Prospector: Improving LLM Agents with Self-Asking and Trajectory Ranking

Byoungjip Kim[1], **Youngsoo Jang**[1], **Lajanugen Logeswaran**[1], **Geon-Hyeong Kim**[1], **Yu Jin Kim**[1], **Honglak Lee**[1], **Moontae Lee**[1,2]
[1] LG AI Research, [2] University of Illinois Chicago
{bjkim, youngsoo.jang, llajan, ghkim, yujin.kim}@lgresearch.ai
{honglak, moontae.lee}@lgresearch.ai

Abstract

Large language models (LLMs) have shown the ability to solve complex decision-making tasks beyond natural language processing tasks. LLM agents based on few-shot in-context learning (ICL) achieve surprisingly high performance without training. Despite their simplicity and generalizability, ICL-based agents are limited in their ability to incorporate feedback from an environment. In this paper, we introduce **Prospector**, an LLM agent that consists of two complementary LLMs, an Actor and a Critic. To elicit better instruction-aligned actions from the LLM agent, we propose *AskAct prompting* that performs an additional self-asking step such as goal and progress checking before generating an action. Furthermore, to implicitly incorporate the environment feedback, we propose *Trajectory Ranking* that orders generated trajectories by predicting the expected total reward. Prospector encourages the LLM Actor to generate diverse (creative) trajectories, and harnesses the LLM Critic to select the most rewarding trajectory. On representative decision-making benchmark environments such as ALFWorld and WebShop, we empirically demonstrate that Prospector can considerably increase the success rate of given tasks, while outperforming recent advancements such as ReAct and Reflexion.

1 Introduction

Large language models (LLMs) (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020; Raffel et al., 2020; Touvron et al., 2023) have achieved remarkable success in natural language processing (NLP). Recently, LLM-based autonomous agents (Yao et al., 2023; Shinn et al., 2023; Song et al., 2024; Muthusamy et al., 2023; Shao et al., 2023) have shown the ability to solve complex decision-making tasks such as householding (Shridhar et al., 2021), science experiments (Wang et al., 2022), web navigation (Yao et al., 2022), computer control (Kim et al., 2024), etc. Current LLM agents can be broadly categorized into (1) in-context learning (ICL)-based LLM agents (Yao et al., 2023; Shinn et al., 2023; Kim et al., 2024), and (2) fine-tuning (FT)-based LLM agents (Chen et al., 2023b,a; Song et al., 2024).

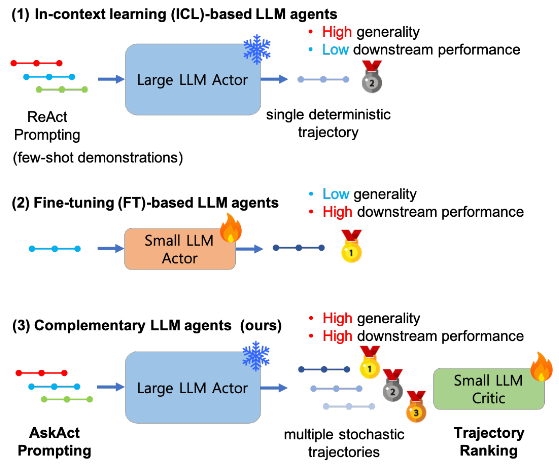


Figure 1: High level comparison of LLM agent approaches. We compare three main approaches: (1) in-context learning (ICL)-based LLM agents, (2) fine-tuning (FT)-based LLM agents, and (3) complementary LLM agents (ours). Prospector is designed to combine the advantages of the both worlds: in-context learning and fine-tuning. Prospector consists of two complementary LLMs such as a very large LLM Actor and a relatively small LLM Critic.

ICL-based LLM agents such as ReAct (Yao et al., 2023) can solve interactive decision-making tasks by imitating the few-shot demonstrations given in the prompt. They achieve surprisingly high performance without training. Despite the simplicity and generalizability, the ICL-based approaches lack optimizing trajectories based on the reward from an environment. In contrast, FT-based LLM agents such as FireAct (Chen et al., 2023a) and ETO (Song et al., 2024) have been recently introduced. Even though they can achieve high performance on a target task by fine-tuning the LLM agent based on environment interaction, they may not be gener-

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 14958–14976
November 12-16, 2024

ally applicable to diverse tasks due to the out-of-distribution (OOD) problem.

In this paper, we introduce **Prospector**, a LLM agent that consists of two complementary LLMs such as LLM Actor and LLM Critic. As shown in Figure 1, Prospector is designed to combine the advantages of the both worlds: in-context learning and fine-tuning. Prospector uses a very large LLM (GPT-3-175B (Brown et al., 2020) or Llama-2-70B (Touvron et al., 2023)) for the LLM Actor, and the LLM Actor generates a language action for a given language observation by leveraging the ability of few-shot in-context learning of LLMs. To align the LLM Actor on an environment, Prospector uses a relatively small LLM as the LLM Critic that can rank candidate trajectories by predicting the expected total reward from the given environment. The LLM Critic is fine-tuned on trajectory-reward paired data to precisely reflect the rewards from the environment. We would like to note that Prospector can not only maintain the strong generality of the LLM Actor by using the ability of in-context learning, but also effectively reflect rewards from an environment by fine-tuning the LLM Critic. Also, we would like to highlight that simply adding a LLM Critic fine-tuned on a specific environment, Prospector can effectively expand decision-making tasks that can be solved without fine-tuning the LLM Actor with huge parameters.

More specifically, to improve ICL-based LLM agents, Prospector provides two techniques: (1) AskAct prompting for LLM Actor, and (2) Trajectory Ranking (TR) for LLM Critic.

First, even though ICL-based LLM agents such as ReAct provide a surprisingly high success rate, it can suffer from hallucination such as unaligned action generation. To alleviate hallucination and elicit better instruction-aligned actions from the LLM Actor, we propose *AskAct prompting* that performs an additional self-asking step such as goal and progress checking before generating an action. This allows the LLM Actor to collect the information necessary for decision-making by itself, and help to elicit better instruction-aligned actions than reason-only prompting such as ReAct.

Second, ICL-based LLM agents lack reflecting environment feedback when generating actions. To implicitly incorporate environment feedback, we provide *Trajectory Ranking (TR)* that can order generated trajectories by predicting the expected total reward. Prospector encourages the LLM Actor to generate diverse (creative) trajectories at high temperature, and harnesses the LLM Critic to select the most rewarding trajectory among the candidates. Since fine-tuning a LLM agent from environment feedback is often prohibitive due to its enormous training cost, we consider fine-tuning a relatively small LLM Critic by constructing a (trajectory, reward) dataset. Prospector can achieve high performance through the synergy of AskAct prompting, which can generate promising trajectory candidates, and Trajectory Ranking, which can select the most rewarding trajectory from the candidates.

In the experiments, we demonstrate that Prospector outperforms recent advancements such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023) on the representative language-based interactive decision-making benchmarks including ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022).

The contributions of this paper can be summarized as follows:

- We introduce **Prospector**, a LLM agent that consists of two complementary LLMs, an Actor and a Critic, to improve the performance of ICL-based LLM agents (see Figure 1).
- We provide *AskAct prompting* that introduces additional self-asking steps in a ReAct prompt to improve the performance of LLM Actor (see Figure 3). Also, we provide *Trajectory Ranking (TR)* that selects the most rewarding trajectories among diverse trajectories generated by LLM Actor (see Figure 2).
- We empirically demonstrate that Prospector can provide better success rate than ICL-based LLM agents such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023) on two representative decision-making environments such as ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022) Table 1 and Table 4).

2 Method

The overview of Prospector is shown in Figure 2. Prospector consists of two complementary LLMs such as the LLM Actor and LLM Critic.

