-QA	36.4	42.8	99K	205	-	-
		MultiHop-RAG	23.7	48.0	909K	322	-	-
	NotesWriting	Fanout-QA	41.9	21.3	68K	444	902K	1.8K
		Frames	43.9	42.3	37K	280	658K	862
		Hotpot-QA	46.2	53.8	26K	193	433K	650
		MultiHop-RAG	36.0	65.6	189K	412	40K	324
LLaMA-3.1-70B	Baseline	Fanout-QA	21.0	8.4	197K	511	-	-
		Frames	19.4	21.1	192K	732	-	-
		Hotpot-QA	31.5	38.2	167K	557	-	-
		MultiHop-RAG	35.4	64.8	429K	313	-	-
	NotesWriting	Fanout-QA	36.0	22.9	154K	552	1.4M	7.6K
		Frames	26.9	33.3	116K	735	1.2M	3.9K
		Hotpot-QA	36.5	53.0	100K	561	1.1M	3.3K
		MultiHop-RAG	38.0	64.8	99K	680	338K	2.9K

Table 8: IRCoT results for GPT-4o-mini and LLaMA-3.1-70B across Baseline and *NotesWriting*. Token counts are rounded to the nearest thousand (input/output).

Model	Setting	Benchmark	F1 (%)	GPT-4	Main Tokens		Notes Tokens	
				Score (%)	Input	Output	Input	Output
GPT-4o-mini	Baseline	Fanout-QA	35.1	14.2	268K	384	-	-
		Frames	26.3	23.7	50K	314	-	-
		Hotpot-QA	34.8	39.0	36K	218	-	-
		MultiHop-RAG	28.9	65.7	204K	319	-	-
	NotesWriting	Fanout-QA	42.3	22.2	65K	409	968K	2.2K
		Frames	27.7	29.8	43K	358	47K	56
		Hotpot-QA	34.5	45.8	33K	258	45K	49
		MultiHop-RAG	30.2	66.6	205K	312	14K	49
LLaMA-3.1-70B	Baseline	Fanout-QA	23.0	11.4	195K	453	-	-
		Frames	16.4	18.6	86K	640	-	-
		Hotpot-QA	24.7	31.2	83K	573	-	-
		MultiHop-RAG	36.1	67.0	301K	558	-	-
	NotesWriting	Fanout-QA	35.8	24.2	154K	555	1.3M	7.5K
		Frames	20.0	25.3	67K	592	47K	66
		Hotpot-QA	34.0	47.0	65K	563	45K	77
		MultiHop-RAG	30.5	66.4	67K	710	15K	38

Table 9: FLARE results for GPT-4o-mini and LLaMA-3.1-70B across Baseline and *NotesWriting*. Token counts are rounded to the nearest thousand (input/output).

Question: What law schools did the most recent four Supreme Court justices attend?
Thought: I need to gather information on the law schools attended by the most recent four Supreme Court justices. First, I will search for the names of the most recent four justices to find out where they studied law. **Action:** search[Supreme Court justices; Who are the most recent four Supreme Court justices?] **Observation:** (Result 1) List of justices of the Supreme Court of the United States - The most recent four Supreme Court justices are: Amy Coney Barrett, Ketanji Brown Jackson, Neil Gorsuch, and Brett Kavanaugh. (Result 2) Ideological leanings of United States Supreme Court justices - The most recent four Supreme Court justices are: Amy Coney Barrett (appointed in 2020), Ketanji Brown Jackson (appointed in 2022), Brett Kavanaugh (appointed in 2018), and Neil Gorsuch (appointed in 2017). (Result 3) Demographics of the Supreme Court of the United States - The most recent four Supreme Court justices are: Ketanji Brown Jackson (appointed in 2022), Amy Coney Barrett (appointed in 2020), Brett Kavanaugh (appointed in 2018), and Neil Gorsuch (appointed in 2017).
Thought: The most recent four Supreme Court justices are Amy Coney Barrett, Ketanji Brown Jackson, Neil Gorsuch, and Brett Kavanaugh. Next, I need to find out which law schools they attended. I will start by searching for the law school attended by Amy Coney Barrett. **Action:** search[Amy Coney Barrett; Which law school did Amy Coney Barrett attend?] **Observation:** (Result 1) Amy Coney Barrett attended Rhodes College (BA) and University of Notre Dame (JD). She graduated from Notre Dame Law School in 1997 with a Juris Doctor, *summa cum laude*, ranked first in her class.
Thought: Amy Coney Barrett attended Notre Dame Law School. Next, I need to find out which law school Ketanji Brown Jackson attended. **Action:** search[Ketanji Brown Jackson; Which law school did Ketanji Brown Jackson attend?] **Observation:** (Result 1) Ketanji Brown Jackson attended Harvard Law School, where she graduated in 1996 with a Juris Doctor, *cum laude*. (Result 2) Joe Biden Supreme Court candidates - Ketanji Brown Jackson attended Harvard University for her law degree. (Result 3) Demographics of the Supreme Court of the United States - Ketanji Brown Jackson attended Harvard Law School.
Thought: Ketanji Brown Jackson attended Harvard Law School. Now, I need to find out which law school Neil Gorsuch attended. **Action:** search[Neil Gorsuch; Which law school did Neil Gorsuch attend?] **Observation:** (Result 1) Neil Gorsuch received his legal education at Harvard Law School and graduated in 1991 with a Juris Doctor, *cum laude*.
Thought: Neil Gorsuch also attended Harvard Law School. Finally, I need to find out which law school Brett Kavanaugh attended. **Action:** search[Brett Kavanaugh; Which law school did Brett Kavanaugh attend?] **Observation:** (Result 1) Brett Kavanaugh - Kavanaugh graduated from Yale Law School with a Juris Doctor degree in 1990.
Thought: Brett Kavanaugh graduated from Yale Law School. Now I have gathered the law schools attended by the most recent four Supreme Court justices: Amy Coney Barrett attended Notre Dame Law School, Ketanji Brown Jackson and Neil Gorsuch both attended Harvard Law School, and Brett Kavanaugh attended Yale Law School. **Action:** finish[Notre Dame Law School, Harvard Law School, Harvard Law School, Yale Law School]

Table 10: An example of *NotesWriting* on the FanoutQA dataset using GPT-4o-mini.

