Improving Multi-Hop Reasoning and Fact Verification with Consistency-Ranked ReACT Agents

Rania Mani
ENSAE Paris
rania.mani@ensae.fr

Abstract

Retrieval-Augmented Generation (RAG) pipelines enhance large language models (LLMs) by grounding their responses in retrieved evidence. However, even with high-quality documents, LLMs often struggle to generate consistent and accurate answers, especially in multi-step tasks such as multi-hop question answering and fact verification. In this work, we target this generation bottleneck by combining ReACT prompting (interleaving reasoning and actions) with CRAG (Consistency-Ranked Augmented Generation), a post-hoc selection mechanism that ranks multiple reasoning trajectories based on logical coherence. Our method fixes the retrieval step and focuses exclusively on improving the generation layer in RAG. We evaluate our system on HotpotQA and FEVER, showing consistent gains of 5–6 EM points over baseline ReACT prompting. These improvements are achieved without fine-tuning and with modest overhead, demonstrating the effectiveness of consistency-based filtering for enhancing factuality and reasoning in RAG setups.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities in solving complex NLP tasks, especially when coupled with external tools or context via Retrieval-Augmented Generation (RAG). In RAG, a fixed retrieval step supplies evidence passages, and the LLM is tasked with generating a coherent and correct answer based on these documents. While retrieval often succeeds at locating relevant information, the generation step remains a bottleneck: models hallucinate facts, construct logically inconsistent reasoning paths, or produce unsupported conclusions.

This challenge is particularly evident in multi-hop question answering and fact verification, where reasoning must integrate multiple pieces of evidence. Prompting strategies such as Chain-of-Thought (CoT) offer partial relief by encouraging step-by-step explanations, but often fail in high-noise or ambiguous settings.

To address this, ReACT (Reasoning + Acting) introduces a more structured approach: the model alternates between natural language thoughts and tool-based actions, e.g., search or lookup. This design brings reasoning and evidence retrieval into a dynamic loop. However, ReACT alone cannot guarantee consistency or factuality, especially when external tools return incomplete or noisy results.

We propose to enhance ReACT within a fixed-retrieval RAG pipeline by integrating CRAG (Consistency-Ranked Augmented Generation). Instead of relying on a single generated trajectory, CRAG samples multiple completions and scores them based on internal coherence, factual alignment, and absence of contradictions. This post-hoc selection step improves the robustness of the generation process without requiring retraining or new data.

Our contributions are as follows:

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

- We identify generation inconsistency as a primary weakness in RAG pipelines for multi-hop reasoning and fact verification.
- We combine ReACT prompting with CRAG filtering to improve generation quality without modifying retrieval or fine-tuning the base model.
- We demonstrate consistent improvements on HotpotQA and FEVER using GPT-3.5, achieving up to 6 EM points over the ReACT baseline.

2 Related Work

2.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) [Lewis et al., 2020] enhances language models by combining them with external knowledge retrieval systems, enabling responses grounded in contextually relevant documents. As illustrated in Figure 1, a standard RAG system consists of two phases: (1) offline indexing, where documents are chunked, embedded, and stored in a vector database, and (2) online querying, where user questions are matched to relevant documents using semantic search. These retrieved documents are then injected into the prompt given to the language model for answer generation.

