2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

FLTRNN: Faithful Long-Horizon Task Planning for Robotics with Large Language Models

Jiatao Zhang[1,2], Lanling Tang[3,2], Yufan Song[1], Qiwei Meng[2], Haofu Qian[1,2], Jun Shao[1,2], Wei Song[2,1⋆], Shiqiang Zhu[1⋆], and Jason Gu[2]

***Abstract*— Recent planning methods based on Large Language Models typically employ the In-Context Learning paradigm. Complex long-horizon planning tasks require more context(including instructions and demonstrations) to guarantee that the generated plan can be executed correctly. However, in such conditions, LLMs may overlook(unfaithful) the rules in the given context, resulting in the generated plans being invalid or even leading to dangerous actions. In this paper, we investigate the faithfulness of LLMs for complex long-horizon tasks. Inspired by human intelligence, we introduce a novel framework named FLTRNN. FLTRNN employs a language-based RNN structure to integrate task decomposition and memory management into LLM planning inference, which could effectively improve the faithfulness of LLMs and make the planner more reliable. We conducted experiments in VirtualHome household tasks. Results show that our model significantly improves faithfulness and success rates for complex long-horizon tasks. Website at https://tannl.github.io/FLTRNN.github.io/**

I. Introduction

Task planning is a crucial decision-making process extensively utilized in various robotics applications, including navigation [1], [2], manipulation [3], [4], and everyday household tasks [5]–[7]. In comparison to short-term planning tasks like pick-and-place, tackling complex long-horizon tasks such as food preparation and table cleaning is considered a challenging problem. These tasks involve longer action sequences and interactions with multiple objects or diverse environments. Moreover, long-horizon planning is more susceptible to error propagation, as a mistake in the early stages can lead to deviations in subsequent plans.

Large Language Models(LLMs) [8], [9] have recently been applied to various applications. Trained on vast volumes of unsupervised text data and endowed with numerous parameters, LLMs demonstrate strong capabilities in commonsense reasoning and logical inference. We therefore ask: Can we apply LLMs to complex long-horizon planning tasks?

Several studies have utilized LLMs for task planning [5], [10]–[13]. These methods typically employ an approach known as In-Context Learning (ICL) [14]–[16]. In this paradigm, LLM receives a contextual prompt(includes instructions, demonstrations, etc.) as input, and then outputs the generated plan for the task without fine-tuning, which allows the LLM to learn planning policy from the provided context without the need for specific training data [9], [17].

[1]Zhejiang University
[2]Research Center for Intelligent Robotics, Zhejiang Lab
[3]University of Chinese Academy of Sciences
⋆Corresponding emails: songwei_zju@163.com
sqzhu@sfp.zju.edu.cn

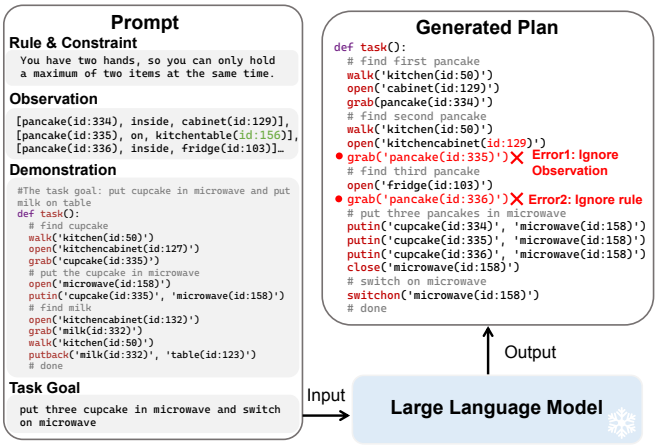

Fig. 1. Illustration of Faithfulness problem of LLMs under In-Context Learning(ICL). ICL requires a contextual prompt containing instruction and demonstration written in natural language. Taking the prompt as input, LLM is responsible for planning inference. However, LLMs may ignore the provided context and make unfaithful planning

While the above methods achieve promising results, a key challenge in applying LLMs to long-horizon tasks lies in their unfaithfulness to the contextual prompt, which provides LLMs with specific rules including task constraints and output formats that the planning process should adhere to. However, LLMs may overlook and ignore these rules [18], [19]. For instance, in Figure 1, the generated plan requires the robot to hold three pancakes at the same time, which ignores the given constraint that the robot only has two hands. Besides, in this example, the LLM also ignores the correct location ID given by the prompt. This unfaithfulness to the given rules can lead to invalid plans and poor execution, and in some cases, even result in dangerous actions, compromising the reliability of the robot.

