CARF: Causal Reasoning Augmented Reflection for Long-Term Planning via LLMs

Tuocheng Hu[1], Jiatao Zhang[1], Qingmiao Liang[2], Zeng Gu[2], Yufan Song[1],
Wei Song[13](✉), and Shiqiang Zhu[1](✉)
[1]Zhejiang University, Hangzhou, China
Email: {tuochenghu, jiataozh, 22360553, weisong-rob, sqzhu}@zju.edu.cn
[2]University of Chinese Academy of Sciences, Hangzhou, China
Email: {lqmi0523@163.com, josephgu0323@outlook.com}
[3]Yuhang Humanoid Robot Industry Innovation Center, Hangzhou, China

***Abstract*—Recent developments suggest that Large Language Models (LLMs) can identify and correct errors in their generated responses using reflection mechanisms. However, when applied to long-term task planning, these methods reveal significant limitations. Reflection methods may neglect the causes of errors in earlier planning, producing results that contain inaccurate information, thereby leading to further mistakes in subsequent planning. This paper explores the reflection frameworks suitable for long-term task planning. Inspired by human causal cognitive processes, we introduce the Causal Reasoning Augmented Reflection Framework (CARF). CARF employs systematic causal reasoning to accurately identify the root causes of errors and to generate effective action plan revisions by integrating association information. We conducted experiments with household tasks in Alfworld, and the results show that our framework substantially increases the success rate in complex long-term tasks. Website at https://hutcber.github.io/CARF.github.io/**

***Index Terms*—Robotics, Task Planning, Reflection, Causal Reasoning, Large Language Model**

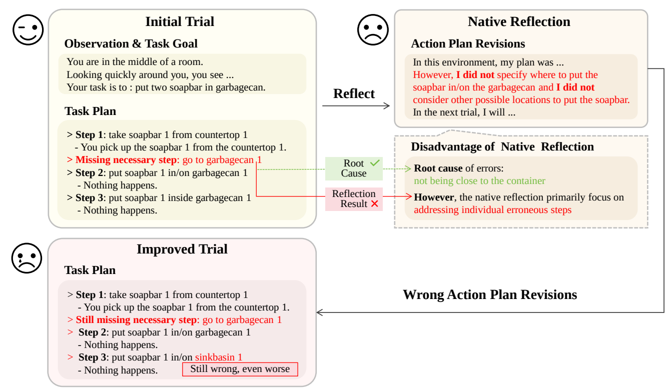

Figure 1. Illustrates the limitation in using reflection methods for long-term task planning. These methods merely identify and correct one or a few erroneous steps, overlooking the root causes of errors. However, correcting only these steps may lead to greater errors in subsequent trials.

I. Introduction

Task planning is a critical decision-making process, widely applied in diverse fields including robotics navigation [1], manipulation [2], [3], everyday household tasks [4]–[6], and industrial domains such as manufacturing and assembly [7], [8]. While research on short-term decision-making has progressed significantly, the complexity and error accumulation inherent in long-term task planning continue to present substantial challenges. Long-term task planning requires the management of extended action sequences and more complex environmental states. Additionally, in long-term planning, the issue of error accumulation becomes more evident, as initial mistakes may cause progressively greater deviations over time.

Recently, Large Language Models (LLMs) have been integrated into task planning due to their superior common sense reasoning and logical capabilities, as seen in methods like ReAct [9] and SayCan [4]. However, these LLM-based methods can face challenges like hallucinations and contextually unfaithful responses [10]–[12]. To address these issues, recent studies have introduced reflection mechanisms into task planning, employing either human or automatic feedback to refine erroneous outputs. This transition from an open-loop to a closed-loop system is exemplified by methods like RETROFORMER [13] and Reflexion [14]. Therefore, we pose the question: Can the LLM-based reflection framework be applied to long-term task planning?

Although the above methods have achieved some initial success, a key challenge in applying reflection methods to long-term task planning lies in their inability to identify the root causes of errors (RCE), which is essential for the subsequent generation of action plan revisions. Presently, these reflection methods do not concentrate on understanding the reasons behind errors in long-term task planning. Instead, they focus solely on revising the action plan by addressing only part of the erroneous steps [13]–[15]. For example, in Figure 1, reflection methods focus on testing different placement strategies for a soapbar instead of investigating why the failure occurred in the first place. This neglect of identifying the RCE leads to subsequent trials that are even more erroneous. In some cases, it can even result in dangerous actions that compromise the robot's reliability.