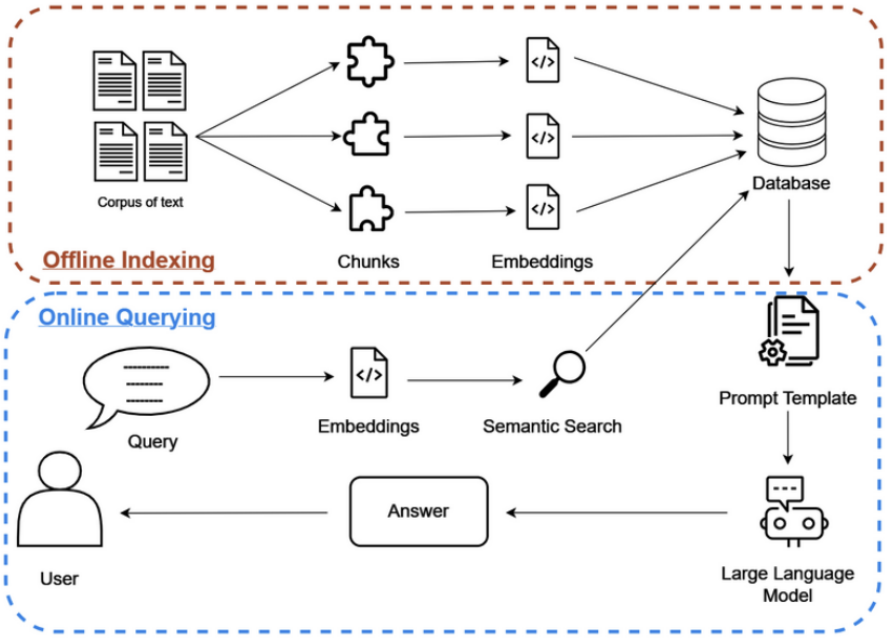

Figure 1: Standard RAG pipeline. In offline indexing, documents are chunked and embedded into a database. During querying, relevant chunks are retrieved and injected into the prompt sent to the language model. Our work assumes fixed retrieval and focuses on improving generation.

While RAG pipelines improve factual grounding, the generation stage remains a major source of hallucination and inconsistency—especially in tasks requiring multi-hop reasoning or synthesis across multiple evidence spans. Prior research has explored retrieval-aware decoding strategies [Izacard and Grave, 2020, Guu et al., 2020], but the generation module is often treated as a black box. In contrast, we focus entirely on enhancing the generation step using structured agentic reasoning and post-hoc consistency filtering.

2.2 Chain-of-Thought and ReACT Prompting

Chain-of-Thought (CoT) prompting [Wei et al., 2022] improves multi-step reasoning by guiding large language models to produce intermediate reasoning steps before delivering a final answer. This decomposition helps models handle arithmetic, logic, and multi-hop questions more systematically.

However, CoT assumes all required knowledge is present in the prompt and does not support dynamic access to external information—leading to potential hallucinations and brittle logic when faced with incomplete context.

To address this, ReACT [Yao et al., 2023] interleaves natural language reasoning ("thoughts") with external tool use ("actions") such as web search or document lookup. This agentic prompting framework enables models to retrieve new information mid-reasoning and adapt their reasoning paths based on observations. ReACT has demonstrated improved factual accuracy and interpretability in tool-augmented tasks. Nevertheless, its generated trajectories can still contain inconsistencies, redundancies, or premature halting—particularly when the retrieval results are noisy or ambiguous.

2.3 Consistency-Ranked Generation (CRAG)

Consistency-Ranked Augmented Generation (CRAG) [Yan et al., 2024] improves generation robustness by sampling multiple reasoning trajectories and ranking them based on logical consistency. Inspired by self-consistency methods [Wang et al., 2023], CRAG introduces a lightweight scoring function that rewards coherence, factual alignment, and absence of contradiction.

Originally proposed in generic retrieval and generation tasks, we adapt CRAG to enhance ReACT-style agentic prompting. By generating multiple trajectories and filtering them based on consistency criteria, CRAG enables the rejection of flawed reasoning chains without requiring additional model training. This makes it a natural fit for improving the generation component in fixed-retrieval RAG pipelines.

2.4 Reasoning Benchmarks

We evaluate our method on two representative benchmarks designed for retrieval-based reasoning: HotpotQA [Yang et al., 2018] and FEVER [Thorne et al., 2018]. HotpotQA requires multi-hop reasoning across multiple supporting documents, testing the model's ability to synthesize distributed evidence. FEVER focuses on fact verification, where the model must support or refute a claim based on retrieved Wikipedia passages.

Both benchmarks provide realistic scenarios where the retrieval step is separate from the reasoning step—aligning well with our focus on generation-stage improvements. They serve as rigorous testbeds for evaluating how CRAG-enhanced ReACT improves consistency and factual accuracy in retrieval-augmented generation tasks.

3 Problem Setup

3.1 Task Definitions

This work focuses on multi-step reasoning tasks that require both information retrieval and logical inference. We evaluate our approach on two standard datasets:

Dataset	Task Type	Input	Output Format	Challenge Type
HotpotQA	Multi-hop QA	Question q	Answer string a	Multi-hop reasoning
FEVER	Fact verification	Claim c, Evidence E	Label $y \in \{\text{SUPPORTS}, \text{REFUTES}, \text{NEI}\}$	Fact checking

Table 1: Overview of the datasets used for multi-step reasoning evaluation.

