

Think, Search, Correct: A Self-Reflection Mechanism for Adaptive Tool-Use Agents

Megha Katak **Arshia Vadhani** **Joyce Lu** **Jiixin Yang**
mkatak@ucsd.edu avadhani@ucsd.edu jol072@ucsd.edu jiy016@ucsd.edu

Kun Zhou
kuzhou@ucsd.edu

Abstract

Building upon our Quarter 1 study, we redesign the Search-o1 framework to improve performance, reliability, and factual accuracy in complex multi-hop question answering tasks. Through a prior case study on a subset of the HotpotQA dataset, we identified recurring architectural weaknesses including skipped search calls leading to hallucinated answers, information loss during document extraction, and ineffective multi-hop search reflection.

Unlike traditional Retrieval-Augmented Generation (RAG) systems, Search-o1 iteratively retrieves and analyzes information through a Reason-in-Documents module, dynamically updating reasoning as new evidence is found. However, our earlier evaluations revealed failures in retrieval activation, extraction consistency, and multi-hop planning, resulting in incomplete and unsupported answers.

To address these issues, we introduce a structured self-reflection mechanism consisting of hallucination checks, adaptive query refinement, extraction recovery loops, and constructive content feedback. Our refined architecture enables the agent to iteratively Think, Search, and Correct until sufficient grounded evidence is obtained. We demonstrate how structured reflection and adaptive feedback significantly enhance the reliability of search-augmented language model agents in complex reasoning tasks.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable performance across reasoning and question answering tasks. However, they remain prone to hallucination — the confident generation of incorrect or unsupported information. This issue becomes especially problematic in multi-hop question answering, where answers require synthesizing evidence from multiple sources. In such settings, purely parametric reasoning is insufficient and external retrieval becomes necessary to ensure factual grounding.

Retrieval-Augmented Generation (RAG) systems have emerged as a popular solution by incorporating external documents into the reasoning process. While effective for single-hop retrieval, traditional RAG architectures often struggle with iterative reasoning and dynamic evidence integration.

To address this limitation, our team builds upon Search-o1 [Li et al. \(2025\)](#), a search-enhanced reasoning framework that interleaves chain-of-thought generation with iterative information retrieval. Search-o1 introduces a Reason-in-Documents module that analyzes retrieved content before integrating it into the reasoning stream. Unlike standard RAG pipelines, it dynamically updates its reasoning trajectory as new evidence becomes available.

Despite its architectural strengths, our Quarter 1 case study revealed significant weaknesses. Evaluating Search-o1 on a subset of the HotpotQA dataset using Qwen2.5-3B-Instruct [Yang et al. \(2024\)](#) and the Serper.dev search API, we observed low performance (Accuracy = 0.20, F1 = 0.20). Through detailed log analysis, we identified three recurring failure modes:

- Skipped search calls leading to unsupported final answers.
- Information loss during extraction despite successful retrieval.
- Ineffective multi-hop planning that failed to correct earlier reasoning errors.

These findings suggest that simply interleaving search and reasoning is insufficient. Agents must reflect on retrieved evidence, evaluate gaps, and strategically refine future search actions. Motivated by these observations, we introduce *Think, Search, Correct*, a structured self-reflection framework that incorporates hallucination detection, adaptive query refinement, and constructive feedback mechanisms. Our goal is to design a more self-aware and self-corrective tool-use agent capable of stabilizing multi-hop reasoning and improving factual reliability.

2 Methods

2.1 Adopted Baseline Framework: Search-o1

This study builds upon the Search-o1 architecture [Li et al. \(2025\)](#), a multi-step reasoning framework that interleaves “Chain-of-Thought” generation with iterative information retrieval. The original framework uses a “Reason-in-Documents” module to analyze retrieved information before integrating it into the reasoning stream.

To establish a functional baseline, we modified the backend infrastructure of the original implementation. We retained the Jina API to parse web page content but replaced the deprecated Bing Search API with the Serper.dev (Google) Search API for retrieval tasks. All experiments used Qwen2.5-3B-Instruct [Yang et al. \(2024\)](#) as the backbone Large Reasoning Model.

As illustrated in Figure 1, Visual overview of Search-o1 framework compared to vanilla reasoning pattern and agentic RAG framework, showing the loop between the Large Reasoning

Model, Search Tool, and Reason-in-Documents Module.