Question: According to the 2000 United States census, what was the 2000 population of the birth city of the only 21st-century mayor of Austin, Texas who also served as mayor in the 1990s? Round your answer to the nearest thousand.
Thought: I need to determine the birth city of the only 21st-century mayor of Austin, Texas who also served as mayor in the 1990s and then find its population in the 2000 United States census. First, I will identify the mayor who meets this criterion. **Action:** search[Austin, Texas mayors; Who was the 21st-century mayor of Austin, Texas who also served as mayor in the 1990s?] **Observation:** (Result 1) Mayor of Austin - Kirk Watson took office as mayor on January 6, 2023, for a second term, having served as mayor from 1997 to 2001. (Result 2) 2024 Austin mayoral election - Kirk Watson (Result 3) Austin, Texas - Kirk Watson
Thought: The only 21st-century mayor of Austin, Texas who also served as mayor in the 1990s is Kirk Watson. I now need to find his birth city. **Action:** search[Kirk Watson; Where was Kirk Watson born?] **Observation:** (Result 1) Kirk Watson - Oklahoma City, Oklahoma, U.S. (Result 2) Austin, Texas - Kirk Watson was born in 1958 in Oak Park, Illinois
Thought: The information provided contains conflicting birth cities for Kirk Watson. To confirm his birth city, I will perform another search. **Action:** search[Kirk Watson (American politician); What is the confirmed birth city of Kirk Watson?] **Observation:** (Result 1) Kirk Watson - Oklahoma City
Thought: Based on the confirmed information, Kirk Watson's birth city is Oklahoma City. Now, I need to find the population of Oklahoma City in the 2000 United States census. **Action:** search[Oklahoma City, Oklahoma; What was the population of Oklahoma City in the 2000 United States census?] **Observation:** (Result 1) Oklahoma City - 506,132
Thought: The population of Oklahoma City in the 2000 United States census was 506,132. To answer the question, I need to round this number to the nearest thousand. **Action:** finish[506,000]

Table 11: An example of *NotesWriting* on the Frames dataset using LLama-3.1-70B.

Question: Diago Costa played for which club when he was awarded the first FIFA World Cup Goal based on a VAR Decision?
Thought: I need to determine which club Diago Costa was playing for when he was awarded the first FIFA World Cup Goal based on a VAR Decision. First, I will gather information on the first FIFA World Cup Goal awarded based on a VAR Decision. **Action:** search[FIFA World Cup Goal; What was the first FIFA World Cup Goal awarded based on a VAR Decision, and who scored it?] **Observation:** (Result 1) 2022 FIFA World Cup - Ao Tanaka
Thought: It seems like the search result did not provide the relevant information about the first FIFA World Cup Goal awarded based on a VAR Decision. **Action:** search[FIFA World Cup VAR Decision; What was the first FIFA World Cup Goal awarded based on a VAR Decision, and who scored it?] **Observation:** (Result 1) 2018 FIFA World Cup - Diego Costa's first goal against Portugal became the first World Cup goal based on a VAR decision. (Result 2) 2022 FIFA World Cup - The first FIFA World Cup Goal awarded based on a VAR Decision was scored by Ao Tanaka of Japan against Spain. (Result 3) 2019 FIFA Women's World Cup - The first FIFA World Cup goal awarded based on a VAR decision was scored by French midfielder Antoine Griezmann in the 58th minute of a group stage match against Australia on June 16, 2018. (Result 4) Video assistant referee - The first VAR decision at the World Cup came on 16 June 2018 in a group stage match between France and Australia, where referee Andres Cunha awarded a penalty to France after consulting with the VAR. The resulting penalty kick was successfully converted by Antoine Griezmann of France, but the goal was initially disallowed for offside. However, the VAR team correctly advised the referee that the offside infraction was committed by an Australian player, not a French player, and the goal was awarded.
Thought: It seems like the search results provided conflicting information about the first FIFA World Cup Goal awarded based on a VAR Decision. However, according to Result 1 of the second search, Diego Costa's first goal against Portugal became the first World Cup goal based on a VAR decision.
Action: search[Diego Costa; Which club did Diego Costa play for in the 2018 FIFA World Cup?] **Observation:** (Result 1) Diego Costa - Atlético Madrid
Thought: Diego Costa played for Atlético Madrid when he was awarded the first FIFA World Cup Goal based on a VAR Decision.
Action: finish[Atlético Madrid] **Observation:** Atlético Madrid

Table 12: An Example of *NotesWriting* on Frames dataset using GPT-4o-mini.

Question: If my future wife has the same first name as the 15th first lady of the United States' mother and her surname is the same as the second assassinated president's mother's maiden name, what is my future wife's name?

