

Asking Before Action: Gather Information in Embodied Decision Making with Language Models

Xiaoyu Chen *
Tsinghua University

Shenao Zhang
Northwestern University

Pushi Zhang
Microsoft Research Asia

Li Zhao
Microsoft Research Asia

Jianyu Chen
Tsinghua University
Shanghai Qizhi Institute

Abstract

With strong capabilities of reasoning and a generic understanding of the world, Large Language Models (LLMs) have shown great potential in building versatile embodied decision making agents capable of performing diverse tasks. However, when deployed to unfamiliar environments, we show that LLM agents face challenges in efficiently gathering necessary information, leading to suboptimal performance. On the other hand, in unfamiliar scenarios, human individuals often seek additional information from their peers before taking action, leveraging external knowledge to avoid unnecessary trial and error. Building upon this intuition, we propose *Asking Before Action* (ABA), a method that empowers the agent to proactively query external sources for pertinent information using natural language during their interactions in the environment. In this way, the agent is able to enhance its efficiency and performance by mitigating wasteful steps and circumventing the difficulties associated with exploration in unfamiliar environments. We empirically evaluate our method on an embodied decision making benchmark, ALFWorld, and demonstrate that despite modest modifications in prompts, our method exceeds baseline LLM agents by more than 40%. Further experiments on two variants of ALFWorld illustrate that by imitation learning, ABA effectively retains and reuses queried and known information in subsequent tasks, mitigating the need for repetitive inquiries. Both qualitative and quantitative results exhibit remarkable performance on tasks that previous methods struggle to solve.

1 Introduction

Recent advances in large language models (LLMs) have exhibited remarkable abilities in language comprehension, text generation, question answering, dialogue, reasoning, and can even exhibit human-level performance on various benchmarks (Ouyang et al., 2022; Chowdhery et al., 2022; OpenAI, 2023; Google, 2023). Since LLMs have been trained on extensive and diverse text corpora, they have captured a broad range of commonsense understanding about the world, enabling them to adeptly handle a multitude of various and complex scenarios. Therefore, recently, researchers have proposed to integrate LLMs in embodied decision making (Huang et al., 2022a; Li et al., 2022; Ahn et al., 2022; Huang et al., 2022b; Singh et al., 2022a; Yao et al., 2022; Wang et al., 2023; Driess et al., 2023; Carta et al., 2023), either using LLMs to do task planning or directly using LLM as an agent.

Although language agents are capable of making sound decisions when provided with ample information, they face challenges when encountering situations with limited or insufficient information

*chen-xy21@mails.tsinghua.edu.cn

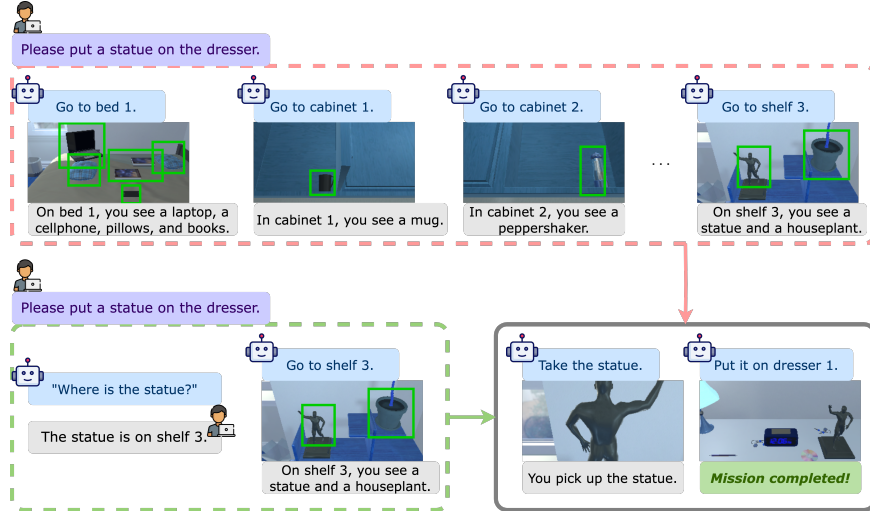


Figure 1: *Asking Before Action* (ABA) allows the agent to efficiently query for pertinent information from external sources via natural language and subsequently execute actions based on the acquired responses. Imagine the room owner instructs the robot to "put a statue on the dresser" (in the purple box), however, the robot lacks precise knowledge of the statue’s location. To accomplish the task, the robot must first gather the necessary information (in this case, the statue’s location), then take further actions as shown in the gray box. In the process of information gathering, classical methods (in red dashed box) search for the statute through onerous trial and error, which is both inefficient and demanding especially in complex environments. In contrast, our proposed method (in green dashed box) empowers the robot to directly ask for the location in natural language and proceed directly to the statute according to the answer, which significantly improves both efficiency and success rates.