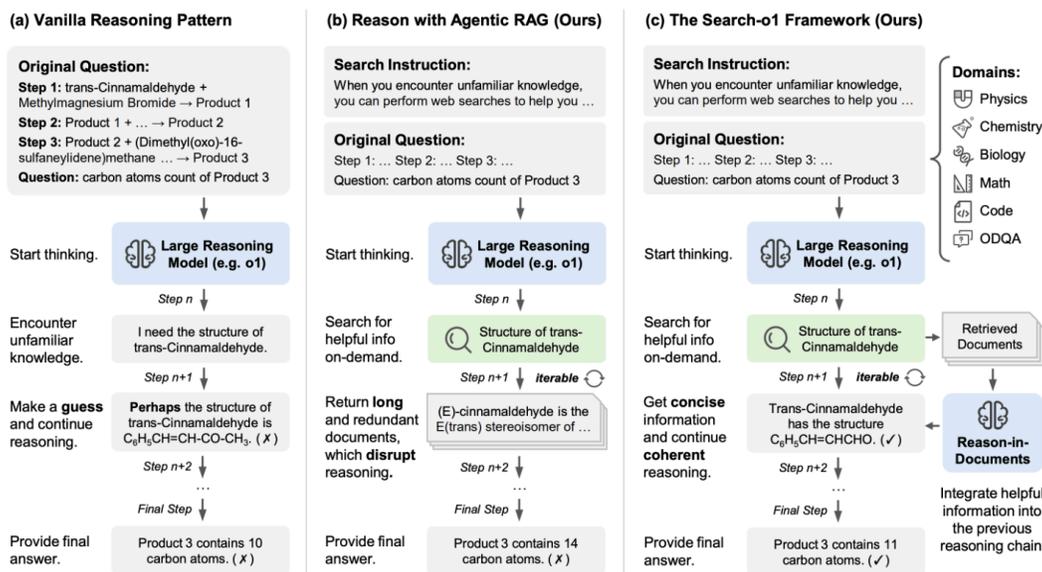


Figure 1: Visual overview of the Search-o1 framework compared to vanilla reasoning and agentic RAG architectures. The system interleaves a Large Reasoning Model, Search Tool, and Reason-in-Documents module in an iterative loop.

2.2 Phase 1: Initial Architecture Design

During our preliminary case study, we identified that the baseline architecture suffered from hallucination, information loss during extraction, and ineffective follow-up queries (poor search planning). To address these limitations, we designed an initial pipeline that incorporates several layers of validation and reflection logic.

2.2.1 Information Extraction Correction

We first addressed a technical bug in the original Search-o1's Reason-in-Documents module, where regex pattern mismatches caused the model to drop information that it had successfully retrieved. We corrected the regex pattern to ensure all generated findings are correctly captured and passed to downstream modules.

2.2.2 Hallucination Detection

To reduce unverified answers, we implemented a verification module. When the model attempts to finalize an answer without prior search, this module scans the generated response for factual claims (e.g., dates, names) that lack citations from the retrieval history. If unverified claims are detected, the system forces the agent to verify the information via

a search cycle. This mechanism prevents the agent from generating confident yet unsupported claims.

2.2.3 Retrieval Judge and Search Query Refinement (Gate 1)

To optimize retrieval relevance, we implemented a “Snippet Judge” to evaluate Serper API results before fetching full documents. This component analyzes retrieved snippets against the current search query to determine if they contain potentially useful leads. If the judge decides the results as irrelevant, it triggers a reflection mechanism prompting the model to refine the search query and search again. This step prevents the agent from wasting inference time and resources on irrelevant documents.

2.2.4 Extraction Refinement Loop

We added an extraction refinement loop to address the issue where the model fails to extract relevant data present in retrieved documents. This loop is triggered when the Reason-in-Documents module returns “No information found.” A presence checker scans the raw document text for query-relevant information. If a positive match is found, the system forces the model to re-analyze the documents. This self-correction mechanism recovers lost information and ensures the agent fully utilizes the retrieved information.

2.2.5 Content Judge and Reflection (Gate 2)

Following extraction (in the Reason-in-Documents module), the extracted information passes through a “content judge,” which assesses the utility of the extracted findings relative to the immediate search query. The judge evaluates whether the specific information requested by the current search query was successfully found. If the extraction is decided as insufficient, the system appends corrective instructions (e.g., “[System Warning: The information is insufficient. Suggestion: reflection...]”) to the reasoning stream to guide the next reasoning step.

2.2.6 Judge-Based Routing (Gate 3)

After each search-extraction cycle, we have introduced a high-level routing function aimed at deciding whether the o1-search agent should continue to search or finalize an answer. The judge receives the original question, the agent’s current reasoning trace and previous search results, the set of retrieved documents, and the search budget usage. It is prompted to utilize one of three output labels: FINALIZE, SEARCH_MORE, and GIVE_UP following a strict criteria that FINALIZE is only called when the model is certain that the key answer facts is supported by information in the retrieved documents and the reasoning is logical and consistent. If the judge returns SEARCH_MORE, the system appends a structured message (e.g., “The evidence is missing key facts. Generate a NEW and more precise search

query...”) that tells the agent to refine the next query and perform another retrieval step in the next turn. If the judge function returns FINALIZE the model will append another system message (“You now have enough information to answer. Do NOT search again. Write the final answer...”) pushing the agent to stop searching and create a final answer in the next turn.

2.3 Phase 2: Architectural Refinement and Final Architecture

Our initial diagnostic run and qualitative analysis of the Phase 1 execution logs revealed that some validation gates were often counterproductive. While intended to improve the searching and reasoning process, the judges sometimes introduced inaccurate decisions and conflicts.

2.3.1 Removal of Retrieval and Routing Gates

Retrieval Judge (Gate 1) Removal: Examination of search traces showed that truncated snippets from the search results often lacked enough context for the judge to make an accurate decision. This often led to false negatives, where relevant pages were rejected due to poor snippets/metadata, causing redundant searches. We removed this gate to prioritize recall, allowing the model to fetch the full content of top-ranked search results rather than judging based on incomplete snippets.

Routing Judge (Gate 3) Removal: We observed that the routing classifier frequently made SEARCH_MORE decisions that led to redundant search loops and sometimes conflicted with the model’s internal reasoning state. We removed this component to restore model autonomy, allowing the model to determine when it has sufficient information to generate a final answer based on accumulated context. If the model requires further information to reach a final answer, the Gate 2 reflection mechanism provides strategic guidance for the next search step.

2.3.2 Refinement of Content Feedback (Gate 2)

In the phase 1 pipeline, the Content Judge evaluated results strictly, checking whether the extracted information fully satisfied the immediate search query. We observed that this binary approach often caused the model to miss potential candidates or partial clues, providing inaccurate feedback to the agent.