Thought: I need to determine the first name of the 15th first lady of the United States and the surname of the second assassinated president's mother's maiden name. Let's start by identifying the 15th first lady of the United States and her mother's name.

Action: search[15th first lady of the United States; Who is the 15th first lady and what is her mother's name?]

Observation: No relevant information, try a different search term.

Thought: I need to rephrase my search to find the correct information about the 15th first lady of the United States and her mother's name. I will search for the name of the 15th first lady directly.

Action: search[15th first lady of the United States; Who is the 15th first lady?]

Observation: No relevant information, try a different search term.

Thought: It seems that I'm having trouble finding information about the 15th first lady of the United States. I will now search for a list of all first ladies to identify her name and then find her mother's name.

Action: search[List of first ladies of the United States; What is the name of the 15th first lady?]

Observation: (Result 1) List of first ladies of the United States - The 15th first lady was Harriet Lane, who served from March 4, 1857, to March 4, 1861.

Thought: I have identified the 15th first lady of the United States as Harriet Lane. Now, I need to find information about her mother's name.

Action: search[Harriet Lane; What is the name of Harriet Lane's mother?]

Observation: (Result 1) Harriet Lane - Harriet Lane's mother was Jane Ann Buchanan Lane.

Thought: I have found that Harriet Lane's mother was named Jane Ann Buchanan Lane. Now, I need to identify the second assassinated president of the United States and find his mother's maiden name.

Action: search[second assassinated president of the United States; Who is the second assassinated president?]

Observation: (Result 1) List of United States presidential assassination attempts and plots - James A. Garfield was the second U.S. president to be assassinated, occurring on July 2, 1881, less than four months after he took office.

(Result 2) List of presidents of the United States by time in office - The second assassinated president is James A. Garfield.

Thought: I need to rephrase my search to find the correct information about James A. Garfield's mother's maiden name. I will search for information about James A. Garfield and his family.

Action: search[James A. Garfield's mother; What was the maiden name of James A. Garfield's mother?]

Observation: No relevant information, try a different search term.

Thought: It seems I'm having difficulty finding the maiden name of James A. Garfield's mother. I will search for biographies of James A. Garfield to find more details about his family.

Action: search[James A. Garfield biography; What is known about James A. Garfield's family?]

Observation: (Result 1) James A. Garfield - James Abram Garfield was born on November 19, 1831, in a log cabin in Moreland Hills, Ohio. He was the youngest of five children. Garfield married Lucretia Rudolph in 1858, and they had seven children, five of whom survived infancy.

Thought: I still need to find the maiden name of James A. Garfield's mother. I will focus my search specifically on his mother's name.

Action: search[James A. Garfield's family; What is the name of James A. Garfield's mother?]

Observation: (Result 1) James A. Garfield - James A. Garfield's mother was named Eliza.

Thought: I am struggling to find the maiden name of James A. Garfield's mother. I will search for historical records or references that detail Eliza Garfield's family background.

Action: search[Eliza Garfield biography; What is known about Eliza Garfield's family and maiden name?]

Observation: No relevant information, try a different search term.

Thought: It seems I am having persistent difficulties finding the maiden name of James A. Garfield's mother, Eliza. I will try searching for historical documentation or genealogical resources that might provide her maiden name.

Action: search[Eliza Garfield maiden name; What is the maiden name of James A. Garfield's mother?]

Observation: (Result 1) Abbie C. B. Robinson - The maiden name of President James A. Garfield's mother is Eliza Ballou.

Thought: I have found that the maiden name of James A. Garfield's mother is Eliza Ballou. Now, I have the necessary information to form the name of my future wife. Her first name is Jane (from Jane Ann Buchanan Lane) and her last name is Ballou (from Eliza Ballou).

Action: finish[Jane Ballou]

Table 13: An example of *NotesWriting* on the Frames dataset using GPT-4o-mini.

Question: What 2016 film stars actor Xavier Samuel and an actress who portrayed Vasilisa Dragomir in the film "Vampire Academy"?
Thought: I need to search for the film that stars Xavier Samuel, then find out who played Vasilisa Dragomir in "Vampire Academy" to identify the actress. Finally, I will check if both appeared in the same film in 2016.
Action: search[Xavier Samuel filmography]
Observation: Wikipedia Title: Xavier Samuel Xavier Samuel (born 10 December 1983) is an Australian film and theatre actor. He has appeared in leading roles in the feature films Adore, September, Further We Search, Newcastle, The Loved Ones, Frankenstein, A Few Best Men, and played Riley Biers... Wikipedia Title: Mohanlal filmography Mohanlal is an Indian actor, producer, distributor, host, and playback singer who has starred in mainstream blockbuster and art-house films. He has made over 355 appearances in feature films during the span of his career, primarily in Malayalam... Wikipedia Title: Benedict Samuel Benedict Samuel is an Australian actor, writer, and director best known for playing Jervis Tetch / Mad Hatter in the Fox crime series Gotham and as Owen in The Walking... Wikipedia Title: Gérard Depardieu Gérard Xavier Marcel Depardieu is a multinational French actor, known to be one of the most prolific in film history. An icon of French cinema, considered a world star in... Wikipedia Title: Jagadish filmography P. V. Jagadish Kumar, known as Jagadish, is an Indian actor, screenwriter, and television presenter. He has starred in over 400 Malayalam...
Thought: The search still did not yield the specific filmography of Xavier Samuel. I will try searching for his name along with the year 2016 to see if I can find a film he starred in that year.
Action: search[Xavier Samuel 2016 film]
Observation: Wikipedia Title: Xavier Samuel Xavier Samuel (born 10 December 1983) is an Australian film and theatre actor. He has appeared in leading roles in the feature films Adore, September, Further We Search, Newcastle, The Loved Ones, Frankenstein, A Few Best Men, and played Riley Biers... Wikipedia Title: Mr. Church Mr. Church is a 2016 American drama film directed by Bruce Beresford and written by Susan McMartin. The film stars Eddie Murphy as the title character with Britt Robertson, Xavier Samuel, Lucy Fry, Christian Madsen and Natascha McElhone also... Wikipedia Title: Benedict Samuel Benedict Samuel is an Australian actor, writer, and director best known for playing Jervis Tetch / Mad Hatter in the Fox crime series Gotham and as Owen in The Walking... Wikipedia Title: Samuel Hunt (actor) Samuel Caleb Hunt (born October 10, 1986) is an American actor best known for his reoccurring role as Greg “Mouse” Gerwitz in the television series Chicago Fire and Chicago P.D., playing Xavier Rosen in the third season of Empire, and for starring... Wikipedia Title: Spin Out (film) Spin Out is a 2016 Australian romantic comedy film directed by Tim Ferguson and Marc Gracie and starring Xavier Samuel and Morgan...