such as unfamiliar environments. In such cases, the agents may struggle to make informed decisions. To illustrate this, let’s consider the scenario depicted in Figure 1, where a robot is deployed in an unfamiliar house with the task to put a statue on the dresser. However, the robot lacks prior information about the house like the statue’s location. Consequently, the robot decides to systematically search for every possible position in order, as shown in the red dashed box in Figure 1. Even though the robot finally manages to find the statue, this decision-making process is notably inefficient, not to mention the possibility of suboptimal searching behavior which may lead to failure in finding the statue.

On the other hand, when we humans encounter such scenarios, we tend to adopt a different approach. Rather than onerous trial and error, it is natural for us to actively query external information from our peers or other information sources to accelerate information gathering and guide decision making. Imagine you are invited to your friend’s house and your friend asks you to help move a statue. As shown in Figure 1, instead of opening each and every cabinet to check whether there is a statue, you would likely opt to directly ask "where is the statue?", then directly go to the specific location after you got the answer (as shown in green dashed box).

Building upon this intuition, we focus on a novel setting where the agent can actively query for additional pertinent information from external sources using natural language during their interactions within environments. Though some existing works have explored scenarios involving human-in-the-loop interactions to provide additional information, our setting is stands apart from these previous ones. A majority of works (Nguyen and Daumé III, 2019; Nguyen et al., 2019; Singh et al., 2022b; Da Silva et al., 2020) ask humans for oracle actions or action descriptions, Nguyen et al. (2022) ask for information about current states and (sub-)goals. Liu et al. (2022) asks three-word-templated questions to accelerate training, while Huang et al. (2022b) ask for scene, task, or preferences descriptions. In contrast to existing works, we concentrate on designing a generic mechanism to gather information through natural language, which imposes fewer restrictions and aligns more closely with human decision-making processes.

In this paper, we aim to investigate the feasibility of designing an agent that is able to automatically ask proper questions in unfamiliar environments via natural language. Two questions linger in our mind: Firstly, can the agent ask various questions to gather a variety of necessary information while filtering out the irrelevant one? Furthermore, can the agent remember and reuse the previously

acquired information, thereby avoiding asking for the same information in later-on tasks? The main paper solves these questions in the following organization:

- We introduce *Contextual MDP with Human / External Information Sources in the Loop*, a novel theoretical formulation that is able to formalize the scenarios where the agent can actively query to efficiently gather information via language (Section 3.1).
- We propose Asking Before Action (ABA), an efficient method that is able to accelerate information gathering by allowing the agent to actively query for pertinent information in natural language while interacting with the environments. ABA can learn to ask proper questions even only with a modest modification of existing agents by providing in-context examples. To further improve the performance, we propose to use imitation learning to enable asking diverse yet pertinent questions as well as remembering and reusing the acquired information (Section 3.2).
- Experiments on a series of tasks in ALFWorld (Shridhar et al., 2021) and its variants empirically demonstrate that our method is capable of asking proper questions and acting upon the answers. Our method demonstrates more than 40% improvement in ALFWorld tasks success rate and achieves remarkable performance on tasks that can hardly be completed using previous methods on ALFWorld variants (Section 4).

2 Preliminaries

To effectively portray the obstacles that arise when deploying an agent to unfamiliar environments, we formulate the embodied decision making problem as Contextual Markov Decision Processes (Contextual MDPs) (Hallak et al., 2015).

Definition 2.1 *Contextual MDP* is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{M}(c))$. Here \mathcal{S} and \mathcal{A} stand for state space and action space respectively. \mathcal{C} is the context space. \mathcal{M} is a function mapping context $c \in \mathcal{C}$ to a specific T -horizon MDP $\mathcal{M}(c) = (\mathcal{S}, \mathcal{A}, p(\cdot|s, a, c), r(s, a, c))$.