In the final architecture, we refined the judge to evaluate whether the extracted content provides any useful information to the query. This allows the system to classify partial clues (e.g., a new entity to investigate) or useful corrections as successes. If the extracted content is still found to be vague or insufficient, a reflection module analyzes the gap between the user’s question and current findings. Unlike the phase 1 version which provided only high-level strategic advice, the refined module formulates a specific strategic follow-up query to guide the agent’s next step.

Furthermore, we revised the feedback mechanism to append neutral observations (e.g., “[Observation: The search results were limited. Suggestion: reflection]”) rather than the punitive “System Warnings” message. This ensures that the feedback guides the model constructively with a neutral tone without disrupting its reasoning flow.

2.3.3 Final System Architecture

The final optimized pipeline, built upon the Search-o1 baseline, integrates the following components:

- **Hallucination Check:** Prevents the generation of unverified factual claims by enforcing validation via a search cycle.
- **Extraction Refinement Loop:** Mitigates information loss by ensuring the model fully utilizes relevant data present in retrieved documents.
- **Content Judge and Reflection (Gate 2):** Analyzes the gap between current search results and user’s question, providing constructive feedback to guide subsequent reasoning steps.

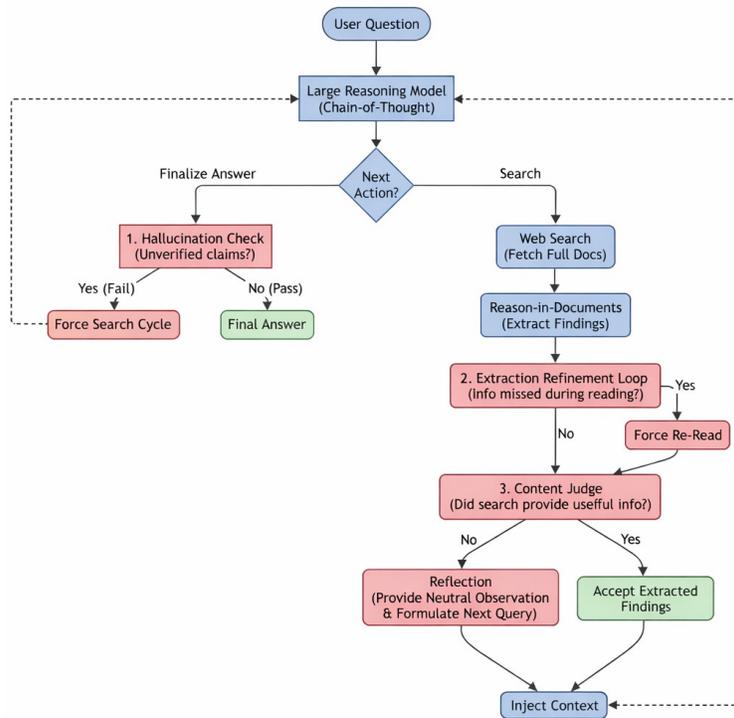


Figure 2: Final system architecture built upon the Search-o1 baseline with additional verification and refinement modules.

2.4 Final Evaluation Setup

Dataset: For our final evaluation, we utilized the HotpotQA test dataset [Yang et al. \(2018\)](#), a benchmark designed to evaluate multi-hop reasoning and document synthesis. The original test split contains 7,405 questions; however, processing full web pages iteratively is highly token-intensive. Due to the rate limits and computational constraints of the free-tier Jina API we used, we evaluated our baseline and improved agent on a subset of 120 samples to maintain a meaningful evaluation set while operating within token limits.

Model and Hyperparameters: All experiments were conducted using Qwen2.5-3B-Instruct as the backbone Large Reasoning Model (LRM) for both the baseline Search-o1 and our improved architecture. The hyperparameters were configured as follows: we set `max_search_limit=10` attempts per question, maximum reasoning turns (`max_turn=15`), top-k retrieved documents (`top_k=10`), and maximum document length (`max_doc_len=3000`).

Quantitative performance was evaluated using two primary metrics: **Accuracy** (whether the ground truth appears anywhere in the prediction) and **F1 Score** (token overlap between the ground truth and the prediction).

For every evaluation batch, the framework automatically generated three JSON logging files, which are saved in our GitHub repository under the `/outputs` directory:

- `<sample_id_range>_metrics.json`: Captures the quantitative scores (e.g., Exact Match, Accuracy, F1, etc.).
- `<sample_id_range>_info_extract.json`: Documents the complete extraction history (e.g., prompts, retrieved documents, raw output, and extracted information) from the Reason-in-Documents module.
- `<sample_id_range>.json`: Documents the prompts, the original user question, the agent’s full step-by-step reasoning trajectory (including reflection analysis and hallucination detection), and the final generated answer.

3 Results

3.1 Quantitative Performance

Table 1 compares the performance of both models across the 120 HotpotQA evaluation samples. Our architecture achieved an overall accuracy of 0.433 and an F1 score of 0.458, showing a quantitative improvement over the baseline’s 0.292 accuracy and 0.363 F1 score. When breaking the results down into batches of ten, our enhanced framework matched or outperformed the baseline’s accuracy in 11 out of the 12 groups and F1 score in 8 out of the 12 groups.

Table 1: Comparison of model performance across sample ID ranges.