Table 14: An example of ReAcT on the HotpotQA dataset using GPT-4o-mini. It starts by repeatedly searching similar query.

Thought: I have found that Xavier Samuel stars in the film “Spin Out” in 2016. Now I need to find out who portrayed Vasilisa Dragomir in “Vampire Academy.”
Action: search[Vasilisa Dragomir Vampire Academy actress]
Observation: Wikipedia Title: Vampire Academy (film) Vampire Academy is a 2014 American fantasy comedy horror film directed by Mark Waters from a screenplay by Daniel Waters, based on the first book of the novel series of the same name by Richelle Mead. The film stars an ensemble cast led by Zoey... Wikipedia Title: Bloodlines (book series) Bloodlines is a book series by Richelle Mead. It is a spin-off of Mead’s Vampire Academy series. The books follow alchemist Sydney Sage. After the incident with Rose and Dimitri, Sydney is being closely watched by the Alchemists (humans who help... Wikipedia Title: List of fictional princesses This is a list of fictional princesses that have appeared in various works of fiction. This list is organized by medium and limited to well-referenced, notable examples of fictional...
Thought: The search did not provide the specific actress who portrayed Vasilisa Dragomir. I will select the “Vampire Academy” film page to find the cast details.
Action: select[Vampire Academy (film)]
Observation: 2014 fantasy comedy horror film by Mark Waters | Vampire Academy | | | Theatrical release poster | | | Directed by | Mark Waters | | Screenplay by | Daniel Waters | | Based on | *Vampire Academy* by Richelle Mead | | Produced by | Don Murphy, Susan Montford, Michael Preger, Deepak Nayar | | Cinematography | Tony Pierce-Roberts | | Edited by | Chris Gill | | Music by | Rolfe Kent | | Production companies | Kintop Pictures, Reliance Entertainment, Angry Films, Preger Entertainment | | Distributed by | The Weinstein Company | | Release dates | February 7, 2014 (US), July 14, 2014 (UK straight-to-DVD) | *Vampire Academy* is a 2014 American fantasy comedy horror film directed by Mark Waters from a screenplay by Daniel Waters, based on the first book of the novel series by Richelle Mead. The film stars an ensemble cast led by Zoey Deutch as Rose Hathaway, a Dhampir, and Lucy Fry as Lissa Dragomir, her Moroi best friend, and follows their return to the St Vladimir’s Academy boarding school, after being on the run for one year.
Thought: I found that Lucy Fry portrays Vasilisa Dragomir in “Vampire Academy.” Now, I will check if both Xavier Samuel and Lucy Fry starred in the same film in 2016. Since Xavier Samuel is in “Spin Out,” I need to find out if Lucy Fry is also in this film.
Action: search[Spin Out film cast]

Table 15: Continued example of ReAcT on the HotpotQA dataset using GPT-4o-mini. It focuses on "Spin Out" but misses the potential answer "Mr. Church".

Observation: Wikipedia Title: Spin Out (film)

Spin Out is a 2016 Australian romantic comedy film directed by Tim Ferguson and Marc Gracie and starring Xavier Samuel and Morgan...

Wikipedia Title: Spinning Out

Spinning Out is an American drama television series, created by Samantha Stratton, that premiered on Netflix on January 1, 2020. In February 2020, the series was canceled after one season.

The series stars Kaya Scodelario as Kat Baker, a young ice...

Wikipedia Title: Spin (2021 film)

Spin is a 2021 American comedy film released as a Disney Channel Original Movie (DCOM). It is directed by Manjari Makijany from a screenplay by Carley Steiner and Josh A. Cagan. Produced by Disney Channel Original Productions, the film stars...

Wikipedia Title: Spinning Man

Spinning Man is a 2018 thriller film directed by Simon Kaijser based on a novel by George Harrar and adapted to screenplay by Matthew Aldrich. It stars Guy Pearce, Pierce Brosnan, Minnie Driver, Alexandra Shipp, Clark Gregg and Odeya Rush. The film...

Wikipedia Title: Dog Man (film)

Dog Man is a 2025 American animated superhero comedy film based on Dav Pilkey's children's graphic novel series Dog Man, produced by DreamWorks Animation and Scholastic Entertainment, and distributed by Universal Pictures. It is a spin-off and story...