In both cases, the model must combine retrieved evidence with internal reasoning steps to produce accurate outputs.

Note: Our contribution focuses solely on the *generation and reasoning phase* of the RAG (Retrieval-Augmented Generation) pipeline. We assume the evidence is already retrieved (via a Wikipedia API or FAQ search) and do not modify the retriever. Our goal is to improve *how the LLM reasons over the retrieved content* through better prompting.

3.2 Problem Formulation

Let $\mathcal{M}$ be a large language model that takes an input x and produces an output $\hat{y}$ via a sequence of reasoning and tool-usage steps. The objective is to enhance the accuracy and consistency of $\mathcal{M}$'s predictions using prompt-level modifications only.

We define a ReACT agent as a function $\mathcal{A}_{\text{ReACT}}(x) \rightarrow \hat{y}$, which alternates between thoughts and actions. Our hybrid agent is defined as:

$$\mathcal{A}_{\text{ReACT+CRAG}}(x) = \arg \max_{i \in [1,N]} \text{Score}(T^{(i)}) \tag{1}$$

where $T^{(i)}$ is the i-th candidate trajectory and $\text{Score}(\cdot)$ measures internal logical consistency. The agent selects the most coherent reasoning path from N samples generated using ReACT prompting.

3.3 Evaluation Metric

We use **Exact Match (EM)** as the primary metric, which measures whether the predicted answer matches the gold answer exactly:

$$\text{EM} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\hat{y}_i = y_i] \tag{2}$$

Here, $\hat{y}_i$ is the predicted answer, y_i is the ground truth, $\mathbb{1}[\cdot]$ is the indicator function, and N is the total number of examples.

This strict metric penalizes hallucinations and inconsistencies, making it ideal for assessing the impact of CRAG-enhanced prompting on reasoning quality.

4 Datasets

We evaluate our approach using two widely adopted reasoning benchmarks: **HotpotQA** and **FEVER**. Both tasks require the model to reason over external textual evidence and generate answers grounded in retrieved information.

4.1 Dataset Sources

HotpotQA is a multi-hop question answering dataset introduced by Yang et al. [2018]. Each question requires synthesizing information from multiple Wikipedia paragraphs. The dataset encourages deep reasoning, including bridge entity linking and comparative questions.

FEVER (Fact Extraction and Verification) was introduced by Thorne et al. [2018]. It consists of over 185,000 claims derived from Wikipedia. Each claim is labeled as *SUPPORTS*, *REFUTES*, or *NEI* (Not Enough Information), with associated evidence for verification.

Both datasets are commonly used to evaluate multi-step reasoning and factual consistency.

4.2 Preprocessing Pipeline

For scalability and reproducibility, we sample 500 examples from the validation set of each dataset. Preprocessing includes:

- Removing ambiguous or incomplete entries
- Formatting inputs into ReACT-compatible prompts
- Standardizing claims/questions and retrieved evidence
- Mapping FEVER labels to discrete classes

We use a basic Wikipedia API or FAQ search to simulate retrieval. No retriever optimization or fine-tuning is performed, ensuring that performance gains reflect only the reasoning and generation phase.

4.3 Dataset Statistics

Dataset	Examples	Avg. Input Length	Avg. Docs	Label Distribution
HotpotQA	500	∼18 tokens	2–3 paragraphs	N/A (short answer)
FEVER	500	∼15 tokens	2–5 sentences	34% Support, 33% Refute, 33% NEI

Table 2: Statistics of sampled validation subsets used for evaluation.

We do not modify model weights or apply domain adaptation. All prompts are fed to the same base language model (`gpt-3.5-turbo`) to ensure consistency across conditions.

5 Methodology

Our approach enhances the ReACT (Reasoning + Acting) prompting framework by integrating a consistency-aware selection step inspired by the CRAG (Consistency-Ranked Augmented Generation) method. This section outlines the ReACT prompting structure, our CRAG-based filtering mechanism, and implementation details.