Motivated by the above challenges, this paper investigates

This work was supported by Key R&D Program of Zhejiang (No. 2025C01091), and by National Natural Science Foundation of China (No. U21A20488).

the analysis of RCE in complex long-term task planning. Such planning typically involves extended action sequences and accumulates errors over time. Therefore, simply identifying and correcting one or a few erroneous steps does not fully rectify the errors in the planning. Only by accurately identifying the RCE can one generate action plan revisions that address the root causes, resolving the accumulating errors and successfully completing the task in subsequent trials. However, current reflection methods predominantly focus on addressing individual erroneous steps to generate action plan revisions. In contrast, human intelligence is characterized by the ability to perform causal reasoning across three levels: Association, Intervention, and Counterfactuals [16]–[18]. These levels range from simple observations to complex imaginations, enabling humans to systematically analyze the RCE and propose effective action plan revisions, thus enhancing the success rate of subsequent trials.

Inspired by the process outlined above, we propose CARF, a general reflection framework that harnesses the causal reasoning capabilities of LLMs to analyze the RCE and generate effective action plan revisions for future trials. First, we employ LLMs to infer associated information from environmental observations, corresponding to the first level of causal reasoning. Next, based on the task trajectory, associated information, and the RCE in the previous action plan, we formulate action plan revisions aimed at addressing the root causes and successfully complete the next trial, corresponding to the second level of causal reasoning. Finally, for the failed action plan revisions, we utilize counterfactual reasoning to substitute critical erroneous steps from previous trials and identify possible the RCE, corresponding to the third level of causal reasoning. To further enhance the reflection optimization process, we integrate memory management to improve the LLM's performance in subsequent trials.

To evaluate the effectiveness of the proposed framework, we conducted experiments in Alfworld and categorized the dataset into three types of tasks. The results show that the our proposed framework significantly outperforms the baseline across various tasks, achieving a 12% increase in identification of RCE across all tasks. In summary, the contributions of this paper are as follows:

1) To our knowledge, we are the first to study the reflection frameworks in long-term task planning.
2) We have introduced a framework named CARF. Our framework effectively enhances the application of reflection frameworks in long-term task planning by identifying the RCE, obtaining association information, and then generating action plan revisions.
3) We conducted experiments in the virtual household environment Alfworld. The results demonstrate that our framework could effectively enhances the success rate of complex long-term task planning.

II. Related works

A. LLMs for Task Planning

In recent years, the general performance exhibited by LLMs has sparked interest among researchers in exploring the potential of LLMs for task planning in open-ended environments. For instance, LID [19] uses GPT-2 as a policy network, predicting the next action based on encoded tasks information. ReAct [9] introduces a new prompting paradigm that combines reasoning and action to enhance the ability of LLMs to solve general tasks. CLIN [20] features persistent dynamic text memory centered on causal abstraction, enabling LLM-based agents to continually learn and adapt in new environments. RAP [21] combines a retrieval-based mechanism with a generative model, boosting multimodal LLM agents' planning capabilities. SYNAPSE [22] employs a trajectory-based prompting mechanism that selects and uses similar past trajectories from memory.

B. LLMs for Reflection

To address the challenges of hallucinations or context unfaithfulness in outputs from LLMs, researchers have investigated the application of reflection mechanisms to improve the quality of outputs. For example, SELF-REFINE [23] is an iterative self-improvement framework that has shown significant improvements in single-step reasoning tasks by providing feedback on the current output and refining it based on the feedback. Reflexion [14] introduces a framework that uses language feedback as a semantic gradient signal. CRITIC [24] interacts with specific tools to assess and correct errors and flaws in the generated text. SelfCheck [25] corrects mistakes by regenerating specific steps in the reasoning process and comparing them with the original steps.

C. LLMs for Causal Reasoning

One potential solution to the aforementioned problem is to leverage causal reasoning to identify the RCE. In complex long-term tasks, errors accumulate over time, and causal reasoning can pinpoint the RCE, enabling the development of effective action plan revisions. Recent research has shown that LLMs have potential in causal reasoning, as they can outperform existing causal reasoning algorithms by utilizing common sense and domain knowledge in tasks such as counterfactual reasoning and real-world causal relationships [16], [26]. Furthermore, even in the absence of specific domain knowledge, LLMs can still perform limited causal reasoning by leveraging the available data [27].

For complex long-term tasks, if reflection frameworks fail to identify RCE, it can lead to ineffective or incorrect plan revisions, which in turn may cause future trials to fail again. Applying reflection frameworks to long-term task planning remains a challenging and unresolved issue.