Here, $\{\mathcal{M}(c), \forall c \in \mathcal{C}\}$ represents a family of MDPs characterized by a shared state space \mathcal{S} and action space \mathcal{A} , but with different transition function $p(\cdot|s, a, c)$ and reward function $r(s, a, c)$ specified by c . The goal of the agent is to learn a policy π to maximize the accumulative rewards on the target environment(s). Denote $\mathcal{C}' \subset \mathcal{C}$ as the context set in evaluation, we would like to optimize for

$$\mathcal{J}(\pi) = \mathbb{E}_{c' \in \mathcal{C}', s_0, p, \pi} \left[\sum_{t=0}^T r(s_t, a_t, c) \right] \quad (1)$$

Note that the context c varies across different environments, and oftentimes, it remains unknown. Optionally, the agent will be additionally provided with a task instruction i , which is usually a concise language description of the goal, providing extra information about the context c . As shown in Definition 2.1, when deployed to a new environment, understanding the context c becomes a prerequisite for comprehending the transition and reward functions and ultimately achieving success. In light of this, one common approach is to gather information about the context c through interactions with the environment by trial and error, i.e., infer the context from history $\hat{c} = f_\theta(s_1, a_1, r_1, \dots, s_t)$ (or $\hat{c} = f_\theta(i, s_1, a_1, r_1, \dots, s_t)$ if i is provided) while trying to solve the task. Here $t \in \{1, 2, \dots, T\}$ and f_θ refers to some learnable encoder of c as in (Zintgraf et al., 2020) Consider an example setting where a robot is tasked with the delivery of food to a bedroom within an unfamiliar house, where c represents the house layout, encompassing the precise locations of the food and the bedroom. To effectively complete the food delivery, the robot must embark on a journey of exploration, aimlessly wandering around to discover both the food and the bedroom.

However, efficiently gathering information in various unknown environments with different contexts c can be challenging. Aside from limited generalization capability Beck et al. (2023), existing methods often rely on dense rewards and sufficiently small state space (Zintgraf et al., 2021), which may lead to catastrophic failure in embodied decision making where the environments often lack carefully crafted dense reward functions and the state spaces are often large.

We argue that this is not, at least always, the case for how we humans deal with unfamiliar environments. Instead of trying to explore everything on our own, we usually turn to another human,

maybe a more experienced senior, and ask for helpful information. This behavior will significantly alleviate the exploration burden in a lot of situations. The above intuition urges us to reconsider the process of embodied decision making in unfamiliar evaluation environments: what if the agent does not necessarily need to figure out everything on itself? Though some prior works have studied the scenarios with human-in-the-loop (refer to Section 5 for a detailed survey), as far as we know, we are the first to deal with the setting of enabling information gathering for embodied decision making with LLM agents.

3 Method

In this section, we present our new problem formulation as well as the corresponding algorithm.

3.1 Contextual MDP with Human / External Information Source in the Loop

To incorporate humans (or other external knowledge sources) in the loop of decision making, the key difference is that the agent is able to interact with humans directly to efficiently gather information:

Definition 3.1 *Contextual MDP with Human / External information source in the loop based on $(\mathcal{S}^U, \mathcal{A}^U, \mathcal{C}, \mathcal{H}(c), \mathcal{M}(c))$. Here $\mathcal{S}^U, \mathcal{A}^U$ are the augmented state space and action space: $\mathcal{S}^U = \mathcal{S} \cup \mathcal{L}_{ans}$, $\mathcal{A}^U = \mathcal{A} \cup \mathcal{L}_{ask}$, where \mathcal{L}_{ask} and \mathcal{L}_{ans} include all possible questions and answers in natural language. $\mathcal{H}(c)$ maps context $c \in \mathcal{C}$ to \mathcal{H}_c , which is a model of human (or other external information source) in context c that can map any questions to information. $\mathcal{M}(c) = (\mathcal{S}^U, \mathcal{A}^U, \mathcal{H}_c, p_U(\cdot|s, a, c, \mathcal{H}_c), r(s, a, c), \gamma)$*

Like Contextual MDP, $\mathcal{M}(c) = (\mathcal{S}^U, \mathcal{A}^U, \mathcal{H}_c, p_U(\cdot|s, a, c, \mathcal{H}_c), r(s, a, c), \gamma)$ has a transition function and a reward function parameterized by c . However, the state space \mathcal{S}^U and the action space \mathcal{A}^U are augmented with answers and questions respectively, and the transition function $p_U(\cdot|s, a, c, \mathcal{H}_c)$ can be factorized as:

$$p_U(s'|s, a, c, \mathcal{H}_c) = p(s'|s, a, c) \cdot \mathbb{1}_{a \in \mathcal{A}} + p(\mathcal{H}_c(a) = s') \cdot \mathbb{1}_{a \in \mathcal{L}_{ask}} \quad (2)$$

With the augmented action space, the agent can now query to gather information while interacting with the environments. For instance, by simply asking "where is the kitchen?", the agent can omit tens of steps of exploration to find the food. However, several challenges exist in this process. Firstly, when deployed to unfamiliar environments, it is important for the agent to identify the key information that is pertinent while filtering out the task-irrelevant ones. Secondly, it would be icing on the cake if the agent can choose to ask only when it cannot reason the answers from historical information.