5.1 ReACT Prompting Framework

ReACT agents generate solutions by interleaving natural language reasoning with tool-based actions. A single trajectory T consists of a sequence of thought-action-observation triples:

$$T = [(t_1, a_1, o_1), (t_2, a_2, o_2), \ldots, (t_k, a_k, o_k)] \tag{3}$$

where:

- t_i is a **thought**: a natural language reasoning step,
- a_i is an **action** (e.g., `Search`, `Lookup`, `Finish`),
- o_i is the **observation** received after executing a_i.

The model continues generating steps until it emits a `Finish` action, which produces the final output $\hat{y}$.

5.2 CRAG-Based Trajectory Filtering

To improve the reliability of ReACT agents, we integrate CRAG (Corrective Retrieval-Augmented Generation), which introduces a knowledge correction layer before final answer generation. As illustrated in Figure 2, CRAG first evaluates the quality of the retrieved documents using a retrieval evaluator, classifying them as correct, ambiguous, or incorrect. Depending on this classification, either a refinement step or a new retrieval is triggered before generation.

Once retrieval correction is applied, CRAG generates N independent ReACT trajectories and selects the one with the highest logical consistency. The final answer is given by:

$$\hat{y} = \arg \max_{i \in [1,N]} \text{Score}(T^{(i)}) \tag{4}$$

Here, $\text{Score}(\cdot)$ is a heuristic function that ranks each trajectory $T^{(i)}$ based on:

- Internal consistency between thoughts and actions,
- Absence of contradictions,
- Agreement between retrieved facts and final answer.

This post-hoc selection mechanism helps avoid common failure modes such as hallucination, looping, or early termination.

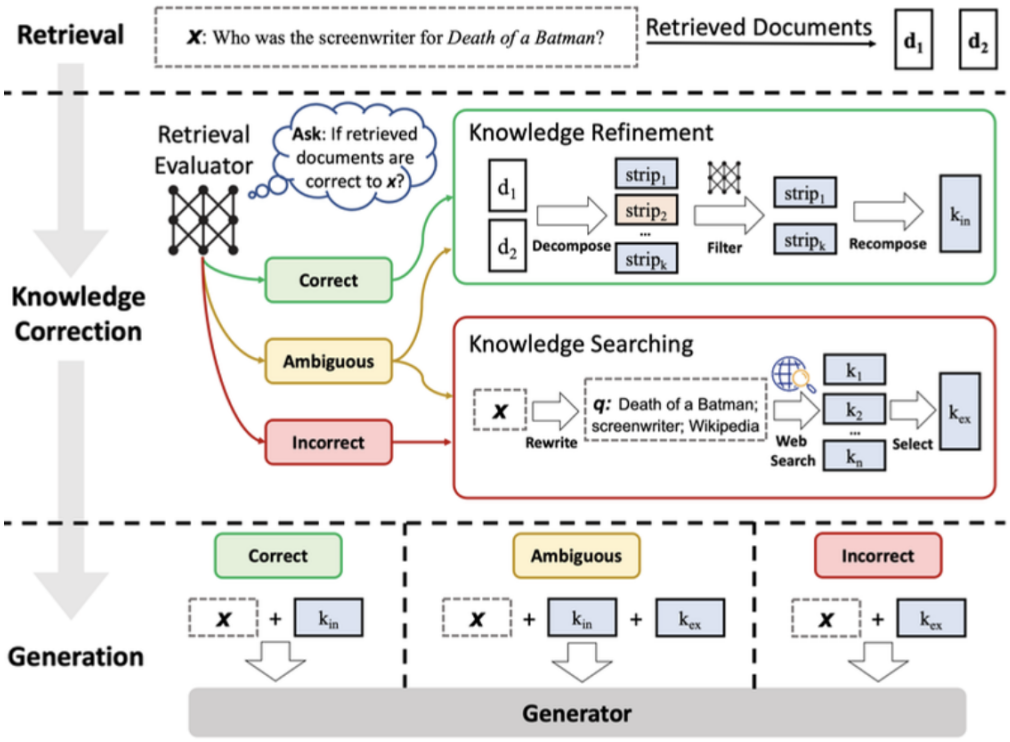

Figure 2: CRAG architecture for retrieval correction. Retrieved documents are evaluated and either refined or replaced based on their correctness before being passed to the generator. This improves alignment between evidence and the generated answer.

5.3 Implementation Details

We implement our agent using the `LangChain` framework, with access to a simple Wikipedia API. The available actions are summarized in Table 3.