Motivated by the above problem, we investigate the faithfulness problem of LLMs for complex long-horizon planning tasks. Complex long-horizon tasks typically involve more rules to ensure the generated plan can be executed correctly, which demands that LLMs retain extensive information and handle complex reasoning. However, due to its limited memory and reasoning capacity, LLMs occasionally fail to follow all the rules. This phenomenon is also observed in humans when faced with complex tasks and too many rules. A characteristic aspect of human intelligence is the ability to decompose a complex task into several simpler short tasks and maintain a limited working memory [20]. These processes direct attention and reasoning towards each subtask

individually, only a subset of the rules and constraints need to be considered at a time, effectively alleviating cognitive pressure and improving the faithfulness to the rules.

Inspired by the aforementioned process, we present FLTRNN, a general framework that integrates task decomposition and memory management into LLMs planning inference to improve the faithfulness of LLMs for long-horizon tasks. We first utilize a LLM to break down the long-horizon task into several sub-tasks, forming an initial abstract plan. Then, we propose language-based RNNs to solve each sub-task following the initial plan, which employs a long-short term memory mechanism to maintain the necessary information for sub-task solving, ensuring the LLM can focus on the rules and constraints relevant to the current problem. To further refine the process, we incorporate a rule chain-of-thought(Rule-CoT) and memory graph to enhance the reasoning capabilities of LLMs.

To evaluate the performance of our framework, we conducted experiments in VirtualHome and utilized three distinct datasets. The results indicate that our framework achieved the best performance on all tasks and improved 29% faithfulness and 28% success rate in the NovelScenes dataset. The contributions of our work are summarized as follows:

1) To the best of our knowledge, we are the first to define and investigate the faithfulness problem of LLMs for complex long-horizon task planning.
2) We propose a framework named FLTRNN. Our framework employs language-based RNNs to integrate task decomposition and long-short term memory into LLM planning inference and uses Rule-CoT and memory graph to enhance the reasoning capability of LLMs.
3) We conducted experiments on the virtual household environment VirtualHome. The results indicate our frameworks could effectively improve the faithfulness and success rate in complex long-horizon tasks.

II. Related Works

A. Large Language Models for Task Planning

The advancement of Large Language Models (LLM) [8] has spurred numerous studies exploring their use in task planning in open-ended environments. For instance, Zero-shot Planner [17] employs two LLMs for plan generation: the first LLM decomposes high-level tasks into sensible actions, while the second translates these steps into admissible actions, ensuring the plans' executability. LID [21] employs GPT-2 as a backbone to encode task information, including observations, goals, and history. This encoded information is subsequently input into a policy network, which, after being fine-tuned with specific training data, predicts actions step by step. ProgPrompt [11] introduces a programmatic LLM prompt structure that facilitates plan generation across diverse environments, robot capabilities, and tasks. SayCan [5] grounds LLMs through the value functions of a pre-trained model, empowering them to execute real-world, abstract, long-horizon commands on robots. Inner Monologue [13] leverages environmental feedback to form an inner monologue, fostering reasoning and replanning to achieve complex long-horizon tasks.

B. In-Context Learning Paradigm

Most of the methods mentioned above rely on In-Context Learning (ICL) to prompt LLMs for planning tasks. ICL is a favored paradigm for the utilization of LLMs, especially when compared to the re-training or fine-tuning approaches [21], [22]. It allows LLMs to make inferences only based on contexts augmented with a few examples [14]. ICL offers several significant advantages. First, since instructions and demonstrations are provided in natural language, it facilitates a user-friendly interface for communicating with LLMs [23], [24]. Second, ICL is similar to the analogy processes of human intelligence [25], making the inference procedure more comprehensible. Lastly, ICL doesn't require training or fine-tuning, which reduces the computational costs when adapting LLMs to new tasks and unseen environments [26].

However, for complex long-horizon tasks, the contexts, which include instructions and demonstrations in the prompt, are frequently overlooked, making the plan inexecutable. The faithfulness problem of LLMs for complex long-horizon tasks remains a non-trivial issue to resolve.

III. Preliminaries

A. Problem Definition

A task can be structured as $<S, I, G, T, A>$, where S, I and G represent all possible states, the initial state and the goal state respectively. A is the set of possible actions. T is the transition model, formally defined as $T : S \times A \rightarrow S$, which portrays the environmental changes that occur in response to the implementation of an action. The objective of is to find a plan π, a sequence of actions, that transitions the initial state I to the goal state G. There is currently no strict definition to distinguish complex long-horizon tasks. Generally, compared to short-term planning, complex long-horizon tasks involve longer action sequences, ranging from 10 to 40 steps or even longer [21], and interactions with a greater variety of items and environments.