To solve these challenges, we propose *Asking Before Action* (ABA), an effective method for the language agent to cleverly gather necessary information.

3.2 Asking Before Action

In this paper, we focus on the setting where the task instruction i is provided. Therefore, the agent will integrate i and the historical observations and actions (s_1, a_1, \dots, s_t) by concatenation to get $\tau_t = \text{concat}(i, s_1, a_1, \dots, s_t)$, which is used as the input to the policy $a_t \sim \pi(\tau_t)$.

To efficiently phrase the questions and comprehend the answers, we use pretrained LLMs as the initialization of the agent’s policy. Therefore without loss of generality, in the following of this paper we assume that both the states and the actions are in the form of natural language. However, our method can be easily extended to visual settings with multimodal LLMs such as (Huang et al., 2023; Driess et al., 2023), or be combined with pretrained low-level policies as in Ahn et al. (2022); Singh et al. (2022a) to solve complex robot control tasks.

While notable progress has been made in instruction-following LLMs (Ouyang et al., 2022; Chung et al., 2022), relying solely on the zero-shot deployment of an LLM agent based on task instruction i falls short of meeting the desired outcomes. To this end, we describe two methods in the following sections to further improve the performance. In Section 3.2.1, we introduce a simple yet effective method to improve policy learning via few-shot in-context examples. In Section 3.2.2, to further improve the performance, we propose to do model finetuning with expert demonstration data.

3.2.1 Asking Before Action via In-context Examples

In-context learning (Brown et al., 2020) allows LLMs to learn new tasks just by prepending several input-output examples before the inputs, without even optimizing any model parameters. Its superior efficiency and the ability of generalization has attracted a lot of attention (Xie et al., 2021; Akyürek et al., 2022; Dai et al., 2022). Recent works focus on embodied planning (Huang et al., 2022a; Singh et al., 2022a) or embodied decision making (Yao et al., 2022) with LLM also leverage in-context learning to learn the policy. Therefore, an intuitive and natural way is to provide the agent with examples which show the ability to ask appropriate questions at appropriate time, and then try to generalize to new tasks via in-context learning.

Instead of directly using the history of current task $\tau_t = \text{concat}(i, s_1, a_1, \dots, s_t)$ as inputs, we provide the agent with K human annotated trajectories $\tau^k = \text{concat}(i^k, s_1^k, a_1^k, \dots, s_T^k, a_T^k)$, where $k \in \{1, 2, \dots, K\}$. τ^k contains proper asking actions as well as actions interacting with the environment. We then sample i^k randomly for different k . With K examples, for the current task, the agent will select actions according to

$$a_t \sim \pi_{LLM}(\tau^1, \dots, \tau^K, \tau_t) \quad (3)$$

Instead of directly letting the LLM agent to generate the final action, we follow Ahn et al. (2022) by directly outputting the conditional probability of each action $a_i \in \mathcal{A}$ by

$$a_t = \arg \max_{a \in \mathcal{A}} \prod_{i=0}^{|a|} \pi_{LLM}(e_i | \tau^1, \dots, \tau^K, \tau_t, e_{1:i-1}) \quad (4)$$

where e_i is the i -th token of the action, and $|a|$ refers to the number of tokens of encoded action a . However, in our paper, the action space is augmented with \mathcal{L}_{ask} , and therefore \mathcal{A}^U is infinite. Thus, we propose to first augment the action space with one special action "ask", then score and select. If the action "ask" is selected, the agent will then keep generating the corresponding questions via LLM until the stop token.

3.2.2 Asking Before Action with Imitation Learning

In experiments, we find that when the task time horizon is relatively short, in-context learning is enough. However, when the time horizon is relatively long, purely in-context learning might be insufficient, and the reasons are as follows. First of all, due to the token limitations of LLMs, sometimes even only providing two examples will also result in truncation during evaluation time. Therefore, sometimes we can only use one example which results in limited diversity and thus hampers the performance of in-context learning. Furthermore, as the task horizon increases, the policy usually tends to become more complex. This complexity exacerbates the need for samples, which may further refrain the agent from learning a good policy especially when the number of samples is limited. Taking these two factors into consideration, more proper treatment is needed for the agent to learn a robust policy.