Action	Description
`Search[e]`	Returns the first 5 sentences from the Wikipedia page for entity e.
`Lookup[s]`	Returns the sentence following string s in the retrieved context.
`Finish[a]`	Ends reasoning and returns the final answer a.

Table 3: Action space available to ReACT and ReACT+CRAG agents.

Each input prompt is processed $N = 5$ times to produce 5 trajectories, which are then scored using a custom consistency evaluator. The scoring mechanism penalizes:

- Repetitive reasoning loops,
- Contradictory thoughts or unsupported conclusions,
- Disagreements between evidence and final prediction.

5.4 System Summary

6 Experiments and Results

We conduct experiments to evaluate the effectiveness of our ReACT+CRAG agent compared to the baseline ReACT agent. The focus is on assessing improvements in reasoning consistency and factual accuracy on two widely used benchmarks: **HotpotQA** and **FEVER**.

Component	Description
Base model	`GPT-3.5-turbo` via OpenAI API
Prompting style	ReACT: Thought + Action + Observation
Tool interface	Wikipedia API (search + lookup)
Trajectories per input	$N = 5$ (CRAG setting only)
Selection strategy	Consistency-based post-hoc scoring

Table 4: Summary of model and system configuration.

6.1 Experimental Setup

All experiments are performed using the `GPT-3.5-turbo` model accessed via the OpenAI API. For each input query, the agent generates $N = 5$ independent reasoning-action trajectories. The CRAG post-hoc scoring function ranks these based on logical coherence and consistency with retrieved evidence. The top-ranked trajectory determines the final prediction $\hat{y}$. The available actions include `Search`, `Lookup`, and `Finish`, as described in Table 3.

We use the **Exact Match (EM)** metric to evaluate performance — defined as the percentage of model outputs that exactly match the ground-truth answer.

6.2 HotpotQA Performance

Method	EM Score (%)
GPT-3.5 + ReACT	19.2
GPT-3.5 + ReACT+CRAG	**25.0**

Table 5: Exact Match scores on HotpotQA.

ReACT+CRAG outperforms the baseline by **+5.8 EM points**, highlighting its advantage in multi-hop reasoning tasks that require integrating information from several sources.

6.3 FEVER Performance

Method	EM Score (%)
GPT-3.5 + ReACT	45.4
GPT-3.5 + ReACT+CRAG	**51.0**

Table 6: Exact Match scores on FEVER.

In fact verification, the CRAG-enhanced agent gains **+5.6 EM points**, confirming its robustness in factual tasks.

6.4 Summary of Improvements

These results confirm that CRAG-based consistency filtering enhances reasoning reliability and factual grounding without requiring additional model training.

7 Analysis

The results in Section 7 demonstrate that ReACT+CRAG consistently outperforms the baseline ReACT agent. In this section, we analyze the factors contributing to these improvements, identify common failure cases, and discuss trade-offs introduced by our method.

7.1 Consistency Gains from CRAG

The key advantage of CRAG lies in its ability to *filter out logically inconsistent trajectories*. Standard ReACT prompting can produce completions that:

Dataset	Baseline (ReACT)	ReACT+CRAG	Gain
HotpotQA	19.2%	25.0%	+5.8
FEVER	45.4%	51.0%	+5.6

Table 7: Summary of accuracy improvements using ReACT+CRAG.

- Contradict earlier reasoning steps,
- Skip necessary retrieval actions,
- Produce hallucinated or unsupported final answers.

By sampling multiple trajectories and selecting the most consistent one, CRAG helps:

- Reinforce coherent reasoning chains,
- Promote factual grounding via consistent use of retrieved observations,
- Eliminate distractive or looping behavior common in single-shot completions.

We observed that CRAG-selected outputs tend to follow a cleaner thought-action-observation loop and rely more accurately on retrieved evidence.

7.2 Observed Trade-Offs

While CRAG improves reasoning quality, it introduces two important trade-offs:

1. **Increased computational cost**: Sampling and scoring multiple trajectories increases latency and API usage by a factor of N (in our case, $N = 5$).
2. **Heuristic limitations**: The ranking function is rule-based and may fail to detect nuanced semantic inconsistencies or factual errors.