Moreover, there are two distinct settings for long-horizon planning. The first is open-loop planning which does not involve any environmental observation except for the initial state. The second setting introduces environmental feedback into the generation process, also known as interactive planning. Our framework can accommodate both settings.

B. Faithfulness Problem of In-Context Learning

Faithfulness refers to the LLM's adherence to the context information provided to it. The concept was first introduced in question answering [27]. It denotes the factual consistency between the response and the source documents.

Regarding the faithfulness issue in task planning, LLMs might overlook the rules or constraints mentioned in the instructions or misinterpret the planning format presented in the demonstration. Formally, given a contextual prompt that includes a set of rules $R = \{r_1, r_2, \ldots, r_n\}$, a plan π

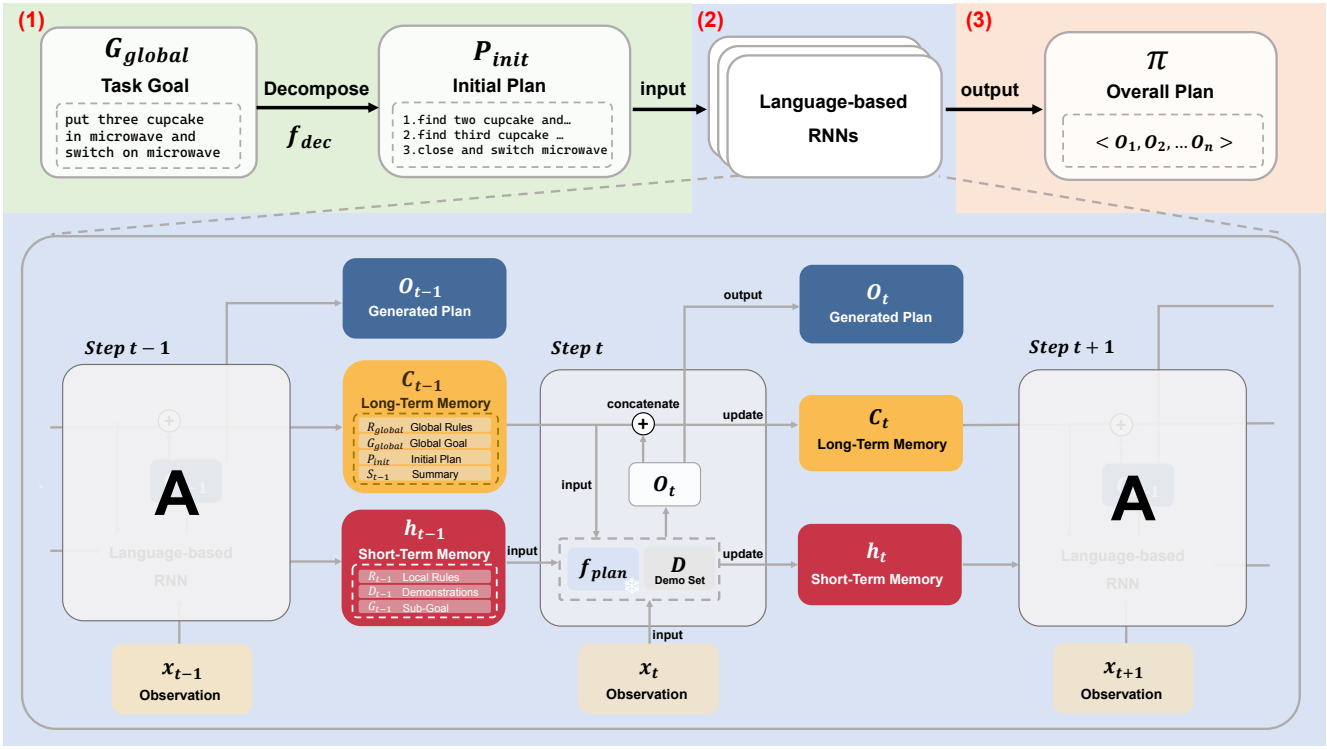

Fig. 2. The framework of FLTRNN. Our framework takes the task goal as input and produces the task plan as output. The framework comprises three stages: 1. Decompose a long-horizon task into several simpler sub-tasks and formulate an initial plan. 2. Use Language-Based RNNs to solve each sub-task in the initial plan, in which the task goal, initial plan, and instructions are represented as long-term memory, while the selected sub-goal in the plan, demonstration, and specific details of the sub-task are designated as short-term memory. 3. Aggregate the plans generated by the RNNs to form the overall task plan. Besides, the rule Chain-of-Thought(Rule-CoT) and memory graph are used to enhance the reasoning ability of LLMs.