The LLM Actor generates a text-based action a_t conditioned on a few-shot demonstration $\tau^d_{1:T} = \{o^d_1, k^d_1, a^d_1, ..., o^d_T, k^d_T, a^d_T\}$, a history of the current interaction $\tau_{1:t-1} =$

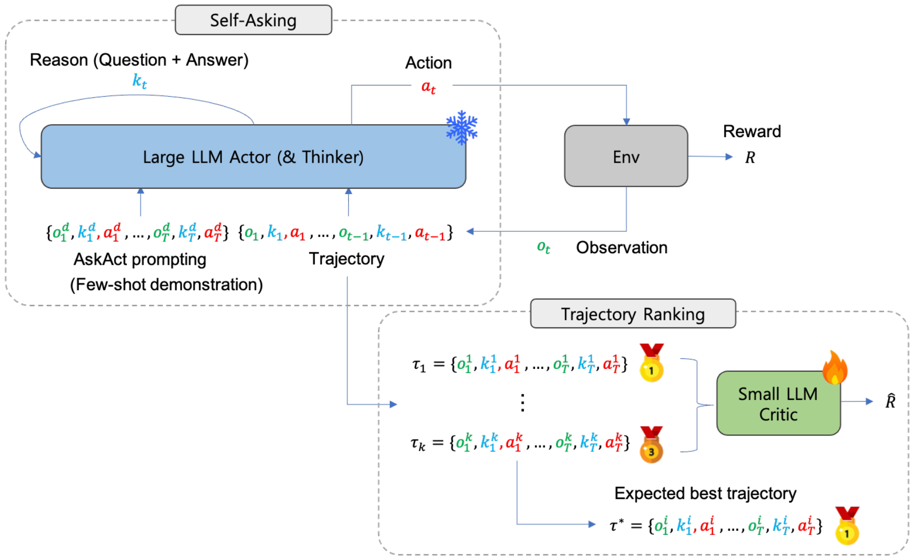


Figure 2: Overview of Prospector. Prospector is a LLM agent that consists of two complementary LLMs such as LLM Actor and LLM Critic for solving complex interactive decision-making tasks. LLM Actor generates actions conditioned on the few-shot demonstrations and the history of observations and actions. To elicit the more proper actions, Prospector LLM Actor uses *AskAct prompting* that interleaves self-asking steps in the few-shot demonstrations. Furthermore, to implicitly incorporate environment feedback, we introduce *Trajectory Ranking* which harnesses LLM Critic that can select the most rewarding trajectories by predicting the expected total reward.

$\{o_1, k_1, a_1, ..., o_{t-1}, k_{t-1}, a_{t-1}\}$, a current observation o_t, and a current think k_t. Formally, it can be denoted as $a_t \sim \pi_\theta(a_t|\tau^d_{1:T}, \tau_{1:t-1}, o_t, k_t)$, where π_θ denotes the probability distribution induced by the LLM Actor. Unlike ReAct-based LLM agents, Prospector LLM Actor uses *AskAct prompting*. We introduce AskAct prompting to alleviate the hallucination of ReAct prompting and elicit better actions from the LLM. We provide the details of AskAct prompting in the following subsection.

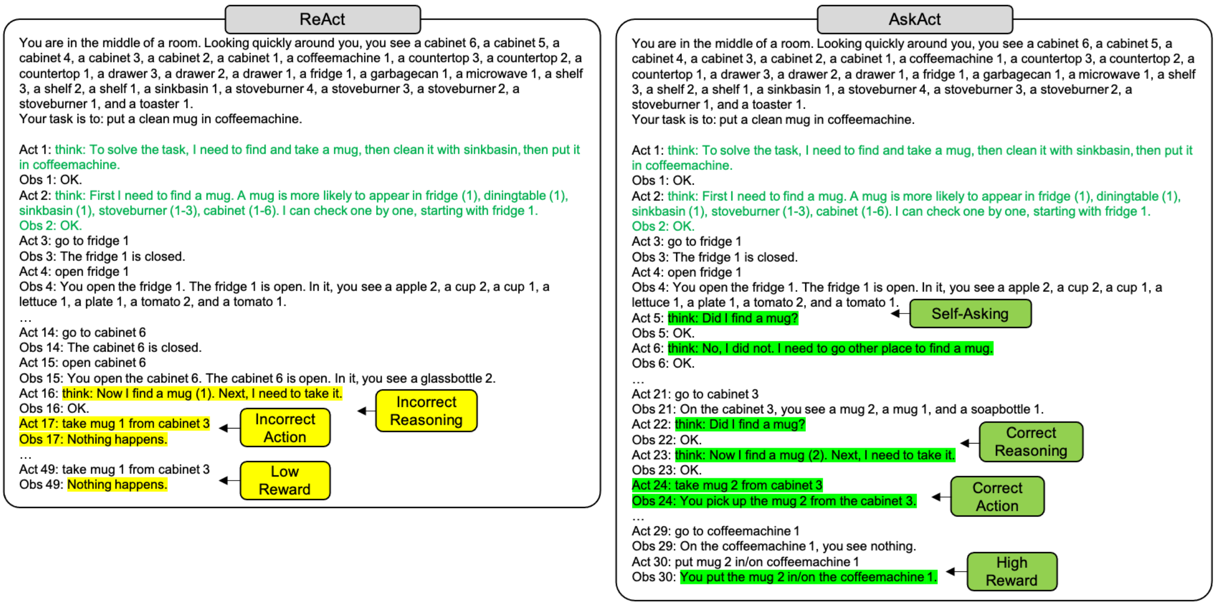
One of the key ideas of Prospector is to encourage LLM Actor to generate diverse (creative) trajectories and leverage LLM Critic to select the most rewarding trajectory among generated trajectories. The LLM Critic is responsible for performing *Trajectory Ranking (TR)* that orders trajectories $\{\tau_i\}_{i=1}^n$ based on the expected total reward $\hat{R}$ from an environment. More concretely, the ranked ordering of the trajectories is given by $(\tau_1, ..., \tau_n)$ such that $(\bar{R}_1 > ... > \bar{R}_n)$. We provide more details of Trajectory Ranking in the following subsection.

2.1 AskAct Prompting

First, we briefly summarize the basic operation of ReAct (Yao et al., 2023), one of the most representative LLM agent. ReAct mainly leverages the ability that a LLM can generate a plausible language action (e.g., "pick up a mug") when a language observation is given (e.g., "You opened a cabinet. You see a mug."). To generate a more proper language action for a given language observation, the LLM agent is usually given few-shot demonstrations in the input prompt. The demonstration typically contains a task description, an instruction (or a goal) of the task, a language observation at each time step, and a language action at each time step. An example few-shot demonstration for ALFWorld (Shridhar et al., 2021) (a text-based embodied environment) is shown on the left of Figure 3. Given the few-shot demonstration, the LLM agent can generates a more proper action by imitating the demonstrations.

AskAct prompting is based on ReAct prompting. ReAct prompting introduces additional reasoning steps between a language observation and a language action in the demonstration, and shows that the interleaved reasoning steps can help the LLM agent to generate better language actions. This approach is mainly inspired by the fact that Chain-of-Thought (CoT) prompting (Wei et al., 2022) improves the ability of LLMs in reasoning tasks.