Thought: The search did not provide the cast details for "Spin Out." I will select the "Spin Out (film)" page to find the cast.

Action: select[Spin Out (film)]

Observation: 2016 Australian film

| Spin Out | |

| — | — |

| Theatrical film poster | |

| Directed by | Tim Ferguson, Marc Gracie |

| Written by | Edwina Exton, Tim Ferguson |

| Produced by | Marc Gracie, David Redman |

| Starring | Xavier Samuel, Morgan Griffin, Lincoln Lewis |

| Cinematography | Justin Brickle |

| Edited by | Ken Sallows |

| Music by | Jon Hume |

Spin Out is a 2016 Australian romantic comedy film directed by Tim Ferguson and Marc Gracie and starring Xavier Samuel and Morgan Griffin.

Thought: The search results confirm that "Spin Out" stars Xavier Samuel, but there is no mention of Lucy Fry in the cast list. Thus, they did not appear in the same film. I will now summarize the findings to answer the question.

Action: finish[There is no common film between Xavier Samuel and Lucy Fry in 2016.]

Observation: There is no common film between Xavier Samuel and Lucy Fry in 2016.

Table 16: Continued example of ReAcT on the HotpotQA dataset using GPT-4o-mini. It fails by focusing exclusively on Xavier Samuel's film "Spin Out" while completely overlooking "Mr. Church," despite it being mentioned in the search results alongside both actors' names. This prevents it from from discovering that "Mr. Church" is the 2016 film starring both Xavier Samuel and Lucy Fry (who played Vasilisa Dragomir in "Vampire Academy").

Question: What 2016 film stars actor Xavier Samuel and an actress who portrayed Vasilisa Dragomir in the film "Vampire Academy"?
Thought: I need to identify the actress who portrayed Vasilisa Dragomir in "Vampire Academy" and then find a 2016 film that stars Xavier Samuel alongside her. First, I will search for the cast of "Vampire Academy" to determine who played Vasilisa Dragomir. **Action:** search[Vampire Academy cast; Who portrayed Vasilisa Dragomir in the film "Vampire Academy"?] **Observation:** (Result 1) Vampire Academy (film) - Lucy Fry portrayed Vasilisa "Lissa" Dragomir in the film "Vampire Academy".
Thought: Lucy Fry portrayed Vasilisa Dragomir in "Vampire Academy." Now, I need to find a 2016 film that stars both Xavier Samuel and Lucy Fry. **Action:** search[Xavier Samuel; What films did Xavier Samuel star in during 2016?] **Observation:** (Result 1) Xavier Samuel - In 2016, Xavier Samuel played the titular character in *The Death and Life of Otto Bloom*, and performed alongside Kate Beckinsale, Chloe Sevigny, Stephen Fry, and James Fleet in the critically-acclaimed film *Love and Friendship*, adapted from Jane Austen novel *Lady Susan*.
Thought: I found two films that Xavier Samuel starred in during 2016: *The Death and Life of Otto Bloom* and *Love and Friendship*. Now, I need to check if Lucy Fry appeared in either of these films. **Action:** search[Lucy Fry; What films did Lucy Fry star in during 2016?] **Observation:** (Result 1) Lucy Fry - In 2016, Lucy Fry starred in the films *Mr. Church* as Poppy and *The Darkness* as Stephanie Taylor.
Thought: Lucy Fry starred in *Mr. Church* and *The Darkness* in 2016. I need to check if either of these films also features Xavier Samuel. **Action:** search[Mr. Church; Did Xavier Samuel star in Mr. Church?] **Observation:** (Result 1) Mr. Church - Xavier Samuel is listed as one of the stars in the film *Mr. Church*.
Thought: I have confirmed that both Xavier Samuel and Lucy Fry starred in the film *Mr. Church* in 2016. **Action:** finish[*Mr. Church*] **Observation:** *Mr. Church*

Table 17: The previouis example with *NotesWriting* on the HotpotQA dataset using GPT-4o-mini.

Question: As of August 1, 2024, what is the largest city of the 9th largest country by land area in Europe?
Thought: I need to determine the 9th largest country by land area in Europe and then find its largest city. First, I will identify the 9th largest country by land area in Europe. **Action:** search[Largest countries in Europe by land area; Which is the 9th largest country by land area in Europe?] **Observation:** (Result 1) List of countries and dependencies by area - The 9th largest country by land area in Europe is Kazakhstan (2,699,700 km^2). (Result 2) List of European countries by area - Poland
Thought: It seems like there are multiple conflicting sources for the 9th largest country by land area in Europe, with one source saying it's Kazakhstan, but Kazakhstan is not in Europe, and another source saying it's Poland but without providing more information. In this case, I'll try to find the correct 9th largest country by land area in Europe. **Action:** search[Largest countries in Europe by land area; What are the top 10 largest countries by land area in Europe?] **Observation:** (Result 1) List of countries and dependencies by area - 1. Russia - 16,376,870 km^2 2. Ukraine - 579,330 km^2 3. France - 640,427 km^2 4. Spain - 498,980 km^2 5. Sweden - 407,284 km^2 6. Norway - 366,704 km^2 7. Germany - 349,390 km^2 8. Finland - 303,815 km^2 9. Poland - 304,255 km^2 10. Italy - 295,717 km^2
Thought: The 9th largest country by land area in Europe is indeed Poland, with a land area of 312,679 km^2 or 304,255 km^2. Now, I need to determine the largest city of Poland. **Action:** search[Poland; What is the largest city in Poland as of August 1, 2024?] **Observation:** (Result 1) Poland - Warsaw (Result 2) Invasion of Poland - The largest city in Poland as of August 1, 2024, is Warsaw (Polish: Warszawa), which is also the capital of Poland.
Thought: The largest city of the 9th largest country by land area in Europe as of August 1, 2024, is Warsaw. **Action:** finish[The largest city of the 9th largest country by land area in Europe as of August 1, 2024, is Warsaw.]