generated by an LLM is said to be 'faithful' if and only if every action a in π satisfies all rules r_i in R:

$$\text{Faithfulness}(\pi, R) = \begin{cases} 1, & \text{if } \forall a \in \pi, \forall r_i \in R, a \models r_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Here, $\models$ denotes that the action a satisfies the rule r_i.

The lack of faithfulness can have detrimental effects on the executability of a plan and may even lead to dangerous actions. When a plan deviates from the specified guidelines, it can result in catastrophic outcomes, such as equipment malfunctions or direct threats to human safety.

IV. Methodology

In this section, we present the details of our proposed framework, FLTRNN. As illustrated in Figure 2, our framework takes the task goal as input and produces the task plan as output. The framework comprises three stages: 1. Decompose a complex, long-horizon task into several simpler sub-tasks and formulate an initial plan. 2. Use Language-Based RNNs to solve each sub-task in the initial plan, in which the task goal, initial plan, and instructions are represented as long-term memory, while the selected sub-goal in the plan, demonstration, and specific details of the sub-task are designated as short-term memory. 3. Aggregate the plans generated by the RNNs to form the overall task plan. Besides, the rule Chain-of-Thought(Rule-CoT) and memory graph are used to enhance the reasoning ability of LLMs.

A. Task Decomposition

Human intelligence enhances adherence and faithfulness to rules and constraints by breaking down complex tasks and effectively managing working memory. Inspired by this, we employ LLM to decompose long-horizon tasks. This decomposition can be represented as $P_{init} = f_{dec}(G)$, where f_{dec} is an LLM equipped with a decomposition prompt. It's worth noting that although decomposing complex tasks and managing memory are not new concepts in task planning(such as hierarchical planning [28]), we are the first to use these approaches to enhance the faithfulness of LLMs and make the planner more reliable.

B. Language-based RNN Blocks

To integrate the task decomposition and memory management into LLM planning inference, we employ an RNN framework. The classical RNN can be represented as:

$$o_t, h_t, c_t = \text{RNN}(o_{t-1}, x_t, h_{t-1}, c_{t-1}), \quad (2)$$

where x is the input, o is the output, h is the hidden state, and c is the cell state. To adapt this to LLM, following [29], we implemented the framework based on natural language, simulating the process with prompts. This process can be described as:

$$o_t, h_t, c_t = \text{NLRNN}(x_t, h_{t-1}, c_{t-1}, \theta). \quad (3)$$

Here θ is the parameter of LLM, x represents the environment observation which could be obtained by vision