AskAct prompting further adds self-asking steps between a language observation and a language action in the demonstration of the input prompt. For example, as shown in Table 12 in the Appendix, on ALFWorld, we manually add a question (e.g., "Did I find a spraybottle?") that can help to accomplish the given task (e.g., "Your task is to: put



(a) ALFWorld

Figure 3: Comparison of ReAct and AskAct. AskAct is a prompting method that introduces additional self-asking steps in a ReAct prompt. In ALFWorld, the self-asking step checks if a target object is found. This can elicit a correct action by alleviating hallucination.

some spraybottle on toilet.") and its answer (e.g., "No, I did not. I need to go another place to find a spraybottle."). As shown in Figure 3, given the demonstration of self-asking in the input prompt, the LLM agent can now ask a similar question (i.e., "Did I find a mug?") when a new task (i.e., "Your task is to: put a mug in coffeemachine.") is prompted. Through this self-asking step, the LLM agent can avoid an improper language action (e.g., "take mug form cabinet.") in case where a mug is not found.

Concrete examples of AskAct prompts can be found in Table 12 and Table 14 in the Appendix. We empirically show that AskAct considerably improves the success rate compared to ReAct (see Table 1 and Table 4).

2.2 Trajectory Ranking

Trajectory Ranking (TR) orders trajectories based on the expected total reward. To enable Trajectory Ranking, Prospector introduces LLM Critic that generates a text-based expected total reward y for a given trajectory $\tau_{1:T} = \{o_1, k_1, a_1, ..., o_T, k_T, a_T\}$. Formally, this can be denoted as $y \sim P_\phi(y|\tau_{1:T})$, where P_ϕ denotes the probability distribution induced by the LLM Critic. Note that a language-based expected total reward y is converted into a real-valued expected total reward $\hat{R}$ by using a function $\hat{R} = f(\hat{y})$. For example, in ALFWorld that provides a binary reward (e.g., 0 and 1), the text-to-value mapping function $f(y)$ can be implemented as follows:

$$\hat{R} = f(\hat{y}) = \begin{cases} 1 & \text{if } \hat{y} = \text{"success"} \\ 0 & \text{if } \hat{y} = \text{"fail"} \end{cases} \tag{1}$$

Then, to improve the estimation quality, we evaluate the expected total reward for a given trajectory $\tau_{1:T}$ by predicting the total reward with multiple times as follows:

$$\bar{R} = \mathbb{E}_{\hat{y}_i \sim P_\phi(y|\tau_{1:T})} f(\hat{y}_i). \tag{2}$$

More concretely, if the LLM Critic generates $\{\text{"success"}, \text{"success"}, \text{"fail"}\}$ in a multiple sampling, then the expected return will be $0.66 = (1+1+0)/3$.

Few-shot LLM Critic. LLM Critic can be implemented by using few-shot prompting. Formally, this can be denoted as $y \sim P_\phi(y|d, \tau_{1:T})$, where d denotes a few-shot prompt. We provide the critic prompt template used for few-shot reward prediction in Table 6. More concrete examples of critic prompts can be found in the Table 13 and Table 15 in the Appendix.

Fine-tuned LLM Critic. In some complex environments such as WebShop (Yao et al., 2022), even powerful LLMs such as GPT-3 have difficulty in reward prediction in a few-shot manner (see Table 9). To address this problem, we provide a fine-tuned LLM Critic that is relatively small open-source LLM fine-tuned on a (trajectory, reward) dataset.

The first step is to construct a (trajectory, reward) dataset. Given a set of trajectories generated by the LLM actor $\mathcal{T} = \{\tau_i\}_{i=1}^N$, we construct a labeled dataset $\mathcal{D}_{SFT} = \{\tau_i, y_i\}_{i=1}^N$ by using environment return R_i as a language-based label y_i for a trajectory τ_i. Here, we use an inverse mapping function $f^{-1}(R_i)$ to convert a real-valued total reward to a language-based total reward. For example, we use the inverse mapping function $f^{-1}(R_i)$ in ALFWorld.

$$y_i = f^{-1}(R_i) = \begin{cases} \text{"success"} & \text{if } R_i = 1 \\ \text{"fail"} & \text{if } R_i = 0 \end{cases} \tag{3}$$

The second step is to train a model P_ϕ parameterized by ϕ. Given a labeled trajectory dataset $\mathcal{D}_{SFT} = \{\tau_i, y_i\}_{i=1}^N$, we formulate the problem as maximum likelihood estimation, and train P_ϕ by using a negative log likelihood objective.

$$\mathcal{L}_{SFT}(\phi) = -\sum_{i=1}^{N} \log P_\phi(y_i|\tau_i) \tag{4}$$

Finally, we would like to emphasize that AskAct and TR provide complementary benefits in improving LLM agents in terms of both performance and efficiency. Since AskAct provides a better baseline, AskAct and TR can achieve much better performance with less sampling.

Remarks on Trajectory Ranking. The Critic (i.e., reward estimator) of Prospector is valuable, since Trajectory Ranking does not use the final total reward from the environment. In many cases, it is hard or limited to receive the final total reward from the environment. For example, in WebShop (a simulated online shopping environment), the final total reward is given only when a LLM agent clicks on the "buy" button. However, purchasing potentially incorrect items multiple times and selecting the best item may not allowed in a practical scenario. Therefore, Prospector stops proceeding before clicking on the "buy" button, and tries to browse other items to find the best items. Also, in ALFWorld, Prospector does not use the final reward. It would be useful that a LLM agent can restart solving the task from the beginning, when it estimates the current trajectories is not promising.

3 Experiments

3.1 ALFWorld

ALFWorld (Shridhar et al., 2021) is a multi-modal interactive decision-making benchmark that is specialized on embodied reasoning tasks such as solving house-holding tasks. In this paper, we perform the experiments in the text-mode of ALFWorld where a natural language instruction is given and the agent is requested to generate text-based actions by interacting the environment. We evaluate LLM agents on the unseen 134 tasks in the ALFRED dataset. For fine-tuning open-sourced LLMs for Trajectory Ranking, we use 3K training tasks in the ALFRED dataset. The details on ALFWorld can found in the Appendix A.1.

3.1.1 Success rate

Comparison. In Table 1, we compare the success rate of Prospector with the recent LLM agents such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023) on ALFWorld. To show the difficulty of the tasks of ALFWorld, we also provide the performance of BUTLER, an agent that does not use LLMs.

The comparison on `text-davinci-002` as the LLM Actor is shown in the middle of the table. Prospector (AckAct + TR) outperforms ReAct and Reflexion. ReAct only uses few-shot demonstrations for solving house-holding tasks on ALFWorld. Reflexion further improves the success rate of the ReAct agent by using iterative refinements of the LLM output. Here, $k = 5$ refinements are performed. For the purpose of comparison, we use ReAct as the base LLM Actor. In Trajectory Ranking (TR), the LLM Actor of Prospector generates diverse trajectories and the LLM Critic selects the expected best trajectory. Here, $k = 5$ trajectories are generated, and TR selects the expected best trajectory by using the 2-shot LLM Critic.

The comparison on Llama-2-70B as the LLM Actor is shown in the bottom of the table. We conduct experiments with four different settings: (1) ReAct only, (2) AskAct only, (3) ReAct + Trajectory Ranking (TR), and (4) AskAct + TR. AskAct effectively improves the success rate of ReAct (from 41.0 to 56.7).

Experiment results on the task-level success rate can be found in Table 7 in the Appendix.