These trade-offs suggest that future improvements could come from integrating *learned consistency models*, or by reducing the trajectory space through confidence-based pruning.

8 Conclusion and Future Work

In this work, we presented a hybrid prompting approach that enhances ReACT agents with a consistency-aware trajectory selection mechanism inspired by CRAG. While ReACT enables large language models to reason and act in a structured manner, it remains prone to hallucination and reasoning errors. Our method, ReACT+CRAG, improves reasoning robustness by sampling multiple reasoning-action trajectories and selecting the most logically consistent one.

We evaluated our approach on two widely used benchmarks, HotpotQA and FEVER, using GPT-3.5 as the base language model. Our results show consistent performance gains of 5–6 EM points without any fine-tuning or architectural changes. These improvements validate our core hypothesis: that prompt-level interventions, coupled with post-generation filtering, can significantly enhance multi-step reasoning quality.

8.1 Summary of Contributions

- We identified key limitations in baseline ReACT prompting and motivated the need for a consistency-aware filtering mechanism.
- We introduced ReACT+CRAG, a method that ranks multiple trajectories for internal coherence and selects the best candidate.
- We demonstrated improved exact match performance on two reasoning-intensive datasets.
- We provided an open-source implementation and detailed experimental analysis for reproducibility.

8.2 Limitations

Despite its effectiveness, our approach has limitations:

- Increased computational cost due to sampling and scoring multiple trajectories.
- Dependence on heuristic scoring functions that may not generalize across tasks or domains.
- Sensitivity to tool output quality (e.g., limitations of the Wikipedia API).

8.3 Future Directions

This work opens several promising directions for future research:

- Learning trainable consistency scoring functions to replace heuristic ranking.
- Expanding the action space to include external APIs (e.g., web search, calculator, code execution).
- Integrating confidence scoring to exit reasoning loops early and prevent overthinking.
- Applying ReACT+CRAG to other domains such as code reasoning, multi-turn dialogue, or procedural planning.

Our results support the growing view that agent-based reasoning and prompt-centric architectures, when carefully structured, can achieve state-of-the-art performance without requiring new model training.

References

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938, 2020.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *arXiv preprint arXiv:2007.01282*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kulkarni, Mike Jones, Adam Fisch, Sebastian Riedel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: A large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 809–819. Association for Computational Linguistics, 2018.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Self-consistency improves chain-of-thought reasoning in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL `https://arxiv.org/abs/2203.11171`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024. URL `https://arxiv.org/abs/2401.15884`.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReACT: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL `https://arxiv.org/abs/2210.03629`.

A Appendix / Supplemental Material

A.1 Conceptual Figures Based on Prior Work

The following figures are synthesized summaries inspired by the original ReACT and CRAG papers. They are included to give readers additional context on related methods and ideas. These are not results from our own experiments.

A.1.1 Fine-tuning ReACT Trajectories

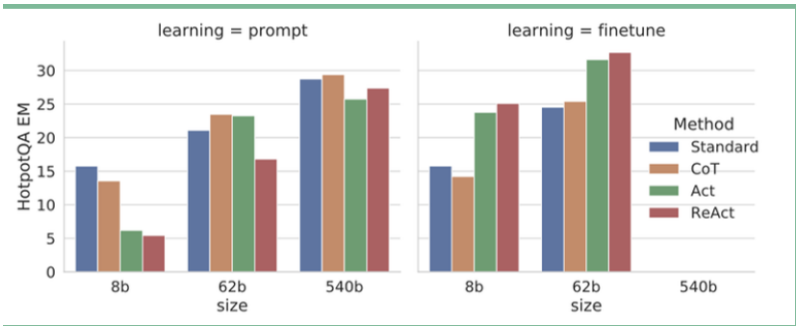

Figure 3: Illustrative summary of fine-tuning behavior of ReACT and CoT prompting as discussed in Yao et al. [2023].

A.1.2 Combining internal and external knowledge

	Type	Definition	ReAct	CoT
Success	True positive	Correct reasoning trace and facts	94%	86%
	False positive	Hallucinated reasoning trace or facts	6%	14%
Failure	Reasoning error	Wrong reasoning trace (including failing to recover from repetitive steps)	47%	16%
	Search result error	Search return empty or does not contain useful information	23%	-
	Hallucination	Hallucinated reasoning trace or facts	0%	56%
	Label ambiguity	Right prediction but did not match the label precisely	29%	28%

Figure 4: Conceptual comparison of ReACT and CoT behavior in multi-hop QA tasks. Based on descriptions in Yao et al. [2023].