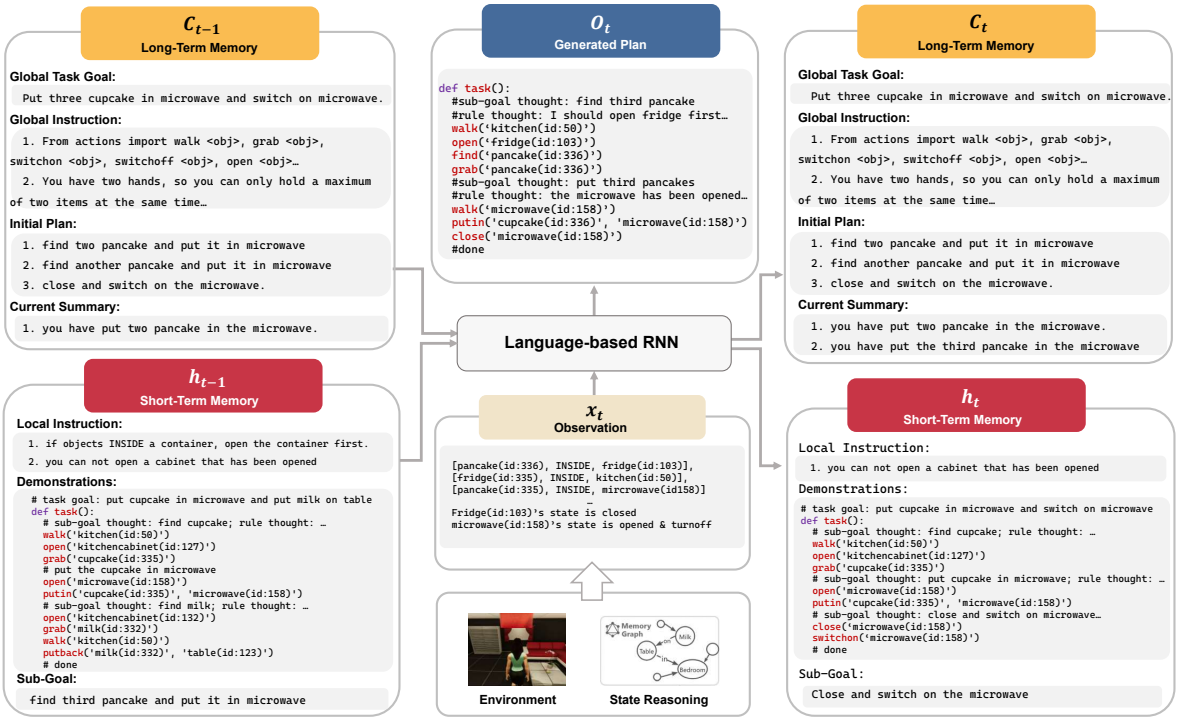

Fig. 3. The detailed illustration of FLTRNN. The language-based RNN takes long-short term memory and observation as input and then outputs the generated plan and updated memory.

models [30], [31], $o_t = f_{plan}(c_{t-1}, h_{t-1}, x_t)$ is the output plan for the sub-task. f_{plan} is implied by an LLM with a planning prompt(the full planning prompt can be seen in the Figure 3). The cell state c embodies long-horizon memory, encapsulating task goals, abstract plans, and summaries of previous actions, while h, the hidden state, symbolizes short-term memory, comprising sub-goals, demonstrations, and task-specific instructions. Language-based RNNs naturally integrate the update and management of memory with LLM's planning inference which effectively alleviates the burden of reasoning and memory on LLMs, thereby enhancing adherence to task instructions.

C. Long-Short Term Memory

Previous work such as [11] involved all the necessary information for planning directly into the prompt. For complex long-horizon tasks, these prompts typically tend to be extremely lengthy. This puts escalating pressure on the LLMs' memory capacities.

To tackle this challenge, we employ a long-short term memory mechanism. The long-term memory, denoted as c_t, primarily manages the information vital for addressing the overarching task. It is defined as $c_t = (R_{global}, G_{global}, P_{initial}, S_t)$, where:

- R_{global} denotes global rules.
- G_{global} represents the overall task goals.
- $P_{initial}$ refers to the initial abstract plan.
- S_t is a summary of previous actions.

Here $S_t = \text{CONCAT}(S_{t-1}, G_t)$. The short-term memory h_t mainly manages memory information required for addressing the current sub-task, given by $h_t = (R_t, D_t, G_t)$, where:

- R_t represents instructions and rules specific to the current task.
- D_t denotes demonstrations for the current sub-goal.
- $G_t = \text{Sample}(P_{initial} \cap S_t)$ is the sub-goal.

The R_t and D_t are samples from an external demonstration set using the sentence similarity of the task goal.

The introduction of short-term memory significantly improves the planning process. By focusing exclusively on the current sub-goal, the LLM can plan more efficiently.

D. Enhanced Reasoning

A primary factor contributing to unfaithfulness is the limited reasoning ability of the LLM. To address this and bolster faithfulness, we adopt two strategies to enhance the LLM's reasoning capabilities: the rule Chain-of-Thought and memory graph.

Research suggests that a Chain of Thought (CoT) can effectively guide LLMs in reasoning, thereby enhancing stability [19]. In light of this, we introduce the Rule-CoT. As illustrated in Figure 5, the LLM does not simply rely on rules from the instruction segment of the prompt during the planning phase. Instead, throughout the planning of each step, it continuously revisits, contemplates, and reasons based on these rules. Such an approach considerably augments the model's adherence to the given instructions.

Furthermore, in the open-loop planning setting, there is no direct feedback or observation from the environment. This requires the LLM to constantly reason about the state of the environment to ensure coherent planning. Such continuous reasoning imposes a latent burden on the LLM. To alleviate this reasoning burden, we introduce an external world-state

Fig. 4. Example of our frameworks for long-horizon task planning.

TABLE I

OVERALL PERFORMANCE FLTRNN AND BASELINES ACROSS VARIOUS DATASETS. OUR FRAMEWORK OUTPERFORMS ALL BASELINES.