Sample ID Range	Our Model Acc	Baseline Acc	Our Model F1	Baseline F1
0-9	0.700	0.200	0.729	0.325
10-19	0.200	0.100	0.200	0.319
20-29	0.300	0.300	0.300	0.300
30-39	0.400	0.500	0.430	0.533
40-49	0.600	0.100	0.517	0.100
50-59	0.300	0.200	0.325	0.231
60-69	0.400	0.200	0.570	0.363
70-79	0.500	0.500	0.500	0.529
80-89	0.400	0.200	0.460	0.353
90-99	0.700	0.500	0.750	0.617
100-109	0.300	0.300	0.262	0.326
110-119	0.400	0.400	0.450	0.365
Average	0.433	0.292	0.458	0.363

To further evaluate the impact of our architectural refinement, we tracked the frequency of search tool invocations across the 120 evaluation samples. The baseline Search-o1 made a total of 137 search calls, averaging 1.14 searches per query. In contrast, our refined agent initiated 260 search calls, resulting in an average of 2.17 searches per query. This represents a 90.3% increase in the average number of search tool utilizations per question on the multi-hop HotpotQA dataset.

We also measured the interventions triggered by our newly added modules. The *Hallucination Check* successfully corrected 12 hallucination attempts, forcing the model to verify information through an additional search cycle rather than relying on unverified internal knowledge. Furthermore, the *Extraction Refinement Loop* recovered lost information from relevant documents in 4 instances where the *Reason-in-Documents* module initially reported “No helpful information found.”

These interventions, combined with the overall increase in search frequency, demonstrate that our refined architecture encourages the agent to complete the necessary multi-hop retrieval process rather than relying on premature or unverified guesses.

3.2 Qualitative Analysis and Case Studies

3.2.1 Hallucination Detection

To demonstrate the effectiveness of the Hallucination Detection module, we analyzed a test case asking: “What screenwriter with credits for ‘Evolution’ co-wrote a film starring Nicolas Cage and Téa Leoni?”

In the baseline model (Figure 4), the agent relied entirely on its internal knowledge rather than utilizing the search tool. This led to a hallucination where the model incorrectly claimed that *Evolution* was a 1996 film directed by Alex Proyas and written by Charles

Leavitt. Because it never initiated a search to verify these claims, it ultimately output the wrong final answer.

On the other hand, our improved agent (Figure 3) successfully mitigated this failure mode. Initially, the model also attempted to guess the answer, incorrectly identifying Michael Chabon. However, the Hallucination Detection module intercepted this unverified claim and appended a system message directly into the reasoning stream:

“[System: Your answer contains unverified factual claims. You must perform a search to verify them].”

This intervention stopped the premature output and forced the model to generate a search query. After retrieving the actual web data, the model correctly identified David Diamond and David Weissman as the true screenwriters, showing the mechanism’s ability to correct hallucination behaviors in real time.

```
d think step by step to solve it. Provide your final answer in the format \boxed{YOUR_ANSWER}. Question: What screenwriter with credits for "Evolution" co-wrote a film starring Nicolas Cage and Téa Leoni? <|im_end> <|im_start>assistant "
answer [ ] 1 item
Output "To answer this question, let's break it down into steps: 1. Identify the film "Evolution" and its screenwriter. 2. Find the screenwriter of the film starring Nicolas Cage and Téa Leoni. 3. Verify if the screenwriter of "Evolution" is the same as the one who co-wrote the Nicolas Cage and Téa Leoni film. Step 1: Identify the film "Evolution" and its screenwriter. - "Evolution" is a 2001 science fiction thriller film starring Rosario Dawson and Clive Owen. - The screenwriter of "Evolution" is Michael Chabon, an American novelist. Step 2: Find the screenwriter of the film starring Nicolas Cage and Téa Leoni. - "Evolution" does not feature Nicolas Cage and Téa Leoni. - Let's look for a film that does have these actors and find its screenwriter. Step 3: Verify if the screenwriter of "Evolution" is the same as the one who co-wrote the Nicolas Cage and Téa Leoni film. - "Unbreakable" (2000) is a film starring Nicolas Cage and starring Téa Leoni in a supporting role. - The screenwriter of "Unbreakable" is Drew Goddard, an American screenwriter and director. Now, we need to check if Michael Chabon co-wrote "Unbreakable" with Drew Goddard. - After a search, it appears that Michael Chabon did not co-write "Unbreakable" with Drew Goddard. Drew Goddard wrote the screenplay alone. Since Michael Chabon did not co-write "Unbreakable," we need to find another film. - "Adaptation" (2002) is a film starring Nicolas Cage and Téa Leoni. - The screenwriter of "Adaptation" is Charlie Kaufman, an American screenwriter and filmmaker. Let's check if Charlie Kaufman co-wrote "Evolution" with Michael Chabon. - After a search, it appears that Charlie Kaufman and Michael Chabon did co-write "Adaptation," but not "Evolution." Since "Evolution" does not have Nicolas Cage and Téa Leoni, and Michael Chabon did not co-write "Adaptation" with Charlie Kaufman, we need to look for another film. - "The Science of Sleep" (2006) is a film starring Nicolas Cage and Téa Leoni. - The screenwriter of "The Science of Sleep" is Charlie Kaufman. After checking, it seems there is no film that meets all the criteria: featuring Nicolas Cage and Téa Leoni, and co-written by Michael Chabon. Therefore, the screenwriter with credits for "Evolution" who co-wrote a film starring Nicolas Cage and Téa Leoni is not found based on the given information. \boxed{Michael Chabon} [System: Your answer contains unverified factual claims. You must perform a search to verify them.] <|begin_search_query|>screenwriter of "Evolution" co-wrote a film starring Nicolas Cage and Téa Leoni<|end_search_query|> <|begin_search_result|>David Diamond and David Weissman co-wrote the DreamWorks Pictures sci-fi comedy Evolution, directed and co-produced by Ivan Reitman. Nicolas Cage and Téa Leoni starred in the film Evolution.<|end_search_result|> [Observation: Useful information or candidates found. Proceed with verification or next step.] Assistant: Based on the search results, David Diamond and David Weissman co-wrote the film "Evolution," which starred Nicolas Cage and Téa Leoni. Since we need to identify the screenwriter of "Evolution" who also co-wrote a film starring Nicolas Cage and Téa Leoni, we should verify if David Diamond and David Weissman co-wrote another film with these actors. <|begin_search_query|>David Diamond and David Weissman co-wrote a film starring Nicolas Cage and Téa Leoni<|end_search_query|> <|begin_search_result|>The web pages provide information that confirms David Diamond and David Weissman co-wrote the film "Evolution," which starred Nicolas Cage and Téa Leoni. This aligns with the previous reasoning steps that identified David Diamond and David Weissman as the screenwriters of "Evolution" and noted that it starred Nicolas Cage and Téa Leoni. Therefore, the search query is supported by the information found in the web pages.<|end_search_result|> [Observation: Useful information or candidates found. Proceed with verification or next step.] Assistant: Based on the search results, David Diamond and David Weissman co-wrote the film "Evolution," which starred Nicolas Cage and Téa Leoni. Therefore, the screenwriter with credits for "Evolution" who co-wrote a film starring Nicolas Cage and Téa Leoni is David Diamond and David Weissman. \boxed{David Diamond and David Weissman}"
Pred_Answer "David Diamond and David Weissman"
```