Effect of the number of trials. In Table 2, we show the change in the success rate with reward to

Method	LLM Actor	LLM Critic	Success Rate (%)
BUTLER	-	-	37.0
Act	PaLM-540B	-	45.0
ReAct	PaLM-540B	-	70.9
ReAct	text-davinci-002	-	77.6
ReAct + Reflexion ($k = 5$)	text-davinci-002	-	86.0
ReAct + **TR** ($k = 5$)	text-davinci-002	text-davinci-002	91.0
AskAct	text-davinci-002	-	78.4
AskAct + TR ($k = 5$) **[Prospector]**	text-davinci-002	text-davinci-002	**91.0**
ReAct	Llama-2-70B	-	41.0
ReAct + **TR** ($k = 5$)	Llama-2-70B	FLAN-T5-3B (SFT)	77.6
AskAct	Llama-2-70B	-	56.7
AskAct + TR ($k = 5$) **[Prospector]**	Llama-2-70B	FLAN-T5-3B (SFT)	86.6

Table 1: Performance comparison of LLM agents on ALFWorld. Prospector with AskAct and Trajectory Ranking (TR) considerably improves the success rate on ALFWorld, compared to the recent advancements such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023)

Method	LLM Actor	LLM Critic	$k = 1$	2	3	4	5
ReAct + **TR** ($k = 5$)	Llama-2-70B	FLAN-T5-3B (SFT)	41.0	59.0	69.4	73.1	77.6
AskAct + TR ($k = 5$) **[Prospector]**	Llama-2-70B	FLAN-T5-3B (SFT)	56.7	76.1	80.6	84.3	86.6

Table 2: Success rate with regard to the number of trajectories.

LLM Critic	Param.	Adaptation Method	# Trainable Param.	Accuracy (success/fail)
text-davinci-003	-	2-shot ICL	0	95.5
text-davinci-002	-	2-shot ICL	0	**97.0**
Bloom	7.1B	LoRA FT on 3K data	3.9M	79.1
Llama-2-Chat	7B	LoRA FT on 3K data	4.2M	94.8
Bloomz	7.1B	LoRA FT on 3K data	3.9M	95.5
Llama-2	7B	LoRA FT on 3K data	4.2M	96.3
GPT-J	6B	LoRA FT on 3K data	3.7M	97.3
T5	3B	LoRA FT on 3K data	5.9M	98.5
FLAN-T5	3B	LoRA FT on 3K data	4.7M	**98.5**

Table 3: Fine-tuning reward prediction accuracy of LLM Critics on ALFWorld.

the number of generated trajectories. As shown in the figure, the success rate of Prospector (AskAct + TR) increases as the number of generated trajectories (k) increases. To generate diverse (creative) trajectories, the LLM Actor of Prospector sets the temperature to 0.8. For Trajectory Ranking (TR), 2-shot LLM Critic (`text-davinci-002`) is used, and its temperature is set to 0.2. Since AskAct provides a better baseline, AskAct + TR can achieve much better performance with less sampling (e.g., AskAct only (56.7) comparable with ReAct + TR (k=2) (59.0)). We emphasize that AskAct and TR can make an effective synergy in improving LLM agents in terms of both performance and efficiency.

3.1.2 Accuracy of LLM Critic

Fine-tuned reward model accuracy. In Table 3, we show the fine-tuning reward prediction accuracy of LLM Critics on ALFWorld. We finetune open-sourced LLMs on 3K ALFWorld trajectory data. For decoder-only models, we choose GPT-J (Wang and Komatsuzaki, 2021), Bloom (Scao et al., 2022), Bloomz (Muennighoff et al., 2023), and Llama-2 (Touvron et al., 2023). For encoder-decoder models, we choose T5 (Raffel et al., 2020) and FLAN-T5 (Chung et al., 2022). For parameter-efficient fine-tuning, we use LoRA (Hu et al., 2022). By fine-tuning open-sourced LLMs on 3K ALFWorld trajectory data, they can achieve comparable or better reward prediction accuracy with the closed LLMs such as `text-davinci-002`. The hyperparameters used for fine-tuning LLM Critics can be found in Table 11 in the Appendix.

Method	LLM Actor	LLM Critic	Reward	Success Rate (%)
Human (expert)	-	-	82.1	59.6
Human (average)	-	-	75.5	50.0
IL	-	-	59.9	29.1
IL + RL	-	-	62.4	28.7
Small LLM	Llama-2-7B-Chat	-	17.9	-
Small LLM + SFT	Llama-2-7B-Chat	-	63.1	-
Small LLM + RLHF (PPO)	Llama-2-7B-Chat	-	63.6	-
Small LLM + ETO	Llama-2-7B-Chat	-	67.4	-
ReAct	text-davinci-002	-	63.3	35.8
ReAct + Reflexion ($k = 8$)	text-davinci-002	-	-	35.0
AskAct	text-davinci-002	-	66.5	39.8
AskAct + TR ($k = 8$)	text-davinci-002	text-davinci-002	69.3	41.4
AskAct + TR ($k = 8$) **[Prospector]**	text-davinci-002	Llama-2-7B-Chat (SFT)	**70.8**	43.0
ReAct	Llama-2-70B	-	62.3	37.6
ReAct + **TR** ($k = 8$)	Llama-2-70B	FLAN-T5-3B (SFT)	69.3	42.2
AskAct	Llama-2-70B	-	68.6	42.2
AskAct + TR ($k = 8$) **[Prospector]**	Llama-2-70B	FLAN-T5-3B (SFT)	70.2	**43.6**

Table 4: Performance comparison of LLM agents on WebShop. Prospector with AskAct and Trajectory Ranking (TR) can improves the success rate on WebShop, compared to the recent advancements such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023). The rewards of fine-tuning-based methods such as SFT, PPO, and ETO are referenced from the ETO (Song et al., 2024) paper.

LLM Critic	Param.	Adaptation Method	# Trainable Param.	Accuracy (hi/mi/lo)
text-davinci-003	-	2-shot ICL	0	42.2
text-davinci-002	-	2-shot ICL	0	47.0
Bloom	7.1B	LoRA FT on 12K data	3.9M	67.2
GPT-J	6B	LoRA FT on 12K data	3.7M	72.0
Llama-2	7B	LoRA FT on 12K data	4.2M	73.8
Bloomz	7.1B	LoRA FT on 12K data	3.9M	75.8
Llama-2-Chat	7B	LoRA FT on 12K data	4.2M	76.2
T5	3B	LoRA FT on 12K data	5.9M	77.0
FLAN-T5	3B	LoRA FT on 12K data	4.7M	**78.0**

Table 5: Fine-tuning reward prediction accuracy of LLM Critics on WebShop. Fine-tuned LLM Critics (e.g., Llama-2 fine-tuned on 12K trajectory data) provide significantly improved reward prediction accuracy, compared to few-shot LLM Critics (e.g., `text-davinci-002`) in WebShop. Improved prediction accuracy of fine-tuned LLM Critics help to increase the success rate of Prospector

3.2 WebShop

WebShop (Yao et al., 2022) is a large-scale online shopping environment with more than 1M real-world products crawled from Amazon. WebShop provides about 12K human instructions, and reserves 500 instructions for tasting. In this paper, we perform experiments in the text-based mode and evaluate LLM agents on the official 500 test instructions. We use 12K human instructions (without test instruction) for generating trajectories and fine-tuning LLM Critics on them. The details on WebShop can be found in Applendix A.2.

3.2.1 Success rate

In Table 4, we compare the performance of Prospector with the recent LLM agents such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023). In addition to these methods, to assess the difficulty of the environment, we also provide the performance of human as upper bound, and the performance of the traditional methods that use Imitation Learning (IL) or Reinforcement Learning (RL) as strong baselines. These results are quoted from the WebShop (Yao et al., 2022) paper. As shown in the table, Prospector achieves better success rate (43.0%) than the recent advancements such as ReAct (35.8%) and Reflexion(35.0%). Compared to the traditional IL (29.1%) and RL (28.7%) methods, ReAct agents based on `text-davinci-002` surprisingly achieve high success rate without training. However, there is a gap with the human performance (50%). Prospector can considerably reduce this gap by using Self-Asking and Trajectory Ranking (TR). Prospector using AskAct prompting can increase the success rate up to 39.8% compared to ReAct (35.8%). Prospector with AskAct and TR further increase the success rate up to 41.3%. However, since the few-shot LLM Critic based on `text-davinci-002` does not provide high accuracy (47.0%) in reward prediction, the improvement is not significant. In contrast, since the fine-tuned LLM Critic based on Llama-2-7B-Chat (Touvron et al., 2023) provides much higher accuracy in reward prediction, Prospector can achieve better success rate (43.0%). Note that if the oracle with known reward is used, the success rate can be reached by up to 47.0%, while considerably closing the gap with the human performance (50%). Note that the performance of LLM Critic is important to improve the performance of LLM agents. Regarding this, we provide the detailed additional experiments on LLM Critics in Table 9, Table 5, and Table 10 in the following subsection.