	In-Distribution		**NovelScenes**		**NovelTasks**	
	Faithfulness	Coherence	Faithfulness	Coherence	Faithfulness	Coherence
Planning-Only	95.33 ± 0.67	98.33 ± 0.33	56.67 ± 3.18	74.33 ± 2.03	62.33 ± 2.73	79.67 ± 2.03
Planning-Reasoning	90.67 ± 2.85	97.33 ± 0.33	84.33 ± 1.86	90.33 ± 1.45	80.67 ± 1.86	86.33 ± 0.88
Our	**98.00 ± 0.00**	**99.33 ± 0.33**	86.00 ± 1.15	**94.33 ± 1.20**	88.33 ± 1.20	92.33 ± 2.03
Our + obs	**98.00** ± 0.58	**99.33 ± 0.33**	**91.00 ± 1.15**	93.33 ± 1.45	**92.00 ± 1.53**	**96.67 ± 1.20**
	Correctness	Success Rate	Correctness	Success Rate	Correctness	Success Rate
Planning-Only	87.33 ± 2.02	82.33 ± 1.76	38.67 ± 1.45	32.33 ± 1.45	49.67 ± 3.18	49.00 ± 3.21
Planning-Reasoning	79.67 ± 3.38	79.33 ± 3.18	54.33 ± 1.76	53.33 ± 1.76	47.33 ± 1.67	46.00 ± 1.15
Ours	**92.33** ± 2.19	**86.00** ± 2.65	70.67 ± 2.67	60.67 ± 0.33	70.67 ± 3.06	70.00 ± 3.21
Our+obs	92.00 ± 1.15	**86.00 ± 1.00**	**74.00 ± 1.15**	**64.33** ± 1.76	**76.00** ± 2.00	**75.33** ± 2.40

```
def task():
    ...
    # Rule Thought: Put the pancake (id:334) in the microwave (id:158).
    The microwave is closed, so we should open the microwave first.
    # Rule Thought: You have grabbed two pancakes. Remember, you should
    grab only one item at a time, so put one pancake down first.
    find('microwave(id:158)')
    open('microwave(id:158)')
    putin('pancake(id:334)', 'microwave(id:158)')
    close('microwave(id:158)')
    ...
```

Fig. 5. Example of Rule CoT.

reasoning, termed the memory graph. This module aids in inferring environmental changes during the planning process. The graph is defined by $\{V, R\}$, where V denotes the set of nodes, representing observed objects, and R designates the relationships between these objects(e.g., on and in). The update process of the graph operates under the assumption that the previous action has been successfully executed.

V. EXPERIMENTS

In this section, we evaluate our framework by conducting experiments in a virtual household environment.

A. Experimental Setup

Environment. We conducted our evaluation on VirtualHome [32], which is a realistic 3D environment that encompasses a collection of realistic 3D homes and objects. These objects can be manipulated to carry out household tasks.

Datasets. We conduct experimental testing on three datasets provided by [21]: **In-Distribution**, **NovelTasks**, and **NovelScenes**. The three datasets vary in task complexity. We use action steps to measure the task complexity of each individual task. The average action steps for In-Distribution, NovelTasks, and NovelScenes are 13.4, 25.61, and 27.11, respectively. For each dataset, we conducted experiments on 100 tasks three times to obtain the results.

Compared Methods. We compare our method with three categories of methods: 1. **Planning-Only**: ProgPrompt [11], where the LLM directly outputs the planning result using the ICL. 2. **Planning-Reasoning**: Inner Monologue [13], wherein a reasoner is incorporated into the planning process to bolster reasoning capabilities. 3.**Variant Models of FLTRNN**: **Ours+obs** obtains the environmental state through observation, while **Ours** utilizes a memory graph for state reasoning. For all the methods compared, we employ GPT-3.5-Turbo as the backbone LLM.

Metrics. We evaluated the performance from two perspectives: Following [33], we use **Faithfulness** and **Coherence** as metrics to evaluate faithful reasoning ability, where **Faithfulness** evaluates whether the plan is entirely faithful to the instruction and **Coherence** examines if the action conflicts with previous actions. Besides, drawing from [17], [21], we employ **Success Rate** and **Correctness** to measure whether the generated plan can be executed and if it accomplishes the

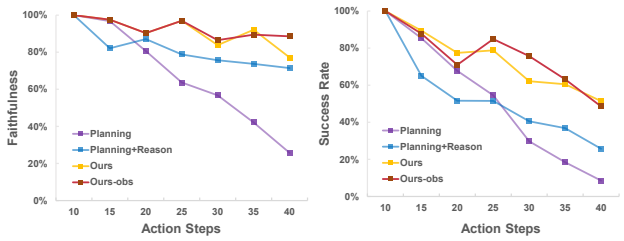

Fig. 6. Result of faithfulness and success rate with different action steps. FLTRNN effectively improves the performance of long-horizon tasks

task's goal, where **Success Rate** evaluates if the output plan achieves the given task's goal and **Correctness** examines if the output plan possesses semantic correctness.