III. Preliminaries

A. Planning Framework

A task can be defined as a tuple $\langle S, O, A, \mathrm{T} \rangle$, where S represents the set of states, O is the set of observations,

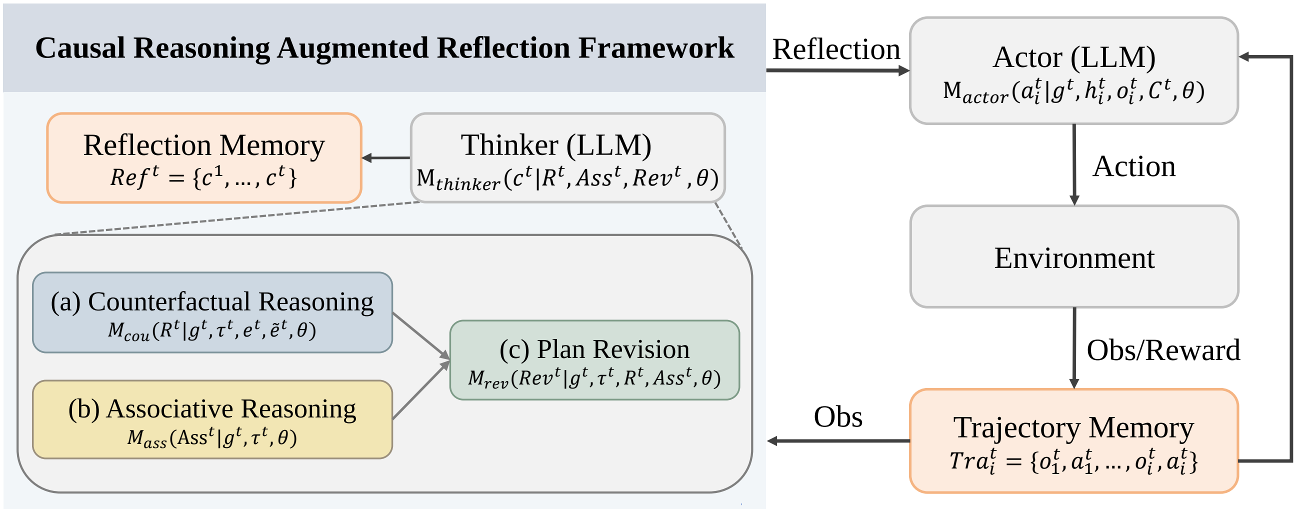

Figure 2. The framework of CARF. Our framework operates by taking historical task information and environmental feedback as input and producing reflection results as output. The framework comprises three stages: 1. Counterfactual Reasoning is responsible for identifying and correcting critical erroneous steps and analyzing the RCE. 2. Associative Reasoning is tasked with inferring about the associations. 3. Plan Revision is charged with generating the action plan revisions for subsequent trials. Additionally, we have introduced a Memory Management module to manage the historical results generated.

A denotes the set of actions, and T describes the stochastic transitions [28]. The transition function, formalized as $\mathrm{T} : S \times A \rightarrow S$, represents the environmental state changes resulting from actions taken. The objective is to determine a sequence of actions that transitions from the initial state to the target state, thereby forming a plan. Although there is no strict definition for long-term tasks, they generally involve 5 to 15 or more steps and necessitate extensive interactions with objects and the environment, in contrast to short-term tasks.

In the reflection method, the sequence of actions is generated by the strategy function. It can be defined as $\Phi(a_i^t | g^t, h_i^t, o_i^t, C^t)$, where determines the next action $a_i^t \in A$, based on the task goal g^t, the historical information $h_i^t = \{o_1^t, a_1^t, \ldots, o_{i-1}^t, a_{i-1}^t\}$, the observation $o_i^t \in O$, and the historical reflection information set C^t. Here, t is the index of the trial, and i is the index of the step within the t-th trial.

B. Error Accumulation Problem

In many sequential decision-making and learning systems, **error accumulation** critically impacts performance, especially in long-term task planning involving complex and dynamic home environments [29]. While recent methods have made progress in alleviating this problem, they still struggle to handle the diverse contexts, extended action sequences, and uncertainty inherent in household tasks.

In long-term task planning, the issue of accumulated errors occurs when serialized decisions are made, with early mistakes continuously affecting subsequent decisions, leading to a gradual magnification of errors throughout the execution of the task. Accumulated errors in the planning process can result in multiple erroneous steps, making it difficult for reflective frameworks to pinpoint the root causes and devise plan corrections to optimize subsequent attempts. When these plan corrections continue to deviate, it leads to further deviations from the correct trajectory in subsequent attempts, and in some cases, even to the execution of hazardous actions.