On the WebShop environment, Llama-2-70B, one of representative open-source LLMs can achieve comparable performance with text-davinci-002, one of the most powerful LLMs. In both cases of text-davinci-002 and Llama-2-70B, AskAct meaningfully improves the success rate compared to ReAct: from 35.8 to 39.8 on text-davinci-002, and from 37.6 to 42.2 on Llama-2-70B. This means that AskAct, a simple prompting method that adds extra question prompts on ReAct, can be effective. ReAct + TR can improve ReAct from 37.6 to 42.2 in the success rate. AskAct + TR further improves the success rate of AskAct (from 42.2 to 43.6), and provides better performance than ReAct + TR (42.2).

3.2.2 Accuracy of LLM Critic

Fine-tuned reward model accuracy. In Table 5, we compare the reward prediction accuracy of fine-tuned LLM Critics. We finetune open-sourced LLMs on 3K ALFWorld trajectory data. For parameter-efficient fine-tuning, we use LoRA (Hu et al., 2022). Fine-tuned LLM Critics (e.g., Llama-2 fine-tuned on 12K trajectory data) provide significantly improved reward prediction accuracy, compared to few-shot LLM Critics (e.g., `text-davinci-002`) in the WebShop environment. Improved prediction accuracy of fine-tuned LLM Critics help to increase the success rate of Prospector.

4 Related Work

Reasoning in LLMs. Wei et al. (2022) introduced chain-of-thoughts (CoT) that generates a series of short sentences that mimic the human reasoning process. CoT with Self-Consistency (CoT-SC) (Wang et al., 2023) samples k diverse reasoning paths instead of selecting the greedy one and subsequently returns the most frequent answer. On the other hand, Self-Ask (Press et al., 2022) improves CoT on QA tasks by transforming a chain-of-thought into a multi-turn self-question-answering process. This study also introduced the concept of conducting reasoning through question-answering concurrently with our work, but we want to emphasize that while Self-Ask focuses on QA tasks, our work enhances LLM agents for interactive decision-making tasks through synergizing self-asking and trajectory ranking.

LLM-based agents. ReAct (Yao et al., 2023) is an algorithm that integrates reasoning and action within language models to tackle a diverse range of language reasoning and decision-making tasks. When task feedback is accessible, Reflexion (Shinn et al., 2023) and Self-Refine (Madaan et al., 2023) reinforce LLM agents by learning a linguistic reward model to verbally reflect the task feedback signal. We note that Reflexion iteratively generates trajectories and reflects rewards verbally in sequence, while Prospector generates diverse trajectories in parallel and chooses the best one in terms of rewards.

Reward models and rankings. Reward models and rankings are widely employed within the LLM context and their applications. In order to enhance LLM performance, InstructGPT (Ouyang et al., 2022) and Llama-2 (Touvron et al., 2023) leverage RL for fine-tuning the LLMs themselves. Furthermore, LLMs have showcased their impressive capability to generate code across diverse programming tasks, highlighting their versatility. Within this domain, a neural ranker, CodeRanker (Inala et al., 2022), was introduced to improve the accuracy of various code generation models, enabling them to predict the correctness of sampled code without actual execution. On the other hands, to harness the LLM's semantic knowledge about the real world, SayCan (Brohan et al., 2023) proposed an innovative approach to combine LLM and RL.

5 Conclusion

We introduce Prospector, a LLM agent that consists two complementary LLM agents, LLM Actor and LLM Critic. To improve the baseline performance of LLM Actor, we provide AskAct prompting that introduces additional self-asking steps in the few-shot demonstrations. Also, to implicitly reflect environment feedback, we provide Trajectory Ranking that selects the most rewarding trajectory by predicting the expected total reward. Evaluation on representative decision-making benchmark environments such as ALFWorld and WebShop shows that Prospector can considerably increase the success rate of given tasks, while outperforming recent ICL-based agents such as ReAct and Reflexion.

Limitations

We empirically show that Prospector can considerably improves the success rate of interactive decision-making tasks by using AskAct prompting and Trajectory Ranking. There may be some limitations, since it requires additional computations than ReAct (Yao et al., 2023). Recent advancements such as Reflexion (Shinn et al., 2023) also requires additional inference cost for performing iterative refinements. Also, we believe that investigating efficient LLM agents is one of promising research topics.

Ethics Statement

This paper introduces Prospector, a LLM agent that consists of two complementary LLMs auch as LLM Actor and LLM Critic. Prospector can improve LLM-based autonomous agents that can solve interactive decision-making tasks such as ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022). While performing this research, we comply with the Ethics Policy. Also, we hope that research on LLM agents such as Prospector can enhance our workplaces and enrich our daily lives, since it can increase the productivity and creativity of our work and life.

References

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *CoRL*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023a. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*.

Xiaoyu Chen, Shenao Zhang, Pushi Zhang, Li Zhao, and Jianyu Chen. 2023b. Asking before action: Gather information in embodied decision making with language models. *arXiv:2305.15695*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv:2210.11416*.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2018. Textworld: A learning environment for text-based games. In *IJCAI Computer Games Workshop*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Jeevana Priya Inala, Chenglong Wang, Mei Yang, Andres Codas, Mark Encarnación, Shuvendu Lahiri, Madanlal Musuvathi, and Jianfeng Gao. 2022. Fault-aware neural code rankers. In *NeurIPS*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. In *NeurIPS*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *NeurIPS*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2023. Crosslingual generalization through multitask finetuning. In *ACL*.

Vinod Muthusamy, Yara Rizk, Kiran Kate, Praveen Venkateswaran, Vatche Isahagian, Ashu Gulati, and Parijat Dube. 2023. Towards large language model-based personal agents in the enterprise: Current trends and open problems. In *EMNLP*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv:2211.05100*.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. In *EMNLP*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. In *NeurIPS*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *ICLR*.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for llm agents. In *ACL*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *EMNLP*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

A Benchmark Details

A.1 ALFWorld

ALFWorld (Shridhar et al., 2021) is a multi-modal interactive decision-making benchmark that is specialized on embodied reasoning tasks such as solving house-holding tasks. It is designed by aligning TextWorld (Côté et al., 2018), an interactive text-based game, and ALFRED (Shridhar et al., 2020), a representative embodied AI benchmark. It includes 6 types of tasks such as (1) pick and place, (2) examine in light, (3) clean and place, (4) heat and place, (5) cool and place, and (6) pic two and place. The ALFRED dataset provides 3,553 tasks for training, 140 tasks for seen testing, and 134 tasks for unseen testing. In this paper, we perform the experiments in the text-mode of ALFWorld where a natural language instruction is given and the agent is requested to generate text-based actions by interacting the environment. We evaluate LLM agents on the unseen 134 tasks in the ALFRED dataset. For fine-tuning open-sourced LLMs for Trajectory Ranking, we use 3K training tasks in the ALFRED dataset.