B. Results

The primary results are presented in Table I. We can observe that: 1. Across all metrics and datasets, our framework consistently outperforms other methods. This underscores our method's ability to effectively boost faithfulness, which in turn enhances the success rate. An illustrative example of FLTRNN applied to a long-horizon task can be seen in Figure 4. 2. In the majority of datasets, our model exhibits the lowest standard deviation. This suggests that our framework not only elevates faithfulness and success rate but also augments stability and robustness. 3. There is a noticeable trend wherein results with subpar success rates correspond with low faithfulness. This accentuates the prevailing issue of faithfulness: deviations from the provided context can significantly compromise the successful execution of a plan. 4. Among the evaluated methods, the **Planning-Only** method excels in In-Distribution. Conversely, in the NovelScenes, the **Planning-Reasoning** method registers superior performance. This could imply that while reasoners are beneficial for intricate tasks, they might introduce superfluous, potentially misleading information in simpler tasks, thus diminishing performance. Moreover, the variation in faithfulness and success rates across different datasets suggests that LLM's faithfulness could be influenced by task complexity.

VI. Analysis and Discussion

A. Faithfulness with Task Complexity

To further investigate the relationship between task complexity and the faithfulness problem of LLMs, we conducted correlation analysis on the tasks of above three datasets. The results are illustrated in Figure 6. Our observations are as follows: 1. Across all action step intervals, our framework consistently surpasses the baselines, particularly for tasks with a high number of steps. This implies that our model is particularly adept at enhancing both faithfulness and the success rate for complex long-horizon tasks. 2. While the baseline model shows impressive faithfulness and success rates in the 15-step interval, its performance wanes for tasks with larger step counts. Notably, we identified that for tasks with fewer action steps, the Planning-Only approach fares better than the Planning-Reasoning method. Yet, as the action steps increase, the Planning-Reasoning method excels over Planning-Only. This reinforces our prior assertion that the reasoner is particularly well-suited for long-horizon tasks. These findings emphasize the robust relationship between an LLM's faithfulness and the complexity of the task. As tasks grow more complex and their action sequences lengthen, it becomes increasingly formidable for the LLM to maintain strict adherence to the instructions. Through adept task decomposition, memory management and reasoning enhancement, our framework effectively improves LLM's faithfulness and success rate.

B. Ablation Study

TABLE II
Ablation of various modules of FLTRNN in NovelTasks. All design modules contribute to the performance.

	Faithfulness	Success Rate
Ours full	88.33 ± 1.20	70.00 ± 3.21
w/o Decomposition	77.67 ± 4.10	54.67 ± 2.19
w/o LSTM	85.67 ± 0.67	63.67 ± 0.67
w/o Rule-CoT	80.33 ± 1.76	58.33 ± 1.20

To comprehensively demonstrate the effectiveness of each module in our framework, we conducted an ablation study on the NovelTasks. The results are detailed in Table II. In the **w/o Decomposition** model, planning is generated without decomposing the task. Results suggest that subdividing a complex task into several simpler tasks considerably improves faithfulness and success rate. For the **w/o LSTM** model, the LSTM module is removed from the RNN. In this setup, every RNN is given the same prompt containing all task information. Compared to our complete model, the w/o LSTM variant shows a 2.66% decrease in faithfulness, resulting in a 6.33% decline in success rate. This indicates that utilizing LSTM for memory management significantly enhances instruction adherence. The **w/o Rule-CoT** model omits the Rule-CoT from the demonstration. Against our full model, this variant displays an 8% drop in faithfulness and an 11.67% reduction in success rate. Such results highlight the pivotal role of Rule-CoT in enhancing compliance with the rules and constraints specified in the instructions.