Table 18: An example from the FRAMES dataset where note taking leads to conflicting information (in bold). To clarify the confusion the agent re-tries the search to find the correct information.

Question: A general motors vehicle is named after the largest ward in the country of Monaco. How many people had walked on the moon as of the first model year of the vehicle? Note: the model year is not the same as the year the model was first produced.
Thought: I need to determine which General Motors vehicle is named after the largest ward in Monaco and find out the first model year of that vehicle. After that, I will find out how many people had walked on the moon by that year. Let's start by searching for the General Motors vehicle named after the largest ward in Monaco. **Action:** search[General Motors vehicle named after the largest ward in Monaco; What is the largest ward in Monaco and which GM vehicle is named after it?] **Observation:** No relevant information, try a different search term.
Thought: I need to find the largest ward in Monaco first and then see if there is a General Motors vehicle named after it. I will search for the largest ward in Monaco. **Action:** search[largest ward in Monaco; What is the largest ward in Monaco?] **Observation:** (Result 1) Monaco - The largest quarter in Monaco is Monte Carlo.
Thought: The largest ward in Monaco is Monte Carlo. Now, I need to search for any General Motors vehicle named after Monte Carlo and find out its first model year. **Action:** search[Monte Carlo; What is the General Motors vehicle named after Monte Carlo and what is its first model year?] **Observation:** (Result 1) Chevrolet Monte Carlo - The Chevrolet Monte Carlo is the General Motors vehicle named after Monte Carlo, and it was introduced for the 1970 model year.
Thought: The Chevrolet Monte Carlo is named after the largest ward in Monaco and was introduced for the 1970 model year. Now, I need to find out how many people had walked on the moon by the end of 1970. **Action:** search[moon landings; How many people had walked on the moon by the end of 1970?] **Observation:** (Result 1) Moon landing - **A total of twelve astronauts have landed on the Moon**. (Result 2) Apollo program - By the end of 1970, twelve people had walked on the Moon.
Thought: By the end of 1970, a total of twelve astronauts had walked on the Moon. Now I can provide the final answer regarding how many people had walked on the Moon as of the first model year of the Chevrolet Monte Carlo. **Action:** finish[12] **Observation:** 12

Table 19: An example of hallucination by the note taking LLM in *NotesWriting* on the Frames dataset by GPT-4o-mini. The in-correct note taken is in bold. Instead the note writing LLM should have calculated the number of people walked before 1970 which is 4.

Question: How old would the founder of the publishing company of the magazine that serialized the manga series Raw Hero have been the year the magazine ended publication?
Thought: I need to first identify the founder of the publishing company that serialized the manga series Raw Hero. Then I will determine the year the magazine ended publication and calculate the founder's age at that time. **Action:** search[Raw Hero; Which publishing company serialized the manga series Raw Hero?] **Observation:** (Result 1) Raw Hero - *Raw Hero* was serialized in Kodansha's *Evening*.
Thought: The manga series Raw Hero was serialized in Kodansha's *Evening* magazine. Now, I need to find out who the founder of Kodansha is. **Action:** search[Kodansha; Who is the founder of Kodansha?] **Observation:** (Result 1) Kodansha - Kodansha was founded by Seiji Noma in 1909.
Thought: Seiji Noma is the founder of Kodansha, which serialized *Raw Hero*. Next, I need to determine the year *Evening* magazine ended publication to calculate his age at that time. **Action:** search[Evening magazine; When did Evening magazine cease publication?] **Observation:** (Result 1) *Evening* (magazine) - The magazine ended publication on February 28, 2023. (Result 2) PM Magazine - *Evening Magazine* ceased publication on August 30, 1991.
Thought: The *Evening* magazine ceased publication on February 28, 2023. **I will use the founding year of Kodansha (1909) to calculate the age of Seiji Noma** at the time of the magazine's end in 2023. **Action:** finish[113] **Observation:** 113

Table 20: An example of hallucination by the main LLM in *NotesWriting* on the Frames dataset by GPT-4o-mini. The LLM uses the founding year of the magazine as the birth year of the founder (highlighted in bold) instead of first identifying the birth year.

IRCoT and FLARE fewshot prompt

Task: Given a question and relevant documents, generate a step-by-step reasoning process to solve it.

Instructions:
- Enclose each reasoning step within <s> </s> tags.
- Reason in short sentences as the sentences will be used as queries to retrieve relevant documents for subsequent steps.
- If provided with a partial reasoning chain, complete it by generating the next set of steps.
- If you do not have up-to-date knowledge, try to think of a plan to solve the question.
- Provide the final answer inside <answer> </answer> tags.

DOCUMENTS:
Nobody Loves You (When You're Down and Out) - "Nobody Loves You (When You're Down and Out)" is a song written by John Lennon released on his 1974 album "Walls and Bridges". The song is included on the 1986 compilation "Menlove Ave.", the 1990 boxset "Lennon", the 1998 boxset "John Lennon Anthology", the 2005 two-disc compilation "", and the 2010 boxset "Gimme Some Truth".