A.2 WebShop

WebShop (Yao et al., 2022) is a large-scale online shopping environment with more than 1M real-world products crawled from Amazon. The agent is given a natural language instruction (e.g., "I would like 3 ounce bottle of bright citrus deodorant for sensitive skin, and price lower than 50.00 dollars."), and required to make a sequence of actions (e.g., querying the search engine with keywords and clicking on a product title) to accomplish the given instruction. More specifically, the task mainly consists of five stages: (1) searching products with query words, (2) selecting a product in the search results, (3) selecting proper options, (4) reviewing the product details, and (5) clicking on the "Buy Now" button. WebShop provides two modes: (1) multi-modal mode with product figures, and (2) text-based mode. Also, WebShop provides about 12K human instructions, and reserves 500 instructions for tasting. In this paper, we perform experiments in the text-based mode and evaluate LLM agents on the official 500 test instructions. We use 12K human instructions (without test instruction) for generating trajectories and fine-tuning LLM Critics on them.

B Additional Experiments

ALFWorld

```
Evaluate if the instruction given in the input
is accomplished by performing a sequence
of actions (fail/success).

### Input:
{Example success trajectory}
### Response: success

### Input:
{Example fail trajectory}
### Response: fail

### Input:
{Input trajectory}
### Response:
```

WebShop

```
Evaluate if the instruction given in the input
is accomplished by selecting the proper item
(low/middle/high).

### Input:
{Example high-reward trajectory}
### Response: high

### Input:
{Example low-reward trajectory}
### Response: low

### Input:
{Input trajectory}
### Response:
```

Table 6: Critic prompt template for few-shot reward prediction.

B.1 ALFWorld

Task-level success rate. In Table 7, we provide the detailed success rate for each task type in the ALFWorld benchmark.

Few-shot reward model accuracy. In Table 8, we show the few-shot reward prediction accuracy of LLM Critics on ALFWorld. The few-shot accuracy is high enough to be used in Trajectory Ranking without the need for find-tuning open-sourced LLMs on AFLWorld trajectory data. Since the reward prediction accuracy of 2-shot LLM Critic is very high (97%), the LLM Critic of Prospector can select the highly-rewarding trajectory from diverse trajectories and considerably increase the success rate.

B.2 WebShop

Few-shot reward model accuracy. In Table 9, we provide the few-shot reward prediction ac-

Method	LLM Actor	LLM Critic	Pick	Clean	Heat	Cool	Look	Pick2	All (%)
BUTLER	-	-	46	39	74	100	22	24	37.0
Act	PaLM-540B	-	88	42	74	67	72	41	45.0
ReAct	PaLM-540B	-	92	58	96	86	78	41	70.9
ReAct	text-davinci-002	-	88	61	78	86	89	71	77.6
	(#success/#tasks)	-	21/24	19/31	18/23	18/21	16/18	12/17	104/134
ReAct + Reflexion ($k = 5$)	text-davinci-002	-	-	-	-	-	-	-	86.0
ReAct + **TR** ($k = 5$)	text-davinci-002	text-davinci-002	**100**	84	91	**95**	**100**	**76**	**91.0**
			24/24	26/31	21/23	20/21	18/18	13/17	122/134
ReAct	Llama-2-70B	-	42	26	61	62	22	35	41.0
			10/24	8/31	14/23	13/21	4/18	6/17	55/134
ReAct + **TR** ($k = 5$)	Llama-2-70B	FLAN-T5-3B (SFT)	92	74	91	86	61	53	77.6
			22/24	23/31	21/23	18/21	11/18	9/17	104/134
AskAct	Llama-2-70B	-	58	54	43	81	72	29	56.7
			14/24	17/31	10/23	17/21	13/18	5/17	76/134
AskAct + TR ($k = 5$)	Llama-2-70B	FLAN-T5-3B (SFT)	92	**87**	**96**	**95**	94	47	86.6
			22/24	27/31	22/23	20/21	17/18	8/17	116/134

Table 7: Performance comparison of LLM agents on ALFWorld. Prospector with AskAct and Trajectory Ranking (TR) considerably improves the success rate on ALFWorld, compared to the recent advancements such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023)

LLM Critic	1-shot	2-shot	3-shot
text-davinci-002	94.8	**97.0**	95.5
text-davinci-003	93.3	95.5	94.0

Table 8: Few-shot reward prediction accuracy of LLM Critics on ALFWorld.

curacy of API-based LLM Critics such as `text-davinci-002` on WebShop. We find that few-shot LLM Critics have some difficulty in predicting the reward of a given trajectory in a complex environment such as WebShop. LLM Critics with low reward prediction accuracy can not be used for reliable Trajectory Ranking (TR). This result requires us to fine-tune open-sourced LLMs such as Llama-2 on WebShop trajectory data.

LLM Critic	1-shot	2-shot	3-shot
text-davinci-002	34.4	**47.0**	42.4
text-davinci-003	37.0	42.2	36.2

Table 9: Few-shot reward prediction accuracy of LLM Critics on WebShop. Few-shot LLM Critics have some difficulty in predicting the reward of the agent's trajectory in a complex environment such as WebShop. This requires LLM Critics fine-tuned on WebShop trajectory data.

Fine-tuned reward model accuracy. In Table 10, we provide the change in reward prediction accuracy with regard to the size of trajectory data. We can see that the reward prediction accuracy increases as the data size increases. The hyperparameters used for fine-tuning LLM Critics can be found in Table 11 in the Appendix.

LLM Critic	3K	6K	9K	12K
Llama-2-7B-Chat (LoRA)	70.0	71.1	76.2	**76.2**

Table 10: Fine-tuning accuracy over the dataset size.

C Hyper-parameters of LLM Critic fine-tuning

In Table 11, we provide the hyper-parameters used for fine-tuning the LLM Critic on the trajectory data.

Hyper-parameter	Value
model max length (context length)	1024
batch size	128
max epochs	50
optimizer	AdamW
learning rate	3e-4
weight decay	0.1
learning rate scheduler	cosine
warm-up steps	50% of the max steps
LoRA r	8
LoRA alpha	32
LoRA drop-out	0.1

Table 11: Hyper-parameters of LLM Critic fine-tuning.

D Discussion

Comparison of AskAct and ReAct. AskAct is a prompting method that interleaves additional self-asking steps in a ReAct (Yao et al., 2023) prompt. In ALFWorld, the self-asking step checks if a target object is found. This can elicit a correct action by alleviating hallucination. In WebShop, the self-asking step explicitly tries to determine which item is the most proper. This can elicit a better item selection. We empirically show that AskAct considerably improves the success rate compared to ReAct (see Table 2 and Table 5). In ALFWorld (Shridhar et al., 2021), AskAct on Llama-2-70B provides 56.7% of success rate, while achieving about 15.0% absolute improvement compared to ReAct on Llama-2-70B (41.0% of success rate). In WebShop (Yao et al., 2022), AskAct on Llama-2-70B achieves 42.2% of success rate (about 4.6% improvement), while ReAct on Llama-2-70B provides 37.6% of success rate.

Comparison of AskAct and Self-Ask. AskAct and Self-Ack (Press et al., 2022) commonly have self-asking steps in few-shot examples in a prompt to improve the LLM response. However, AskAct and Self-Ask are significantly different in the purpose and composition. AskAct is mainly designed to solve sequential decision-making tasks, and consists of a sequence of observation, self-question, self-answer, self-reasoning, and action. In contrast, Self-Ask is mainly designed to provide better answers to knowledge-intensive questions, and consist of question, follow-up question, follow-up answer, and answer. AskAct (adding extra question and answer steps in a ReAct prompt) and Self-Ask (adding extra question and answer steps in a CoT prompt) can be seen as having a relationship similar to ReAct (adding intermediate reasoning steps in an Act prompt) and Chain-of-Thoughts (adding intermediate reasoning steps in a direct prompt).