C. Causal Reasoning Framework

The concept of the *Ladder of Causation*, proposed by Judea Pearl, provides a theoretical framework for reasoning about cause and effect across three levels [17]. The first level, **Association**, captures correlations between variables based on observations, such as recognizing the frequent co-occurrence of dirty plates and their presence in the sink. The second level, **Intervention**, reasons about the consequences of deliberate actions, for example, transferring a plate to the dishwasher to achieve a cleaner state. The third and highest level, **Counterfactuals**, addresses "what if" questions by reasoning about alternate realities, such as inferring that the plate would remain dirty if it had not been moved.

In the context of long-term task planning in domestic environments, this hierarchy is particularly relevant for addressing **error accumulation**. By progressing from mere pattern recognition to intervention and counterfactual reasoning, a system can more accurately diagnose the root causes of task failures and anticipate the effects of alternative decisions, thereby mitigating cascading errors in sequential planning.

IV. Methodology

In this section, we will provide a detailed introduction to our proposed framework, CARF, and its interactions with external systems, as shown in Figure 2. The external system primarily includes the strategy model and the environment. We define the strategy model as the Actor, which uses task goal, historical information, environmental observations, and reflection results as inputs to generate actions for interacting with the environment. Our reflection framework is defined as the Thinker, which integrates task information and environmental feedback to conduct causal reasoning. After a trial, the Thinker initially utilizes counterfactual reasoning to identify potential RCE. Subsequently, it applies associative reasoning to infer association information from environmental observations. Finally, the Thinker combines the RCE with the association information to generate action plan revisions. Additionally, we have introduced a Memory Management module to manage the historical results generated.

A. LLMs as Actor

Following [14], we use the React method [9] as the Actor module $Actor$, due to its outstanding performance in task planning. As an implementation of the strategy function, M_{Actor} takes information about the task and environment as inputs and outputs the next action. It can be represented as:

$$a_i^t = M_{actor}(g^t, h_i^t, o_i^t, C^t, \theta), \tag{1}$$

where a_i^t denotes the next action output, g represents the task goal, h_i^t encapsulates the historical information, o_i^t indicates the environmental observations, $C^t = \{c^{t-k}, \ldots, c^{t-1}\}$ represents the set of reflection outcomes from the past k trials, and θ represents the parameters of LLMs. Additionally, the action and observation at each step will be collected by the memory management module. The Actor module possesses strong natural language understanding and reasoning capabilities, enabling it to adapt flexibly to complex tasks and dynamically optimize execution.

B. LLMs as Thinker

To identify the RCE in previous trials and to generate action plan revisions, we designed the Thinker module, $M_{thinker}$. As illustrated in Figure 3, The Thinker consists of three submodules: The Counterfactual Reasoning submodule, responsible for identifying and correcting critical erroneous steps and analyzing the RCE R^t; the Associative Reasoning submodule, tasked with inferring the relationships between objects and containers Ass^t; and the Plan Revision submodule, which generates corrections for the action plan for subsequent trials Rev^t. It can be formalized as follows:

$$c^t = M_{thinker}(R^t, ASS^t, Rev^t), \tag{2}$$

where c^t represents the collection of outputs from each submodule. Furthermore, the results from the Thinker module will be collected by the memory management module.

Counterfactual Reasoning M_{cou} analyzes the RCE by identifying and correcting critical erroneous steps in the original trajectory and evaluating potential outcomes that were not actually executed. During the trial t, M_{cou} takes the task information, and execution results as inputs. It then reasons under the assumption of replacing the actual execution results with counterfactual outcomes, thereby outputting the RCE. This process can be formally expressed as:

$$R^t = M_{cou}(g^t, \tau^t, e^t, \tilde{e}^t, \theta), \tag{3}$$

where R^t represents the RCE for natural language expression, τ^t represents the trajectory, while e^t and $\tilde{e}^t$ respectively represent the actual execution results and the counterfactual execution results. This submodule, through counterfactual reasoning, can efficiently identify the RCE.

Associative Reasoning M_{ass} infers associations between objects and containers based on observations within the trajectory. During the trial t, M_{ass} takes the task goal and trajectory as inputs and outputs the associations between objects and containers. To minimize redundancy and shorten the context length, we focus only on objects directly related to the task goal. It can be expressed as:

$$Ass^t = M_{ass}(g^t, \tau^t, \theta), \tag{4}$$

where Ass^t represents the association information between objects and containers as specified in the task goal. This submodule, through associative reasoning, can precisely identify the locations of objects, thereby improving search efficiency.

Plan Revision M_{rev} proposes action plan revisions based on the results of counterfactual and associative reasoning. During the trial t, M_{rev} takes the identified RCE, the association information between objects and containers, and the task information as inputs, and outputs the action plan revisions. The process is outlined as follows:

$$Rev^t = M_{rev}(g^t, \tau^t, R^t, Ass^t, \theta), \tag{5}$$

where Rev^t represents the action plan revision for natural language expression. This submodule, by integrating the analysis results, generates optimized action plan revisions, significantly improving the success rate of tasks.

