Focused ReAct: Improving ReAct through Reiterate and Early Stop

Shuoqiu Li
Carnegie Mellon University
shuoqiul@alumni.cmu.edu

Han Xu
University of Illinois at Urbana-Champaign
hanxu8@illinois.edu

Haipeng Chen
William & Mary
hchen23@wm.edu

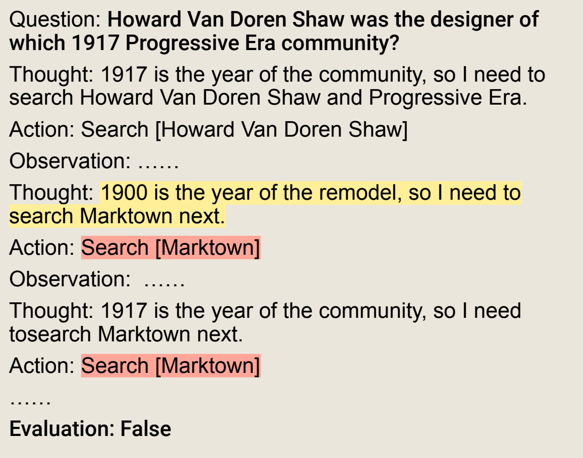
Abstract

Large language models (LLMs) have significantly improved their reasoning and decision-making capabilities, as seen in methods like ReAct. However, despite its effectiveness in tackling complex tasks, ReAct faces two main challenges: losing focus on the original question and becoming stuck in action loops. To address these issues, we introduce **Focused ReAct**, an enhanced version of the ReAct paradigm that incorporates reiteration and early stop mechanisms. These improvements help the model stay focused on the original query and avoid repetitive behaviors. Experimental results show accuracy gains of 18% to 530% and a runtime reduction of up to 34% compared to the original ReAct method.

1 Introduction

Recent advancements in large language models (LLMs) have enabled more sophisticated techniques for reasoning and decision-making. One such technique, the ReAct framework (Reason+Act), has gained popularity for its dual approach of alternating between reasoning and action (Yao et al., 2023). This combination allows ReAct to excel in handling complex tasks by better adapting to dynamic environments (Wang et al., 2024).

Despite its strengths in general question-answering (QA), ReAct sometimes falls short in delivering accurate results, as demonstrated in Figure 1. When confronted with lengthy or intricate questions, the model—paired with the ReAct framework—may lose track of the original question, leading to incorrect responses. Additionally, the ReAct method can get trapped in repetitive action loops, continuously performing the same action without progressing toward a solution. To address these challenges, we introduce **Focused ReAct**, an enhanced version of the original ReAct method specifically designed for QA tasks.



Question: **Howard Van Doren Shaw was the designer of which 1917 Progressive Era community?**

Thought: 1917 is the year of the community, so I need to search Howard Van Doren Shaw and Progressive Era.

Action: Search [Howard Van Doren Shaw]

Observation: ……

Thought: 1900 is the year of the remodel, so I need to search Marktown next.

Action: Search [Marktown]

Observation: ……

Thought: 1917 is the year of the community, so I need tosearch Marktown next.

Action: Search [Marktown]

……

Evaluation: False

Figure 1: The yellow highlighted region illustrates where the model loses focus on the original question, while the red highlighted region depicts repeated actions that result in a failed response.

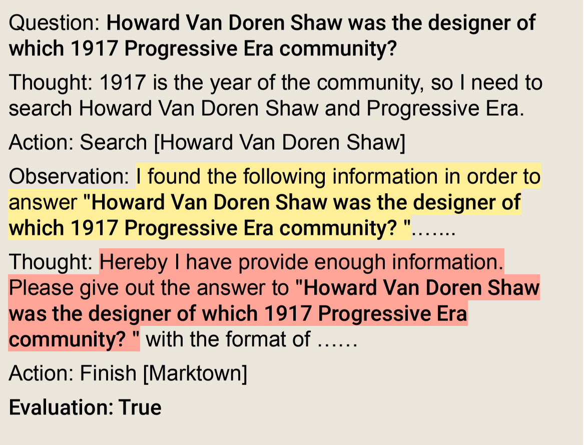
2 Methodology

This section outlines the two core features of the **Focused ReAct** method: **reiterate** and **early stop**.

2.1 Reiterate to Solve Context Loss

As indicated in the introduction, The first challenge stems from the extended reasoning process, where the relevance of the original question diminishes as additional reasoning and actions are appended. To resolve this, the concept of **reiterate** is introduced. In this approach, the original question is restated at the beginning of each reasoning step in the ReAct cycle. This technique is illustrated in the green-highlighted region of Figure 2.