John Lennon/Plastic Ono Band - John Lennon/Plastic Ono Band is the debut studio album by English rock musician John Lennon. It was released in 1970, after Lennon had issued three experimental albums with Yoko Ono and "Live Peace in Toronto 1969", a live performance in Toronto credited to the Plastic Ono Band. The album was recorded simultaneously with Ono's debut avant garde solo album, "Yoko Ono/Plastic Ono Band", at Ascot Sound Studios and Abbey Road Studios using the same musicians and production team and nearly identical cover artwork.

Walls and Bridges - Walls and Bridges is the fifth studio album by English musician John Lennon. It was issued by Apple Records on 26 September 1974 in the United States and on 4 October in the United Kingdom. Written, recorded and released during his 18-month separation from Yoko Ono, the album captured Lennon in the midst of his "Lost Weekend". "Walls and Bridges" was an American "Billboard" number-one album and featured two hit singles, "Whatever Gets You thru the Night" and "#9 Dream". The first of these was Lennon's first number-one hit in the United States as a solo artist, and his only chart-topping single in either the US or Britain during his lifetime.

Question: Nobody Loves You was written by John Lennon and released on what album that was issued by Apple Records, and was written, recorded, and released during his 18 month separation from Yoko Ono?
Step-by-step reasoning:
<s>Identify album issued by Apple Records and recorded during John Lennon's 18-month separation from Yoko Ono.</s>
<s>The album "Walls and Bridges" was issued by Apple Records and recorded during this period.</s>
<s>Determine the album on which "Nobody Loves You (When You're Down and Out)" was released.</s>
<s>"Nobody Loves You (When You're Down and Out)" was released on "Walls and Bridges".</s>
<answer>Walls and Bridges</answer>

DOCUMENTS:
Route 13 (Laos) - Route 13 is the most important highway in the country of Laos. It begins at Boten in the North of Laos at the Chinese border. It connects the city of Vientiane to Luang Prabang in the north and roughly follows the line of the Mekong River down to the border with Cambodia. The road then continues at National Highway 7 in Cambodia. Also, this national highway leads past all three international airports in Laos: Vientiane Airport, Luang Prabang Airport, and Pakse Airport. Between Boten and Nateuy, Route 13 is at a length of 20 km as part of the Kunming-Bangkok Expressway, which is also part of the Asian Highway AH3. From Nateuy to Vientiane, Route 13 is part of AH12, and from Vientiane to the border of Cambodia, it is part of AH11.

National Route 13 (Vietnam) - National Route 13 is a highway in southern Vietnam stretching from the northeastern outskirts of Ho Chi Minh City, the commercial center of the country, towards the border to Cambodia. The highway starts around on the northern outskirts of Ho Chi Minh City, once the site of the military academy of the Army of the Republic of Vietnam, and travels north through the provinces.

Cambodia - Cambodia (Khmer: or Kampuchea), officially known as the Kingdom of Cambodia , is a country located in the southern portion of the Indochina Peninsula in Southeast Asia. It is 181,035 km² in area, bordered by Thailand to the northwest, Laos to the northeast, Vietnam to the east, and the Gulf of Thailand to the southwest.

Question: What is known as the Kingdom and has National Route 13 stretching towards its border?

Step-by-step reasoning:
<s>Cambodia is officially known as the Kingdom of Cambodia.</s>
<s>National Route 13 in Vietnam stretches towards the border of Cambodia.</s>
<s>Since Cambodia is referred to as the Kingdom and National Route 13 leads to its border, Cambodia is the correct answer.</s>
<answer>Cambodia</answer>

DOCUMENTS:
Vandenbrink GTO - The Vandenbrink GTO is a limited re-bodied version of the Ferrari 599 GTB Fiorano. It features an entirely new coachwork designed by Michiel van den Brink of Vandenbrink Design, inspired by the 1962 Ferrari 250 GTO as a tribute.

1969 World Sportscar Championship - The 1969 World Sportscar Championship was the 17th season of FIA World Sportscar racing, featuring Group 6 Prototype Sports Cars, Group 4 Sports Cars, and Group 3 Grand Touring Cars.

Ferrari 250 GTO - The Ferrari 250 GTO was a GT car produced by Ferrari from 1962 to 1964 for homologation into the FIA's Group 3 Grand Touring Car category. It was powered by Ferrari's Tipo 168/62 V12 engine.

Question: Which car, produced by Ferrari from 1962 to 1964 for homologation into the FIA's Group 3 Grand Touring Car category, inspired the Vandenbrink GTO?

Step-by-step reasoning:
<s>The Ferrari 250 GTO was produced by Ferrari from 1962 to 1964 for homologation into the FIA's Group 3 Grand Touring Car category.</s>
<s>The Vandenbrink GTO was inspired by the Ferrari 250 GTO.</s>
<s>Since the question asks for the car that inspired the Vandenbrink GTO, the correct answer is the Ferrari 250 GTO.</s>
<answer>Ferrari 250 GTO</answer>

DOCUMENTS:
Talula - "Talula" is a song by Tori Amos, released as the second single from her 1996 album "Boys For Pele". It reached #22 on the UK Singles Chart and appears in the Jan de Bont film "Twister".

Nema aviona za Zagreb - "Nema aviona za Zagreb" ("There Is No Plane to Zagreb") is a 2012 Dutch film by Louis van Gasteren. The film is a retrospective of events in the director's life from 1964 to 1969, filmed by him in that period and reflected on from his vantage point over 40 years later at the age of 90. It is also Jan de Bont's first film credit as cinematographer since 1992's "Basic Instinct".