The Thinker consists of three submodules, each corresponding to a level of causal reasoning. This structure can efficiently identify RCE, infers association information, and then generates effective action plan revisions, thereby increasing the success rate of long-term tasks.

C. Memory Management

Previous studies frequently incorporated all essential information into the prompt [30]. However, for long-term tasks, this method could result in overly long prompts, taxing the memory capacities of LLMs. To address this challenge, we designed a Memory Management module mem to manage the trajectories and the analysis results from the Thinker module during the planning process. The trajectory management component is denoted as Tra, which manages trajectory information to generate subsequent actions. It is defined as $Tra_i^t = \{o_1^t, a_1^t, \ldots, o_i^t, a_i^t\}$, where Tra_i^t represents the set of actions and observations up to step i of the trial t. The reflection management component is represented as Ref, which manages the reflection results from the Thinker to optimize the next trial. It is defined as $Ref^t = \{c^1, \ldots, c^t\}$, where Ref^t represents the accumulated set of reflection results up to the trial t. The introduction of the Memory Management module alleviates the pressure on LLMs from handling overly

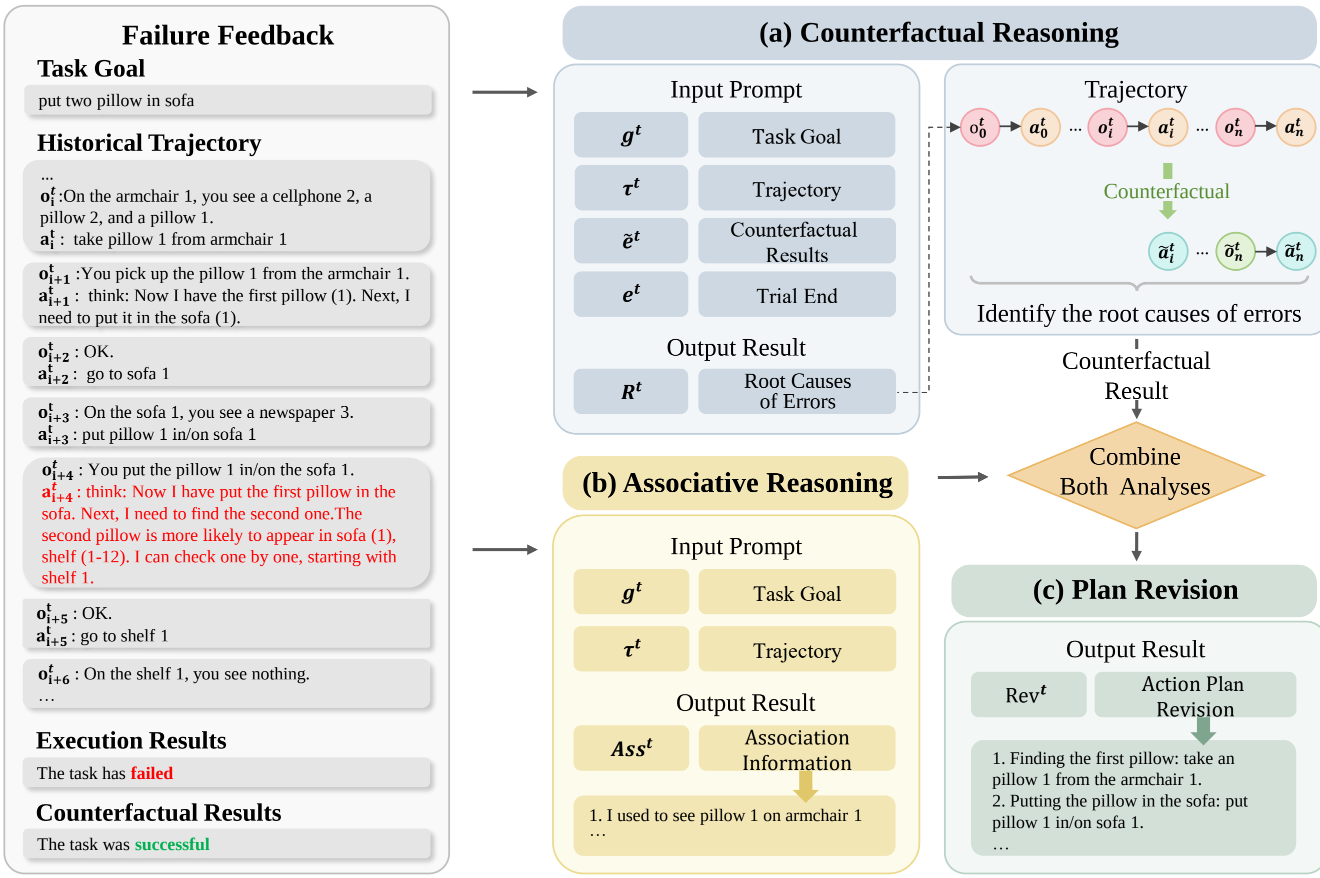

Figure 3. The detailed illustration of CARF. This framework integrates task information and environmental feedback as input and outputs the RCE in the planning process, the association information, and the action plan revisions