Figure 3: Model output from our improved agent. The highlighted line shows the system message appended to the reasoning stream instructing the model to verify its claims through a search.

```
Question: What screenwriter with credits for "Evolution" co-wrote a film starring Nicolas Cage and Téa Leoni? <|im_end> <|im_start>assistant "
answer [ ] 1 item
Output "To solve this problem, let's break it down step by step: 1. **Identify the film "Evolution":** - "Evolution" is a 1996 science fiction film directed by Alex Proyas. - It stars Nicolas Cage and Téa Leoni. 2. **Determine the screenwriter for "Evolution":** - The screenwriter for "Evolution" is Charles Leavitt. 3. **Identify screenwriters who have co-written films with Charles Leavitt:** - Charles Leavitt has co-written several films. One notable film he has co-written with other screenwriters is "The Crow" (1994). 4. **Check if "The Crow" fits the criteria:** - "The Crow" stars Brandon Lee and Sinbad. - While "The Crow" does not have Nicolas Cage and Téa Leoni in the cast, it is a film that Charles Leavitt has co-written. Given that Charles Leavitt is the screenwriter for "Evolution" and has co-written "The Crow," the screenwriter we are looking for must be Charles Leavitt. Therefore, the screenwriter with credits for "Evolution" who co-wrote a film starring Nicolas Cage and Téa Leoni is \boxed{Charles Leavitt}."
Pred_Answer "Charles Leavitt"
```

Figure 4: Model output from the baseline agent. The model did not perform any searches to verify its information and therefore produced an incorrect final answer.

3.2.2 Content Judge and Reflection

To demonstrate the impact of the Content Judge and Reflection module, we examined the query: “Kaiser Ventures Corporation was founded by an American industrialist who became known as the father of modern American shipbuilding?”

Without the reflection and hallucination mechanisms, the baseline agent (Figure 6) failed to engage with the search tool. Instead, it relied on inaccurate internal knowledge and incorrectly guessed “Andrew Carnegie” as the final answer.

In contrast, our improved agent (Figure 5) used the search tool together with the Judge and Reflection mechanism to navigate search dead ends. The agent correctly initiated a search and identified Henry J. Kaiser as the “father of modern American shipbuilding.” However, its follow-up searches to connect him to Kaiser Ventures initially returned no helpful information. Rather than guessing, the reflection mechanism intervened by appending a structured reflection message into the reasoning stream:

“[Observation: The search results were limited. Suggestion: ...]”

This reflection analyzed why the query failed, identified the missing information gap, and suggested a more effective next query. Guided by this feedback, the agent successfully adjusted its search queries, retrieved the necessary evidence, and received a positive observation message indicating that useful information had been found. This allowed the model to confidently produce the correct final answer.

```
d think step by step to solve it. Provide your final answer in the format \boxed{YOUR_ANSWER}. Question: Kaiser Ventures corporation was founded by an American industrialist who became known as the father of modern American shipbuilding? <|im_end> <|im_start>-assistant "
```