A.1.3 CoT with Self-Consistency (CoT-SC)

A.1.4 About the topic

Prompt Method[a]	HotpotQA (EM)	Fever (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT-SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	**64.6**
ReAct → CoT-SC	**35.1**	62.0
Supervised SoTA[b]	67.5	89.5

Table 1: PaLM-540B prompting results on HotpotQA and Fever.

[a] HotpotQA EM is 27.1, 28.9, 33.8 for Standard, CoT, CoT-SC in Wang et al. (2022b).
[b] (Zhu et al., 2021; Lewis et al., 2020)

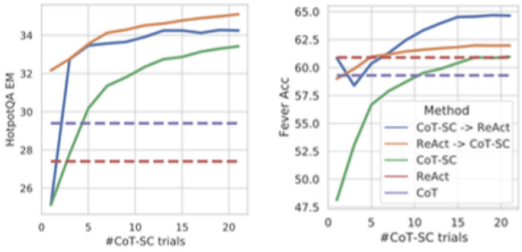

Figure 2: PaLM-540B prompting results with respect to number of CoT-SC samples used.

CoT-SC (Chain-of-Thought with Self-Consistency)

Figure 5: Prompting results on HotpotQA and FEVER using PaLM-540B, showing improvements from ReACT, CoT, and CoT-SC prompting strategies. Reproduced from Yao et al. [2023].

Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
Act (best of 6)	88	42	74	67	72	**41**	45
ReAct (avg)	65	39	83	76	55	24	57
ReAct (best of 6)	**92**	58	**96**	86	**78**	**41**	**71**
ReAct-IM (avg)	55	59	60	55	23	24	48
ReAct-IM (best of 6)	62	**68**	87	57	39	33	53
BUTLER$_g$ (best of 8)	33	26	70	76	17	12	22
BUTLER (best of 8)	46	39	74	**100**	22	24	37

Table 3: AlfWorld task-specific success rates (%). BUTLER and BUTLER$_g$ results are from Table 4 of Shridhar et al. (2020b). All methods use greedy decoding, except that BUTLER uses beam search.

Method	Score	SR
Act	62.3	30.1
ReAct	**66.6**	**40.0**
IL	59.9	29.1
IL+RL	62.4	28.7
Human Expert	82.1	59.6

Table 4: Score and success rate (SR) on Webshop. IL/IL+RL taken from Yao et al. (2022).

Figure 6: Task success rates and scores for ReACT and baseline agents on AlfWorld and WebShop benchmarks. Reproduced from Yao et al. [2023].

NeurIPS Paper Checklist

1. **Claims** We claim that ReACT+CRAG improves reasoning consistency and exact match accuracy in multi-hop QA and fact verification tasks without modifying model weights or retrievers. Our experiments support this.
2. **Limitations** Our method increases inference cost due to trajectory sampling. The CRAG ranking function is heuristic and not trained. Performance depends on tool outputs (e.g., Wikipedia API).
3. **Theory Assumptions and Proofs** No formal theory or proofs are provided; this is an empirical study.
4. **Open access to data and code** We use publicly available datasets (HotpotQA and FEVER). We will release code and processed data on GitHub.
5. **Experimental Setting/Details** We describe model choice, prompting structure, trajectory generation, evaluation metric, and dataset statistics in Sections 3–6.
6. **Experiment Statistical Significance** We report results over 500 randomly sampled examples per dataset. No confidence intervals or significance tests are reported.
7. **Experiments Compute Resources** Experiments were conducted using GPT-3.5 via OpenAI API. Each experiment batch (500 samples, 5 trajectories) required approximately 3 hours of wall time.

8. **Code Of Ethics** We adhere to responsible LLM use. Outputs are filtered to avoid offensive completions. No personally identifiable data is used.
9. **Broader Impacts** Improved LLM reasoning may benefit education, research, and factual QA. However, more reliable generations could also amplify misinformation if used improperly.
10. **Safeguards** We rely on reproducible, transparent prompting and do not deploy our method in sensitive or autonomous systems. Code and evaluation will be made public.