Jan de Bont - Jan de Bont (born 22 October 1943) is a Dutch cinematographer, director, and film producer. He is widely known for directing the 1994 action film "Speed," starring Keanu Reeves and Sandra Bullock, and the disaster film "Twister." As a director of photography, he has shot numerous blockbusters and genre films, including "Cujo, Flesh and Blood, Die Hard, The Hunt for Red October," and "Basic Instinct."

Question: Who was born first? Jan de Bont or Raoul Walsh?

Step-by-step reasoning:
<s>Jan de Bont was born on October 22, 1943.</s>
<s>Raoul Walsh was born on March 11, 1887.</s>
<s>Since 1887 is earlier than 1943, Raoul Walsh was born first.</s>
<answer>Raoul Walsh</answer>

Figure 4: Fewshot prompt used for the evaluation of IRCoT and FLARE methods.

Prompt for Notes Writing

Extract relevant information which is not previously extracted from the Wikipedia page provided in markdown format relevant to the given query. You will be provided with the Wikipedia page, query, and the previously extracted content. Do not miss any information. Do not add irrelevant information or anything outside of the provided sources.
Provide the answer in the format: `<YES/NO>#<Relevant context>`.
Here are the rules:

- If you don't know how to answer the query - start your answer with `NO#`
- If the text is not related to the query - start your answer with `NO#`
- If the content is already extracted - start your answer with `NO#`
- If you can extract relevant information - start your answer with `YES#`

Example answers:

- `YES#Western philosophy originated in Ancient Greece in the 6th century BCE with the pre-Socratics.`
- `NO#No relevant context.`

Context: `{Context}`
Previous Context: `{PrevContext}`
Query: `{Query}`

Figure 5: Notes writing prompt for extracting the relevant information.

GPT-4 Judge Prompt

===Task===
I need your help in evaluating an answer provided by an LLM against a ground truth answer. Your task is to determine if the ground truth answer is present in the LLM's response. Please analyze the provided data and make a decision.
===Instructions===

1. Carefully compare the "Predicted Answer" with the "Ground Truth Answer".
2. Consider the substance of the answers – look for equivalent information or correct answers. Do not focus on exact wording unless the exact wording is crucial to the meaning.
3. Your final decision should be based on whether the meaning and the vital facts of the "Ground Truth Answer" are present in the "Predicted Answer."

===Input Data===

- **Question:** `«question»`
- **Predicted Answer:** `«LLM_response»`
- **Ground Truth Answer:** `«ground_truth_answer»`

===Output Format===
Provide your final evaluation in the following format:
Explanation: (How you made the decision?)
Decision: ("TRUE" or "FALSE")
Please proceed with the evaluation.

Figure 6: GPT-4 prompt for evaluating the correctness of predicted answer.

Quality evaluation prompt

You are asked to evaluate the reasoning chain produced in response to a question, particularly focusing on how effectively tools were used throughout the process. The evaluation should be based on the following clearly defined criteria. For each criterion, provide a numerical rating on a scale from 0 to 5, where 5 represents excellent performance and 0 indicates poor or entirely absent performance.

Criterion 1: Efficiency of the Steps Taken
Definition:
Evaluate the overall efficiency of each step in the reasoning chain, with specific focus on whether the tool calls and reasoning steps helped progress toward the final correct answer. Efficient steps reduce uncertainty, narrow the solution space, or directly contribute to solving the problem.
Rating Guide:
5 – Extremely efficient: Every step clearly advances the reasoning; no wasted effort.
4 – Highly efficient: Most steps are purposeful, with only minor inefficiencies.
3 – Moderately efficient: Some steps are valuable, others contribute little.
2 – Minimally efficient: Several steps are misdirected or low-impact.
1 – Poorly efficient: Most steps offer minimal or no progress toward the answer.
0 – Not efficient at all: Steps are irrelevant, aimless, or distracting.

Criterion 2: Redundancy of Steps
Definition:
Assess the reasoning chain for unnecessary repetition or duplication of steps, including redundant tool calls or rephrasing of the same logic without new insight. A low-redundancy chain avoids rework and keeps the progression streamlined.
Rating Guide:
5 – No redundancy: Each step is unique and adds distinct value.
4 – Very low redundancy: Only minor repetition, quickly resolved.
3 – Moderate redundancy: Some ideas or tool uses are repeated without added benefit.
2 – Noticeable redundancy: Multiple steps repeat similar content or actions unnecessarily.
1 – High redundancy: Repetition significantly detracts from conciseness.
0 – Extremely redundant: Most of the chain rehashes prior reasoning with no new value.

Criterion 3: Clarity and Coherence of the Reasoning Chain
Definition:
Examine how clearly and logically the reasoning chain progresses from the question to the final answer. This includes whether steps are easy to follow, logically connected, and free of ambiguity or excessive complexity.
Rating Guide:
5 – Exceptionally clear and coherent: The reasoning is logical, concise, and easy to follow.
4 – Mostly clear: The chain is understandable with minor clarity issues.
3 – Moderately clear: Some transitions or justifications are unclear or weak.
2 – Confusing in parts: Multiple unclear, inconsistent, or disjointed steps.
1 – Difficult to follow: Lacks logical flow or clear structure.
0 – Incomprehensible: The chain cannot be understood or followed logically.

First provide your reasoning of your evaluation then structure your responses as a json with the keys "Criterion 1", "Criterion 2", "Criterion 3" and the values as the ratings you provided.
Chain: {}

Figure 7: Prompt for quality evaluation of reasoning chain.