lengthy prompts, thereby significantly improving the planning process.

V. EXPERIMENTS

In this section, our framework is evaluated through experiments conducted in a virtual household environment.

A. Experimental Setup

Environment.we conducted five trials to assess different methods in Alfworld, a semantically-based virtual environment used for training and assessing intelligence. Alfworld consists of 120 rooms, each dynamically populated with a set of portable objects and static containers. Intelligent agents can interact with these objects and containers to perform everyday tasks such as cooling, cleaning, and heating. The diversity of task environments and goals provides a comprehensive test for intelligent agents.

Dataset. We conducted experimental tests on the unseen dataset provided by [31]. The unseen dataset includes six tasks: Pick & Place, Examine in Light, Clean & Place, Heat & Place, Cool & Place, and Pick Two & Place. We performed experiments on 100 tasks to obtain results. To minimize the impact of the number of tasks, we categorized them based on task complexity and type into three categories: Pick Task, Pick & Act Task, and Examine Task, with each category averaging around ten steps.

Compare methods. We compared our method with three previous approaches: 1. **Plan-Only**: ProgPrompt [30], which directly generates the next action by inputting a simple context containing the task description into the LLM; 2. **React-Only**: React [9], which incorporates a reasoning process during planning to derive the next action based on reasoning; 3. **React-Reflexion**: Reflection [14], which integrates a reflection process during multiple trials in React, obtaining the next action based on the reflection from the previous trial.

Metrics. We evaluated the performance from three aspects, with the first two metrics specifically targeting the reflection method: **Identification** assesses whether the reflection method can identify the RCE by counting whether each task's reflection identifies the RCE; **Revision** evaluates whether the reflection method revises errors from previous plans by counting whether each task's reflection has revised these errors; **Success Rate** measures the effectiveness of the reflection and planning methods by counting whether each task is successfully completed [19], [32].

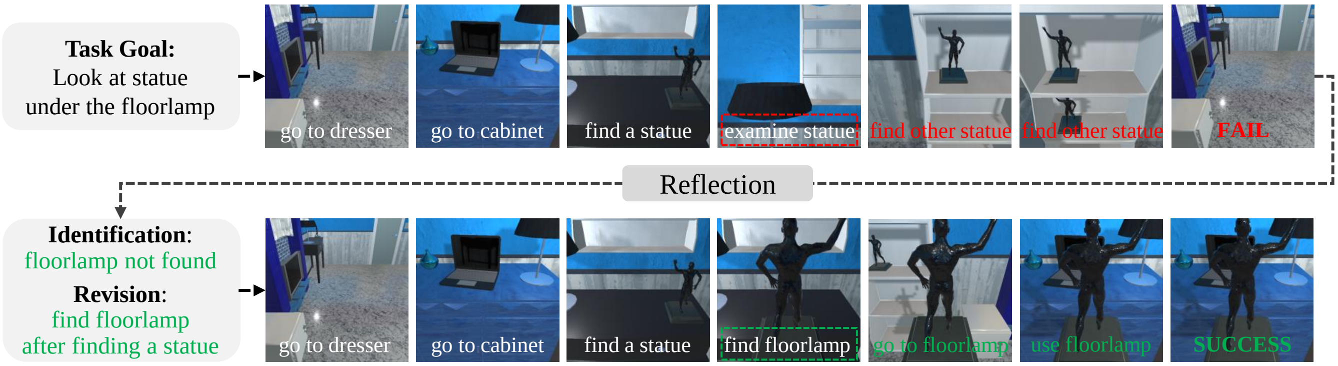

Figure 4. Example of a failed task successfully replanned using our framework.

Table I
OVERALL PERFORMANCE CARF AND BASELINES ACROSS VARIOUS TASKS. OUR FRAMEWORK OUTPERFORMS ALL BASELINES. IDENTIFICATION AND REVISION ARE SOLELY USED TO EVALUATE THE REFLECTION METHOD.