Comparison of Trajectory Ranking (TR) and Reflexion. In Prospector, the LLM agent can generate multiple trajectories in a parallel manner. Given multiple trajectories, Trajectory Ranking (TR) predicts the expected reward of each trajectory and selects a trajectory which expected reward is the maximum. In contrast, Reflexion (Shinn et al., 2023) iteratively refines a trajectory generated by the LLM agent. Since the trajectory improvement is done in a sequential manner, exploration ability is significantly limited. Furthermore, for trajectory improvement, Reflexion requires carefully-crafted verbal feedback prompts. Due to these limitations, Reflexion does not improve the success rate in WebShop.

Synergy of combining AskAct and Trajectory Ranking. Finally, we would like to emphasize that AskAct and TR can make an effective synergy in improving LLM agents in terms of both performance and efficiency. Since AskAct provides a better baseline, AskAct and TR can achieve much better performance with less sampling. For example, in ALFWorld, AskAct only achieves 56.7% of success rate, while ReAct and TR (k=2) reaches 56.0% of success rate.

E Prompts

E.1 Example AskAct prompt on ALFWorld

I this section, we provide the prompts used for the LLM Actor and few-shot LLM Critic to solve the tasks in the ALFWorld. The ReAct prompt for the LLM Actor is provided in Table 12. The Critic prompt for the LLM Critic is provided in Table 13.

E.2 Example AskAct prompt on WebShop

I this section, we provide the prompts used for the LLM Actor and few-shot LLM Critic to solve the decision-making tasks in the WebShop. The AskAct prompt for the LLM Actor is provided in Table 14. The Critic prompt for the few-shot LLM Critic is provided in Table 15.

F Trajectories

F.1 Example trajectory on ALFWorld

In Table 16, we provide an example success trajectory generated by Prospector on ALFWorld.

F.2 Example trajectory on WebShop

In Table 17, we provide an example high-reward trajectory generated by Prosepctor on WebShop.

AskAct Prompt for ALFWorld
`` You are in the middle of a room. Looking quickly around you, you see a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a countertop 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a sinkbasin 2, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1. ```

```
You are in the middle of a room. Looking quickly around you,
you see a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1,
a countertop 1, a garbagecan 1, a handtowelholder 2,
a handtowelholder 1, a sinkbasin 2, a sinkbasin 1, a toilet 1,
a toiletpaperhanger 1, and a towelholder 1.

Your task is to: put some spraybottle on toilet.

> think: To solve the task, I need to find and take a sparybottle,
then put it on toilet.
OK.
> think: First I need to find a spraybottle. A spraybottle is
more likely to appear in cabinet (1-4), countertop (1), toilet (1),
sinkbasin (1-2), garbagecan (1). I can check one by one,
starting with cabinet 1.
OK.
> go to cabinet 1
On the cabinet 1, you see a cloth 1, a soapbar 1, a soapbottle 1.
> think: Did I find a spraybottle?
OK.
> think: No, I did not. I need to go other place to find a spraybottle.
OK.
> go to cabinet 2
The cabinet 2 is closed.
> open cabinet 2
You open the cabinet 2. The cabinet 2 is open. In it,
you see a candle 1, and a spraybottle 2.
> think: Did I find a spraybottle?
OK.
> think: Now I find a spraybottle (2). Next, I need to take it.
OK.
> take spraybottle 2 from cabinet 2
You pick up the spraybottle 2 from the cabinet 2.
> think: Now I take a spraybottle (2). Next, I need to
put it in/on toilet 1.
OK.
> go to toilet 1
On the toilet 1, you see a soapbottle 2.
> put spraybottle 2 in/on toilet 1
You put the spraybottle 2 in/on the toilet 1.
```

Table 12: Example AskAct prompt of LLM Actor on ALFWorld. Reasoning steps are denoted in blue. Additional self-asking steps are denoted in red.

Critic Prompt on ALFWorld

```
Evaluate if the instruction given in the input is accomplished by
performing a sequence of actions (fail/success).

### Input:
You are in the middle of a room. Looking quickly around you,
you see a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3,
a countertop 2, a countertop 1, a diningtable 1, a drawer 6,
a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1,
a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1,
a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1,
and a toaster 1.

Your task is to: clean some lettuce and put it in fridge.

> think: To solve the task, I need to find and take a lettuce,
then clean it with sinkbasin, then put it in fridge.
OK.
> think: First I need to find a lettuce. A lettuce is more likely
to appear in fridge (1), diningtable (1), sinkbasin (1),
stoveburner (1-3), cabinet (1-2). I can check one by one,
starting with fridge 1.
OK.
> go to fridge 1
The fridge 1 is closed.
> open fridge 1
You open the fridge 1. The fridge 1 is open. In it, you see a apple 1,
a cup 1, a lettuce 1, a potato 2, and a potato 1.
> think: Now I find a lettuce (1). Next, I need to take it.
OK.
> take lettuce 1 from fridge 1
You pick up the lettuce 1 from the fridge 1.
> think: Now I take a lettuce (1). Next, I need to go to
sinkbasin (1) and clean it.
OK.
> go to sinkbasin 1
On the sinkbasin 1, you see a apple 2, a butterknife 1, a cup 2,
a knife 2, a potato 3, and a spoon 3.
> clean lettuce 1 with sinkbasin 1
You clean the lettuce 1 using the sinkbasin 1.
> think: Now I clean a lettuce (1). Next, I need to put it
in/on fridge 1.
OK.
> go to fridge 1
The fridge 1 is open. In it, you see a apple 1, a cup 1, a potato 2,
and a potato 1.
> put lettuce 1 in/on fridge 1
You put the lettuce 1 in/on the fridge 1.
### Response: success

### Input:
{Input trajectory}
### Response:
```

Table 13: Critic prompt of few-shot LLM Critic on ALFWorld. The *success* trajectory given as a 1-shot demonstration is generated by `text-davinci-002` with ReAct prompting. Instructions for few-shot reward prediction are denoted in red.

AskAct Prompt for WebShop
Webshop Instruction: i would like a 3 ounce bottle of bright citrus deodorant for sensitive skin, and price lower than 50.00 dollars [Search] Action: search[3 ounce bright citrus deodorant sensitive skin] Observation: [Back to Search] Page 1 (Total results: 50) [Next >] [B078GWRC1J] Bright Citrus Deodorant by Earth Mama | Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce $10.99 [B078GTKVXY] Ginger Fresh Deodorant by Earth Mama | Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce $10.99 [B08KBVJ4XN] Barrel and Oak - Aluminum-Free Deodorant, Deodorant for Men, Essential Oil-Based Scent, 24-Hour Odor Protection, Cedar & Patchouli Blend, Gentle on Sensitive Skin (Mountain Sage, 2.7 oz, 2-Pack) $15.95 Action: think[Given the instruction and candidate times such as B078GWRC1J, B078GTKVXY, and B08KBVJ4XN, which item is the most proper to select?] Observation: OK. Action: think[B078GWRC1J is the most proper, since the item is bright citrus deodorant less then 50 dollars.] Observation: OK. Action: click[B078GWRC1J] Observation: [Back to Search] [< Prev] scent [assorted scents][bright citrus][calming lavender] [ginger fresh][simply non-scents] size [travel set (4-pack)][3 ounce (pack of 1)][3-ounce (2-pack)] Bright Citrus Deodorant by Earth Mama | Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce Price: $10.99 Rating: N.A. [Description] [Features] [Reviews] [Buy Now] Action: think[For 3 ounce bottle of bright citrus deodorant for sensitive skin, the item has options 'bright citrus' and '3 ounce (pack of 1)' and seems good to buy.] Observation: OK. Action: click[bright citrus] Observation: You have clicked bright citrus. Action: click[3 ounce (pack of 1)] Observation: You have clicked 3 ounce (pack of 1). Action: click[Buy Now]

Table 14: Example AskAct prompt of LLM Actor on WebShop. Reasoning steps are denoted in blue. Additional self-asking steps are denoted in red.