answer [] 1 item

Output "To answer this question, let's break it down into steps and search for relevant information: 1. **Identify the American industrialist who became known as the father of modern American shipbuilding.** - We need to find out who this person was. 2. **Determine if Kaiser Ventures corporation was founded by this industrialist.** - Once we identify the industrialist, we need to verify if Kaiser Ventures was indeed founded by them. Let's start with step 1 by searching for the person known as the father of modern American shipbuilding. <|begin_search_query>father of modern American shipbuilding industrialist<|end_search_query> <|begin_search_result>Henry J. Kaiser, born on May 9, 1882, in Sprout Brook, New York, was an American industrialist who became known as the father of modern American shipbuilding. According to various sources, Kaiser played a significant role in the construction of numerous infrastructure projects, including roads, dams, and highways. His company, the Henry J. Kaiser Company, was established in 1914 and later grew to become a major player in the construction and shipbuilding industries. In 1931, Kaiser's company won a contract to build the Hoover Dam, demonstrating his capabilities in large-scale engineering projects. His company, Kaiser Shipyards, was instrumental in building Liberty ships during World War II, contributing significantly to the war effort.<|end_search_result> [Observation: Useful information or candidates found. Proceed with verification or next step.] From the search results, it is clear that Henry J. Kaiser was the industrialist known as the father of modern American shipbuilding. Now, let's move to step 2 and verify if Kaiser Ventures corporation was founded by Henry J. Kaiser. <|begin_search_query>Kaiser Ventures corporation founder Henry J. Kaiser<|end_search_query> <|begin_search_result>No helpful information found.<|end_search_result> [Observation: The search results were limited. Suggestion: Analysis: [Why it failed and what we need] - **Why it failed:** The search did not yield helpful information, likely due to the specificity of "Kaiser Ventures corporation" and the fact that "Henry J. Kaiser" is a prominent American industrialist, but the information specifically focuses on his founding of the shipbuilding industry, not his relationship to a venture capital firm. This might suggest that either the search terms need to be more specific to Kaiser Ventures, or the focus should be on Henry J. Kaiser's broader industrial influence. - **What we need:** A more targeted query to clarify who founded Kaiser Ventures and also find information about Henry J. Kaiser's broader industrial influences and ventures. Alignment: Fixing this will help us answer the original User Question by first confirming Henry J. Kaiser's role in the founding of Kaiser Ventures and then potentially expanding to find information about his other ventures that might contribute to his status as the "father of modern American shipbuilding." Next_Query: "Henry J. Kaiser founder Kaiser Ventures shipbuilding industry"] <|begin_search_query>Henry J. Kaiser founder Kaiser Ventures shipbuilding industry<|end_search_query> <|begin_search_result>No helpful information found.<|end_search_result> [Observation: The search results were limited. Suggestion: ### Analysis: The search failed because the terms used in the previous search are not directly aligned with the information we need. The query "Henry J. Kaiser founder Kaiser Ventures shipbuilding industry" might be too specific and might not yield the right results. Additionally, "Kaiser Ventures" might not be the primary focus, and "shipbuilding industry" alone might not pinpoint the right person. The user is asking about an American industrialist known as the "father of modern American shipbuilding" who founded Kaiser Ventures. This implies we need to focus on the industrialist, the shipbuilding aspect, and the founding of the corporation. However, the emphasis seems to be on identifying the industrialist first. ### Next_Query: "Industrialist known as the father of American shipbuilding founder of Kaiser Ventures" This query is more generalized and aligns better with the User's question. It breaks down the main points: - An industrialist - Known as the father of American shipbuilding - Founder of Kaiser Ventures This should help in finding the relevant information more effectively.] <|begin_search_query>Industrialist known as the father of American shipbuilding founder of Kaiser Ventures<|end_search_query> <|begin_search_result>Henry J. Kaiser is known as the father of modern American shipbuilding and played a significant role in the construction of numerous infrastructure projects, including roads, dams, and highways. His company, Kaiser Shipyards, built Liberty ships during World War II, contributing significantly to the war effort.<|end_search_result> [Observation: Useful information or candidates found. Proceed with verification or next step.] Based on the search results, Henry J. Kaiser is indeed the industrialist known as the father of modern American shipbuilding and he founded Kaiser Shipyards, which built Liberty ships during World War II. Given that Kaiser Ventures corporation was established in 1931 and is known for its involvement in various ventures, it's plausible that Henry J. Kaiser was also involved in founding or overseeing the company. To conclude, Henry J. Kaiser, the industrialist known as the father of modern American shipbuilding, founded Kaiser Ventures corporation. Final Answer: \boxed{Henry J. Kaiser}"

Pred Answer "Henry J. Kaiser"

Figure 5: Model output from our improved agent. The highlighted “Observation” marks the reflection message appended to the reasoning stream to guide subsequent search steps.

```

Please answer the following question. You should think step by step to solve it. Provide your final answer in the format \boxed{YOUR_ANSWER}. Question: Kaiser Ventures corporation was founded by an American industrialist who became known as the father of modern American shipbuilding? <jim_end> <jim_start>assistant "
answer [ ] 1 item
Output "Let's break down the question and solve it step by step. 1. **Identify the corporation and its founder**: The question mentions Kaiser Ventures Corporation and an American industrialist known as the "father of modern American shipbuilding." 2. **Determine the industrialist**: The industrialist in question is known for founding a shipbuilding company that significantly influenced the industry. A prominent figure who fits this description is Andrew Carnegie. 3. **Verify the connection**: Andrew Carnegie is indeed known for founding the Carnegie Steel Company, which later became U.S. Steel, and he is often referred to as the "father of modern American shipbuilding" due to his significant contributions to the industry. 4. **Confirm the corporation**: Kaiser Steel Corporation was founded by William A. Kaiser, not Andrew Carnegie. However, the question specifically mentions Kaiser Ventures Corporation, which could be a later or different entity. 5. **Conclusion**: Given the information, the American industrialist known as the "father of modern American shipbuilding" and who founded a shipbuilding company is Andrew Carnegie. Thus, the answer to the question is: \boxed{Andrew Carnegie}"
Pred_Answer "Andrew Carnegie"

```

Figure 6: Model output from the baseline agent. Without reflection or hallucination detection, the agent hallucinated an incorrect final answer without performing any searches.

3.2.3 Extraction Refinement Loop

To demonstrate the value of the Extraction Refinement Loop, we analyzed the query: “What nationality were social anthropologists Alfred Gell and Edmund Leach?”

Initially, the Reason-in-Documents module scanned the retrieved documents but incorrectly concluded that there was no explicit mention of either person’s nationality. However, inspection of the raw document text confirms that the biographical summaries stated both individuals as “British” (Figure 8).