Categories	Methods	Pick Tasks			Pick & Act Tasks			Examine Tasks		
		Identification	Revision	Success Rate	Identification	Revision	Success Rate	Identification	Revision	Success Rate
w/o reflection	**Planning-Only**	-	-	56	-	-	54	-	-	18
	React-Only	-	-	81	-	-	73	-	-	82
With reflection	**React-Reflexion**	68	62	84	68	61	80	40	60	73
	React-CARF	**73**	**70**	**85**	**77**	**74**	**85**	**100**	**100**	**100**

B. Results

The primary results are presented in Table I. We can observe that: 1. our framework consistently outperforms other methods across all metrics and task categories. This indicates that our framework can efficiently identify the RCE, thereby generating effective action plan revisions and enhancing the success rate. Figure 4 exemplifies the application of CARF to long-term tasks. 2. A noticeable trend is that a low success rate corresponds to a low rate of identifying the RCE. This highlights a major issue in long-term task planning: a low identification of the root causes leads to inaccuracies in action plan revisions, severely impacting the successful execution of subsequent plans. 3. The **React-only** method and the **React-reflexion** method exhibit similar performance across various tasks, indicating that reflection methods that only revises action plans for one or a few erroneous steps are not effectively resolving errors encountered in long-term task planning. 4. The **React-CARF** method demonstrates superior performance compared to methods like **React-reflexion**, indicating that CARF can pinpoint the RCE, thereby effectively enhancing task success rate. Furthermore, the differences in identification, revision, and success rate across different task categories suggest that the effectiveness of reflection methods may be influenced by task complexity.

VI. ANALYSIS AND DISCUSSION

A. Episode Analysis

To analyze the effectiveness of methods, we conducted an episode analysis, with results displayed in Figure 5. Our observations are as follows: 1. Our framework consistently outperformed the baseline across all episodes, demonstrating its effectiveness in improving revision and success rate for long-term tasks. 2. Across all episodes, the React-only method consistently outperforms the Planning-only approach. However, the React-Reflection method does not exceed the performance of the React-only method. This further confirms our previous assertion that without identifying the RCE in planning, reflection methods that merely revises action plans for one or a few erroneous steps cannot achieve the goal of optimizing subsequent trials. These results highlight the importance of integrating causal reasoning into the reflection process, as our framework's ability to accurately identify root causes and generate effective plan revisions significantly enhances success rate in long-term task planning.

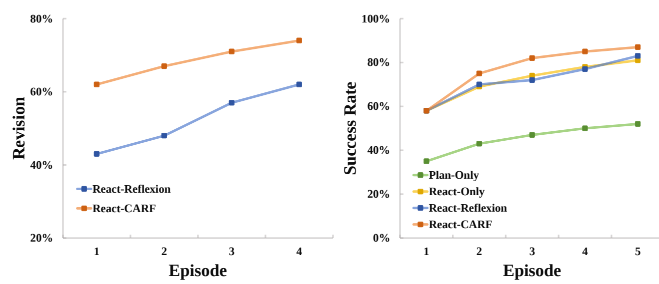

Figure 5. Result of revision and success rate with different episodes. CARF effectively improves the performance of long-term tasks.

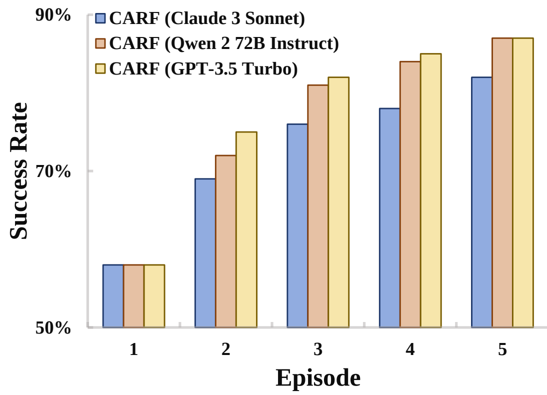

Figure 6. Result of success rate with different LLMs. CARF possesses outstanding universality and robustness.

B. Method Performance Across Different LLMs

To further investigate the universality and robustness of our framework, we conducted experiments with CARF on multiple LLMs of varying scales and architectures. The results, as shown in Figure 6, indicate the following: 1. Across all LLMs, our framework consistently outperforms the baseline, demonstrating outstanding universality. 2. Our framework displays similar performance across LLMs with different architectures, indicating remarkable robustness. Additionally, based on different LLMs, our framework exhibits varying growth rates across episodes, highlighting the distinct advantages each LLM brings to handling complex tasks. The framework effectively utilizes these model characteristics to achieve dynamic adaptability. These results underscore the wide applicability of our framework, enabling sustained optimization of performance across various LLMs and addressing the demands of complex and variable tasks.