Critic Prompt on WebShop

```
Evaluate if the instruction given in the input is accomplished
by selecting the proper item (low/middle/high).

### Input:
WebShop
Instruction:
find me a regular fit machine washable cargo pants with buttoned closure
in 6057 apricot color and 29 size, and price lower than 40.00 dollars
[Search]

Action: search[regular fit machine washable cargo pants buttoned closure
6057 apricot color 29 size]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B00PQFK67G]
Match Men's Athletic-Fit Cargo Pants
$21.99 to $29.99
[B01M4KXXPT]
Vertx Men's Phantom Lt 2.0 Shorts
$21.14 to $76.84
[B01EAP5JD8]
DUNLOP Mens On Site Workwear Trousers Regular Classic
$63.99

Action: think[B00PQFK67G is the one I want.]
Observation: OK.

Action: click[B00PQFK67G]
Observation:
[Back to Search]
[< Prev]
size [29][30][32][34][36][38][40][42][44]
color [6057 apricot][6057 khaki][6062 dark khaki][6531 apricot][6531 black]
 [6531 grayish black][6531 light gray][6531 light khaki][khaki max]
 [6057 light apricot][6069 blue][6531 army green][6531 blue][6531 dark gray]
 [6071 black][6071 dark blue][6071 light khaki][6039m tangerine max]
 [6071 dark apricot][6071 dark khaki][6057 bluish purple][6057 dark gray]
 [6071 brown][6071 gray][6039m multi-max][6039 light khaki][6071 steel gray]
 [6039 dark yellow][6039 blue][6039 apricot][6039 dark gray][6050 peacock blue]
 [6057 silver gray][6066 light apricot]
Match Men's Athletic-Fit Cargo Pants
Price: $21.99 to $29.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]

Action: think[The item has options '29', '6057 apricot', '6057 khaki',
'6062 dark khaki', '6531 apricot', '6531 black', '6531 grayish black',
'6531 light gray', '6531 light khaki', 'khaki max', '6057 light apricot',
'6069 blue', '6531 army green', '6531 blue', '6531 dark gray', '6071 black']
Observation: OK.

Action: click[29]
Observation: You have clicked 29.

Action: click[6057 apricot]
Observation: You have clicked 6057 apricot.

Action: click[Buy Now]
### Response: high

### Input:
{Input trajectory}
### Response:
```

Table 15: Critic prompt of few-shot LLM Critic on WebShop. The *high-reward* trajectory given as a 1-shot demonstration is generated by `text-davinci-002` with ReAct prompting. Instructions for few-shot reward prediction are denoted in red.

Success Trajectory on ALFWorld
You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a shelf 3, a shelf 2, a shelf 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1. Your task is to: put a clean mug in coffeemachine.

```
> think: To solve the task, I need to find and take a mug,
hen clean it with sinkbasin, then put it in coffeemachine.
OK.
> think: First I need to find a mug. A mug is more likely
to appear in fridge (1), diningtable (1), sinkbasin (1), stoveburner (1-3),
cabinet (1-6). I can check one by one, starting with fridge 1.
OK. > go to fridge 1
The fridge 1 is closed.
> open fridge 1
You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a cup 2,
a cup 1, a lettuce 1, a plate 1, a tomato 2, and a tomato 1.
> think: Did I find a mug?
OK.
> think: No, I did not. I need to go other place to find a mug.
OK.
> go to diningtable 1
Nothing happens.
> go to sinkbasin 1
On the sinkbasin 1, you see a cup 3, a egg 3, and a glassbottle 3.
> think: Did I find a mug?
OK.
> think: No, I did not. I need to go other place to find a mug.
OK.
> go to stoveburner 1
On the stoveburner 1, you see a pot 1.
> think: Did I find a mug?
OK.
> think: No, I did not. I need to go other place to find a mug.
OK.
> go to cabinet 1
On the cabinet 1, you see a dishsponge 1, and a soapbottle 2.
> think: Did I find a mug?
OK.
> think: No, I did not. I need to go other place to find a mug.
OK.
> go to cabinet 2
The cabinet 2 is closed.
> open cabinet 2
You open the cabinet 2. The cabinet 2 is open. In it, you see a peppershaker 1.
> think: Did I find a mug?
OK.
> think: No, I did not. I need to go other place to find a mug.
OK.
> go to cabinet 3
On the cabinet 3, you see a mug 2, a mug 1, and a soapbottle 1.
> think: Did I find a mug?
OK.
> think: Now I find a mug (2). Next, I need to take it.
OK.
> take mug 2 from cabinet 3
You pick up the mug 2 from the cabinet 3.
> think: Now I take a mug (2). Next, I need to go to sinkbasin (1) and clean it.
OK.
> go to sinkbasin 1
On the sinkbasin 1, you see a cup 3, a egg 3, and a glassbottle 3.
> clean mug 2 with sinkbasin 1
You clean the mug 2 using the sinkbasin 1.
> think: Now I clean a mug (2). Next, I need to put it in/on coffeemachine 1.
OK.
> go to coffeemachine 1
On the coffeemachine 1, you see nothing.
> put mug 2 in/on coffeemachine 1
You put the mug 2 in/on the coffeemachine 1.
```

Table 16: Success trajectory generated by Prospector on ALFWorld. Reasoning steps are denoted in blue. Additional self-asking steps are denoted in red.

High-reward trajectory on WebShop

```
WebShop
Instruction:
i am looking for mn4 color foundation for my sensitive skin,
and price lower than 40.00 dollars
[Search]

Action: search[mn4 color foundation sensitive skin]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B077PR9TL4]
Physicians Formula The Healthy Foundation with SPF 20, LW2, 1 Fl Oz
$7.59
[B07VLKYJNH]
PÜR 4-in-1 Love Your Selfie Longwear Foundation & Concealer - Full Coverage
Foundation & Concealer Makeup For Dark Spots, Blemishes and Imperfections -
Long-Lasting Liquid Makeup & Skincare
$36.0
[B09J1X9M1P]
Neutrogena Healthy Skin Sensitive Skin Serum Foundation with Pro-Vitamin B5,
Color Correcting & Pore Minimizing Liquid Foundation & Face Serum, Buildable
Coverage, Medium 02, 1 oz
$14.39

Action: think[Given the instruction and candidate times such as B077PR9TL4,
B07VLKYJNH, and B09J1X9M1P, which item is the most proper to select?]
Observation: OK.

Action: think[B077PR9TL4 is the most proper, since the item is mn4 color
foundation less then 40 dollars.]
Observation: OK.

Action: click[B077PR9TL4]
Observation:
[Back to Search]
[< Prev]
color [1][dc1][dn3][dn4][lc1][ln3][ln4][lw2][mc1][mn3][mn4][mw2]
size [1 fl oz (pack of 1)][1 ounce (pack of 1)][1 fluid ounce]
Physicians Formula The Healthy Foundation with SPF 20, LW2, 1 Fl Oz
Price: $7.59
Rating: N.A.
[Description]
[Features]
[Reviews]
[Attributes]
[Buy Now]

Action: think[For mn4 color foundation for sensitive skin, the item has
options 'mn4' and '1 fl oz (pack of 1)' and seems good to buy.]
Observation: OK.

Action: click[mn4]
Observation: You have clicked mn4.

Action: click[1 fl oz (pack of 1)]
Observation: You have clicked 1 fl oz (pack of 1).

Action: click[Buy Now]
Observation: Your score (min 0.0, max 1.0): 1.0
```

Table 17: High-reward trajectory generated by Prospector on WebShop. Reasoning steps are denoted in blue. Additional self-asking steps are denoted in red.