Instead of accepting this extraction failure, the Extraction Refinement Loop recognized the potential false negative and prompted the model to re-evaluate the retrieved documents. During the second pass, the module successfully recovered the overlooked details (Figure 9). The refined extraction updated the reasoning stream with the correct information that both Gell and Leach were British. By recovering this missed evidence, the model bypassed the earlier extraction failure and produced the correct final answer.

```

raw_output "***Final Information** No helpful information found. Upon reviewing the web pages, there is no explicit mention of the nationalities of Alfred Gell and Edmund Leach. The provided pages discuss their educational backgrounds, careers, and some of their works but do not contain information specifically about their nationalities."
extracted_info "No helpful information found. Upon reviewing the web pages, there is no explicit mention of the nationalities of Alfred Gell and Edmund Leach. The provided pages discuss their educational backgrounds, careers, and some of their works but do not contain information specifically about their nationalities."

```

Figure 7: Initial extraction output where the model incorrectly concluded that the documents did not contain the nationalities of Alfred Gell and Edmund Leach.

```

anthropology]]\n[Request Permissions icon link]\n\n* [Show page numbers Hide page numbers]\n\nEdmund Ronald Leach (1910–1989) was one of the most emphatic and colorful figures in modern social anthropology. He spent much of his time writing for and speaking to lay audiences and so became perhaps the most widely known British anthropologist of his generation. Writing on an improbably vast range of topics, between 1937 and 1988, Leach published nine books, four edited volumes, more than 50 scholarly articles, and hundreds of shorter pieces. Although he is remembered primarily as the author of the Seven Years of Gladstone and Obituary of Gladstone, his intellectual contributions are far broader.

\n* [e]\n\n**Alfred Antony Francis Gell**, [FBA] ([/g e l/]; June 12, 1945 – January 28, 1997) was a British social [anthropologist] whose most influential work concerned [art], language, symbolism and [ritual]. He was trained by [Edmund Leach] (MPhil, [Cambridge University]) and [Raymond Firth] (PhD, [London School of Economics])[[1]] and did his fieldwork in [Melanesia] and tribal [India]. Gell taught at the London School of Economics, among other places. He was also a [Fellow of the British Academy]. He died of cancer in 1997, at the age of 51.[[2]]\n\n**Take a short survey and help us improve Wikipedia**\n\nNo thanks Visit survey\n\n* * *\n\nRead our [privacy statement]. To stop Quick Surveys from appearing, [change your preferences].\n\nThought\n--\n\n\n[[edit]]\n\nIn his posthumously published 1998 book Art and Age

```

Figure 8: Snippets from the raw document text showing that the nationalities were present but missed during the first extraction pass.

```

answer in the format \boxed{YOUR_ANSWER}. Question: What nationality were social anthropologists Alfred Gell and Edmund Leach? <|jim_
end|> <|jim_start|>assistant "
answer [] 1 item
Output "To determine the nationalities of Alfred Gell and Edmund Leach, I will first look up their biographical information. Assistant thinking s
tps: 1. Alfred Gell 2. Edmund Leach Assistant: <|begin_search_query|>Alfred Gell nationality<|end_search_query|> <|begin_search_result|
>No helpful information found.<|end_search_result|> [Observation: The search results were limited. Suggestion: Analysis: [Why it failed and
d what we need] **Why it failed and what we need:** The search failed because the specific search terms "Alfred Gell nationality" did not yi
eld relevant results. This could be due to Alfred Gell not being widely recognized by nationality alone, or the search not targeting the exact i
nformation needed. Splitting the query into more specific parts might help, focusing on Alfred Gell's nationality while also providing context
about the other social anthropologist, Edmund Leach. **How fixing this helps answer the original User Question:** By providing information
about Alfred Gell's nationality and potentially Edmund Leach's nationality, we can directly answer the User's question about the nationalitie
s of these social anthropologists. **Next_Query:** Nationality of Alfred Gell and Edmund Leach as social anthropologists] Assistant: <|begi
n_search_query|>Nationality of Alfred Gell and Edmund Leach as social anthropologists<|end_search_query|> <|begin_search_result|>No h
elpful information found.<|end_search_result|> [Observation: The search results were limited. Suggestion: Analysis: [Why it failed and what
we need] The search failed because the results were either not specific enough or did not contain the nationality information for Alfred Gell
and Edmund Leach individually, or they might not have been social anthropologists. To address this, we need to ensure that the search is fo
cused on these two individuals specifically and that we are asking for their nationalities. Next_Query: Nationality of Alfred Gell and Edmund
Leach] Assistant: <|begin_search_query|>Nationality of Alfred Gell and Edmund Leach<|end_search_query|> <|begin_search_result|>Final
Information** The Nationality of Alfred Gell is British. Edmund Leach, who influenced Gell, was British as well.<|end_search_result|> [Obse
rvation: Useful information or candidates found. Proceed with verification or next step.] The search results indicate that both Alfred Gell an
d Edmund Leach were British. Therefore, we can confidently answer the User's question. Final Answer: \boxed{British}'
Pred_Answer "British"

```

Figure 9: Output from our improved agent. The highlighted text shows the correctly re-extracted information produced by the refinement loop.

4 Discussion

Our architectural refinements to the Search-o1 agent proved ultimately successful. Quantitatively, we achieved both a higher f1 and accuracy score than the original baseline evaluation as reported above. This means that the correct answer appeared in the output prediction at a higher rate with our additions than without and that there was a higher token overlap between our model’s prediction and the correct answer. In terms of qualitative evaluation, our hallucination check proved valuable, correcting the model to search externally rather than rely on internal data and deriving incorrect answers. Furthermore, our Content Judge and Reflection Module detected if retrieved information was limited, and redirected the model to conduct a more accurate search utilizing reflective prompt engineering. Both implementations resulted in the model correctly answering the question as compared to the baseline which failed to do so. Finally, we measured the change in the number of search tool-calls, with our model initiating more on average than the baseline. This increase in tool utilization encouraged by our model, produces better answers to multi-hop questions. Our improved Search-o1 agent fared better than the baseline in both quantitative and qualitative metrics of evaluation, despite facing some obstacles.