By reiterating the original question at each step, the model continually emphasizes the user's query, preventing it from being overshadowed by the increasingly long context that ReAct tends to create. This simple yet effective strategy mitigates the context dilution problem illustrated in Figure 1, ensuring that the output remains aligned with the user's request, even in complex or multi-step tasks.



Question: **Howard Van Doren Shaw was the designer of which 1917 Progressive Era community?**

Thought: 1917 is the year of the community, so I need to search Howard Van Doren Shaw and Progressive Era.

Action: Search [Howard Van Doren Shaw]

Observation: I found the following information in order to answer **"Howard Van Doren Shaw was the designer of which 1917 Progressive Era community? "**.......

Thought: Hereby I have provide enough information. Please give out the answer to **"Howard Van Doren Shaw was the designer of which 1917 Progressive Era community? "** with the format of ……

Action: Finish [Marktown]

Evaluation: True

Figure 2: The QA process by Focused ReAct for the same question, which applies reiteration (highlighted in yellow) and early stop (highlighted in red) to resolve the context loss and the repeated action issue.

2.2 Early Stop to Prevent Action Repetition

The second challenge, as outlined in the introduction, occurs when the model gets caught in repetitive loops, generating the same response without progressing toward the correct answer. To tackle this, we propose an **early stop** mechanism. It assumes that by the time a duplicate action occurs, sufficient information has been gathered.

When the program detects repeated actions, it triggers a termination request - highlighted in red in Figure 2 - instructing the model to generate a final answer based on the existing information. This approach prevents unnecessary repetition and helps the QA process arrive at an accurate response more efficiently.

3 Experimentation

We evaluate Focused ReAct against the ReAct baseline using the Gemma 2 2B (Team et al., 2024), Phi-3.5-mini 3.8B (Abdin et al., 2024) and Llama 3.1 8B (Dubey et al., 2024) models. The implementation uses the PyTorch and Transformers libraries[1], with experiments conducted on a single NVIDIA L4 GPU with 24GB of memory. The dataset consists of 150 QA tasks, randomly selected from HotPotQA (Yang et al., 2018). We measure accuracy as the ratio of correctly answered tasks to the total number of tasks, while runtime is recorded for the completion of each task.

Table 1: Accuracy Comparison of ReAct vs. Focused ReAct

Model	ReAct	Focused ReAct	abs./rel. diff
Gemma 2 2B	2.0%	**12.6%**	**+10.6 / 530%**
Phi-3.5-mini 3.8B	22.0%	**26.0%**	+4.0 / 18%
Llama 3.1 8B	14.0%	**23.3%**	+9.3 / 66%

Table 1 presents the accuracy comparison between the vanilla ReAct and Focused ReAct across the Gemma 2, Phi-3.5, and Llama 3.1 models. Focused ReAct demonstrates an 18%-530% improvement in accuracy.

Table 2: Runtime Comparison (Average and Std) for ReAct vs. Focused ReAct

Model	ReAct	Focused ReAct	abs./rel. diff
Gemma 2 2B	11.68±2.66s	**7.68±2.41s**	**-4.0 / 34%**
Phi-3.5 -mini 3.8B	23.23±8.42s	**22.50±11.19**s	-0.73 / 3%
Llama 3.1 8B	24.10±23.48s	**23.12±25.35**s	-0.98 / 4%

Table 2 summarizes the average runtime and standard deviation (std) for both the original ReAct and Focused ReAct methods. Models with fewer parameters show a 34% reduction in runtime, while models with larger parameter sizes exhibit no significant decrease. This discrepancy may be attributed to the fact that smaller models, with weaker reasoning capabilities, benefit more from Focused ReAct optimizations. In contrast, larger models are more robust at maintaining context and performing deeper reasoning, which may reduce the relative impact of Focused ReAct's efficiency gains. As a result, the runtime benefits are less pronounced compared to smaller models.

4 Conclusion

This paper identifies two common issues with the ReAct method in QA: losing focus on the original question during extended reasoning and becoming stuck in repetitive action loops. To overcome these problems, we propose **Focused ReAct**, which incorporates **reiteration** and **early stop** to improve upon the ReAct framework. Compared to the original ReAct method, the new approach achieves accuracy improvements between 18% and 530%, along with a reduction in runtime of up to 34%.

For future work, we plan to extend Focused ReAct to a broader range of tasks and scenarios, evaluate its generalizability and robustness, and explore techniques to further accelerate its performance (Xu et al., 2024).

[1] Our code implementation and experiments are available at https://github.com/wmd3i/Focused-ReAct.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

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*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Han Xu, Jingyang Ye, Yutong Li, and Haipeng Chen. 2024. Can speculative sampling accelerate react without compromising reasoning quality? In *The Second Tiny Papers Track at ICLR 2024*.

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*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.