VII. Conclusion

To the best of our knowledge, we are the first to define and investigate the faithfulness problem of LLMs for complex long-horizon tasks. Drawing inspiration from human intelligence, we propose a novel framework named FLTRNN which integrates task decomposition and memory management into LLM planning inference. Additionally, we design Rule-CoT and a memory graph to enhance the reasoning ability of LLM. Experiments in a virtual household environment suggest that our framework can effectively improve both faithfulness and the success rate in long-horizon tasks, making robots more reliable and trustworthy.

VIII. Acknowledgement

This research was supported by National Natural Science Foundation of China(Grant No. U21A20488) and Key Research Project of Zhejiang Lab(Grant No. G2021NB0AL03).

REFERENCES

[1] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," *arXiv preprint arXiv:2210.05714*, 2022.

[2] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *Microsoft Auton. Syst. Robot. Res*, vol. 2, p. 20, 2023.

[3] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid, "Instruction-driven history-aware policies for robotic manipulations," in *Conference on Robot Learning*. PMLR, 2023, pp. 175–187.

[4] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A generalist agent," *Transactions on Machine Learning Research*, 2022.

[5] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning*, 2023, pp. 287–318.

[6] J. Mai, J. Chen, B. Li, G. Qian, M. Elhoseiny, and B. Ghanem, "Llm as a robotic brain: Unifying egocentric memory and control," *arXiv preprint arXiv:2304.09349*, 2023.

[7] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023.

[8] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[10] Z. Wang, S. Cai, A. Liu, X. Ma, and Y. Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," *arXiv preprint arXiv:2302.01560*, 2023.

[11] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," *arXiv preprint arXiv:2209.11302*, 2022.

[12] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," *arXiv preprint arXiv:2212.04088*, 2022.

[13] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," 2022.

[14] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[15] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 11 048–11 064.

[16] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," in *The Eleventh International Conference on Learning Representations*, 2022.

[17] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*, 2022, pp. 9118–9147.

[18] W. Zhou, S. Zhang, H. Poon, and M. Chen, "Context-faithful prompting for large language models," *arXiv preprint arXiv:2303.11315*, 2023.

[19] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, and C. Callison-Burch, "Faithful chain-of-thought reasoning," *arXiv preprint arXiv:2301.13379*, 2023.

[20] A. Baddeley, "Working memory," *Science*, vol. 255, no. 5044, pp. 556–559, 1992.

[21] S. Li, X. Puig, C. Paxton, Y. Du, C. Wang, L. Fan, T. Chen, D.-A. Huang, E. Akyürek, A. Anandkumar, *et al.*, "Pre-trained language models for interactive decision-making," vol. 35, 2022, pp. 31 199–31 212.

[22] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," *arXiv preprint arXiv:2305.15021*, 2023.

[23] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 2655–2671.

[24] J. Liu, D. Shen, Y. Zhang, W. B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for gpt-3?" in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022, pp. 100–114.

[25] P. H. Winston, "Learning and reasoning by analogy," *Communications of the ACM*, vol. 23, no. 12, pp. 689–703, 1980.

[26] T. Sun, Y. Shao, H. Qian, X. Huang, and X. Qiu, "Black-box tuning for language-model-as-a-service," in *International Conference on Machine Learning*. PMLR, 2022, pp. 20 841–20 855.

[27] S. Longpre, K. Perisetla, A. Chen, N. Ramesh, C. DuBois, and S. Singh, "Entity-based knowledge conflicts in question answering," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 7052–7063.

[28] P. Bercher, R. Alford, and D. Höller, "A survey on hierarchical planning-one abstract idea, many concrete realizations." in *IJCAI*, 2019, pp. 6267–6275.

[29] W. Zhou, Y. E. Jiang, P. Cui, T. Wang, Z. Xiao, Y. Hou, R. Cotterell, and M. Sachan, "Recurrentgpt: Interactive generation of (arbitrarily) long text," *arXiv preprint arXiv:2305.13304*, 2023.

[30] Q. Meng, S. Ji, S. Zhu, T. Jin, T. Li, J. Gu, and W. Song, "Rffce: Residual feature fusion and confidence evaluation network for 6dof pose estimation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2876–2883.

[31] Q. Meng, J. Gu, S. Zhu, J. Liao, T. Jin, F. Guo, W. Wang, and W. Song, "Kgnet: Knowledge-guided networks for category-level 6d object pose and size estimation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 6102–6108.

[32] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8494–8502.

[33] O. Golovneva, M. Chen, S. Poff, M. Corredor, L. Zettlemoyer, M. Fazel-Zarandi, and A. Celikyilmaz, "Roscoe: A suite of metrics for scoring step-by-step reasoning," *arXiv preprint arXiv:2212.07919*, 2022.