During the duration of this project we faced some limitations inhibiting us from making the Search-o1 agent as accurate as possible. When utilizing API keys, we had a limited number of tokens available to have the agent connect externally to fetch and retrieve information. This led to us evaluating a lesser number of trials on a smaller batch size. Furthermore, our addition of the content judge and reflection loop significantly increased the latency time and, thus, was slightly more computationally expensive. This resulted in our individual computers taking several minutes to execute the code and for the search agent to output an answer. Despite these limitations, our addition to the Search-o1 agent yielded promising results that could be further developed for more robust, reliable tool-usage.

The possible future extensions of this project are substantial. With broader token access,

we would be able to conduct evaluations on larger, more diverse sets of data. The Content Judge could be made more robust to produce more specific diagnostic labels about why the search agent failed. This would allow the reflection prompt to curate a prompt designed to directly address the issue, which could result in more productive search calls. Furthermore, we could develop the agent to extend to multi-hop assistant tasks that chain together heterogeneous tools. This would allow the agent to autonomously respond to requests such as “Scan my email and add meetings to calendar” employing external tools calls for complex tasks rather than information retrieval.

5 Conclusion

In this project, we explored how reflection and evaluation mechanisms improve the behavior of tool-use search agents for multi-hop question answering. Starting from the Search-o1 framework, we analyzed several common failure cases, including unsupported answers without sufficient retrieval, missed information in retrieved documents, and ineffective query planning during multi-step search. These observations motivated us to design mechanisms that help the agent better regulate its search and reasoning process.

To address these problems, we introduced several components into the pipeline. A hallucination detection step encourages the agent to rely on retrieved evidence rather than answering directly from an internal knowledge base. We also implemented retrieval and content judges to evaluate whether search results and extracted information are relevant to the question. In addition, a reflection module allows the agent to revise search queries when previous retrieval attempts are not helpful, which enables more relative follow-up searches. These mechanisms help the agent better retrieve information and make more informed decisions about whether to continue searching or finalize an answer.

We evaluated our approach on a subset of the HotpotQA dataset using the Qwen2.5-3B-Instruct model. Compared with simpler baselines, our system demonstrates improved accuracy (0.292 to 0.433), F-1 score (0.363 to 0.458), and numbers of searches per query (1.14 to 2.17). These improvements help to recover concrete information from retrieved documents and generate more effective follow-up queries. These improvements suggest that structured judgment, reflection, and evaluation processes are crucial in improving the reliability of search-based reasoning agents.

It is possible that future work could extend our project in several directions. Running experiments on larger datasets without our token limit would have a more comprehensive evaluation of the system’s performance. Replacing prompt-based judges with more robust or complex structured algorithms may further improve decision-making capacity. Finally, exploring more advanced retrieval strategies could help the agent to confirm evidence more efficiently in complex multi-hop tasks, and eventually be deployed for task completion.

6 Contributions

Arshia

- Attempted to implement components for the Phase 1 pipeline, including a basic judge and reflection prompting function for content refinement (Gate 3).
- Collaborated with Joyce to correct and update the Gate 3 functions to improve overall accuracy and ensure the generated answers met the required output format.
- Contributed to the report by documenting the design and implementation of Gate 3's judge, reflect, and refine components.
- Contributed to the Discussion section, summarizing results, outlining system limitations, and proposing future improvements to the agent.
- Contributed to the final project poster, working on the domain introduction and contextual background.

Megha

- Implemented the initial judge function (Gate 1), determining where document-level judgments should occur within the pipeline.
- Ensured that each document retrieval call was evaluated properly, first using a heuristic-based evaluation approach and later transitioning to a Large Reasoning Model (LRM) approach.
- Experimented with different judgment and reflection strategies and ran full dataset evaluations to improve the final performance metrics.
- Contributed to the Introduction section of the report.
- Managed report formatting in Overleaf and migrated the document from Google Docs to Overleaf.
- Contributed to the design of the project poster and authored the following sections: Data, Methods, Qualitative Results, Quantitative Results, Conclusion, Impact, and References.

Joyce

- Implemented multiple components for the Phase 1 pipeline, including the Retrieval Judge and Reflection module (Gate 1), the Hallucination Detection module, the Extraction Refinement Loop, and the Content Judge and Reflection module (Gate 2).
- Led architectural refinements in Phase 2 by identifying bottlenecks in the pipeline (Gate 1 and Gate 3) and optimizing system performance by refining the Content Judge and Reflection logic (Gate 2).
- Contributed to the Methods section of the report, documenting the baseline framework and the iterative development process across Phase 1 and Phase 2.
- Conducted the final evaluation on the HotpotQA dataset and contributed to the Results section documenting both quantitative and qualitative findings.

Jiixin

- Implemented the initial reflective functions for Gate 1 and contributed to improving prompt structure and reflective reasoning in Phase 1.
- Collaborated with Megha to integrate the judge and reflection modules using an

initial heuristic evaluation approach.

- Contributed to the Abstract section of the report.
- Contributed to the Conclusion section of the website.
- Participated in the overall project design and system implementation.

References

- Li, Xiaoxi, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. “Search-o1: Agentic Search-Enhanced Large Reasoning Models.”
- Yang, An, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, and ... 2024. “Qwen2 Technical Report.”
- Yang, Zhilin, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. “HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering.” *arXiv preprint arXiv:1809.09600*. [\[Link\]](#)