C. Ablation Study

Table II
ABLATION OF VARIOUS SUBMODULES OF CARF. ALL DESIGN SUBMODULES CONTRIBUTE TO THE PERFORMANCE. WITHOUT THE COUNTERFACTUAL SUBMODULE, IDENTIFICATION CANNOT BE MEASURED.

	Identification	Revision	Success Rate
Ours full	76	74	87
w/o Associative	60	67	78
w/o Counterfactual	-	67	79
w/o Both Analyses	-	52	75

To comprehensively demonstrate the effectiveness of each submodule in our framework, we conducted ablation studies. The results are detailed in Table II. In the **w/o Associative** model, action plan revisions are generated without association information. Compared to our complete model, the three metrics respectively decreased by 16%, 7%, and 9%. The results indicate that association information contributes to improving identification, which in turn enhances success rate. This is linked to the key role of association information in forming effective action plan revisions, which bring trajectories closer to the correct paths and reduce the difficulty of identification. In the **w/o Counterfactual** model, action plan revisions are generated without identifying the RCE. Compared to our complete model, the w/o Counterfactual shows a 7% decrease in revision, leading to an 8% decline in success rate. This suggests that utilizing the counterfactual reasoning submodule to identify the RCE can significantly enhance the revision and thereby increase success rate. The **w/o Both Analyses** model directly generates action plan revisions. Compared to our complete model, this variant's revision decreased by 22%, and the success rate dropped by 12%. These results highlight the critical roles of the associative and counterfactual reasoning submodules in improving revision and success rate.

VII. CONCLUSIONS

To our knowledge, we are the first to study the reflection framework for long-term task planning. Inspired by human intelligence, we propose a new framework named CARF, which uses three submodules corresponding to the three levels of causal reasoning to augment reflection frameworks. Additionally, we have implemented a memory management module to efficiently manage the LLM's context. Experiments conducted in a virtual household environment demonstrate that our framework significantly improves the identification, revision, and success rate in long-term task planning, showcasing outstanding universality and robustness.

REFERENCES

[1] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. *Microsoft Autonomous Systems and Robotics Research*, 2:20, 2023.

[2] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187, 2023.

[3] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[4] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318, 2023.

[5] Jinjie Mai, Jun Chen, Bing Li, Guocheng Qian, Mohamed Elhoseiny, and Bernard Ghanem. Llm as a robotic brain: Unifying egocentric memory and control. *arXiv preprint arXiv:2304.09349*, 2023.

[6] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *arXiv preprint arXiv:2305.05658*, 2023.

[7] Lisa Heuss, Daniel Gebauer, and Gunther Reinhart. Concept for the automated adaption of abstract planning domains for specific application cases in skills-based industrial robotics. *Journal of Intelligent Manufacturing*, 35(8):4233–4258, 2024.

[8] Anil Kumar Inkulu and M. V. A. Bahubalendruni. Human-robot collaborative task planning for assembly system productivity enhancement. *Robotic Intelligence and Automation*, 44(1):120–130, 2024.

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

[10] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*, 2023.

[11] Jiatao Zhang, Lanling Tang, Yufan Song, Qiwei Meng, Haofu Qian, Jun Shao, Wei Song, Shiqiang Zhu, and Jason Gu. Fltrnn: Faithful long-horizon task planning for robotics with large language models. In *IEEE International Conference on Robotics and Automation*, pages 6680–6686, 2024.

[12] Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, 2021.

[13] Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*, 2023.

[14] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652, 2024.

[15] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, 2022.

[16] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: A benchmark to assess causal reasoning capabilities of language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[17] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[18] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.

[19] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. In *Advances in Neural Information Processing Systems*, volume 35, pages 31199–31212, 2022.

[20] Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. Clin: A continually learning language agent for rapid task adaptation and generalization. *arXiv preprint arXiv:2310.10134*, 2023.

[21] Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. Rap: Retrieval-augmented planning with contextual memory for multimodal llm agents. *arXiv preprint arXiv:2402.03610*, 2024.

[22] Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *The Twelfth International Conference on Learning Representations*, 2023.

[23] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.

[25] Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*, 2023.

[26] Hengrui Cai, Shengjie Liu, and Rui Song. Is knowledge all large language models needed for causal reasoning? *arXiv preprint arXiv:2401.00139*, 2023.

[27] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

[28] Mikko Lauri, David Hsu, and Joni Pajarinen. Partially observable markov decision processes in robotics: A survey. *IEEE Transactions on Robotics*, 39(1):21–40, 2022.

[29] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[30] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

[31] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[32] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147, 2022.