To this end, we propose to further finetune the model via imitation learning. We collect a dataset of N trajectories using expert policy $\mathcal{D} = \{(\tau_t^i, a_t^i, n_t^i)_{t=0}^{T_i}\}_{i=0}^N$ where each trajectory consists of input-output pairs for T_i timesteps and n_t^i is a mask variable. To alleviate the distribution shift problem (Ross et al., 2011), we intentionally corrupt the expert policy by randomly injecting noisy actions with probability p and mark this as noise by setting $n_t^i = 1$ in the dataset. Then, the policy is trained to maximize the probability of actions across trajectories via the cross-entropy loss with all the noisy actions ignored as follows:

$$\mathcal{L} = - \sum_{i=0}^N \sum_{t=0}^{T_i} \log \pi_{LLM}(a_t^i | \tau_t^i) \cdot \mathbb{1}_{n_t^i=0} \quad (5)$$

4 Experiments

In this section, we empirically evaluate ABA on a series of decision making tasks in ALFWorld Shridhar et al. (2021) and its variants. In Section 4.1, we assess the effectiveness of ABA on

Table 1: Success rate on ALFWorld environments for our methods and baselines. ID and OOD refer to in distribution evaluation set and out-of-distribution evaluation set provided in ALFWorld environment respectively. V7B refers to Vicuna 7B model. We report the best BUTLER success rates across 8 seeds aligned with the original paper (Shridhar et al., 2021). For ReAct Yao et al. (2022) and out method (ABA), we report success rates mean and std across 5 seeds.

		Pick	Examine	Clean	Heat	Cool	Pick 2	All
BUTLER (best of 8)	ID	61	39	44	81	60	29	40
	OOD	46	22	39	74	100	24	37
ReAct + V7B (avg of 5)	ID	9 \pm 7	8 \pm 4	9 \pm 3	4 \pm 5	9 \pm 8	4 \pm 4	7 \pm 3
	OOD	3 \pm 3	6 \pm 3	5 \pm 3	10 \pm 3	2 \pm 3	9 \pm 5	6 \pm 1
ABA + V7B (avg of 5)	ID	60 \pm 6	52 \pm 5	59 \pm 6	46 \pm 6	61 \pm 3	61 \pm 10	56 \pm 3
	OOD	37 \pm 5	53 \pm 5	51 \pm 2	52 \pm 6	50 \pm 15	41 \pm 0	48 \pm 2

ALFWorld, demonstrating its capability to formulate proper questions and take subsequent actions. We show ABA results in improvements exceeding 40% in success rate than LLM baseline without asking. In Section 4.2, we extend our evaluation to two variants of ALFWorld, showing the agent’s adeptness in gathering diverse information through question-asking, as well as its ability to retain and reuse acquired knowledge to avoid redundant querying. Notably, these modifications to the environments introduce new challenges that previous methods struggle to solve, while ABA exhibits exceptional performance in tackling these tasks.

4.1 ALFWorld

ALFWorld Shridhar et al. (2021) is an embodied decision making environment based on TextWorld Côté et al. (2019), which serves widely as a testbed in previous papers analyzing embodied decision making with LLMs Yao et al. (2022); Shinn et al. (2023). ALFWorld contains six types of different everyday tasks from ALFRED Shridhar et al. (2020) encompassing activities such as picking and placing, examining in light, cleaning, etc. Within each episode, the agent is deployed in a new room and assigned specific tasks to accomplish. All observations and actions are in natural language.

To minimize human involvement, during the evaluation phase, we implement the model of human (or other information sources) \mathcal{H} via another language model which is instructed to respond to questions based on the provided information. To incorporate prior knowledge about the current room, we extract the information about the object placement from the simulator and transform it into a descriptive paragraph. This paragraph is then fed to \mathcal{H} . Whenever the agent poses a question, \mathcal{H} is tasked with providing an answer based on the paragraph. To further improve the answer accuracy, \mathcal{H} is prompted with several question-answer examples. We use Vicuna-7B (Chiang et al., 2023) to implement \mathcal{H} . For more details, please refer to Appendix A. It’s worth noting that this design allows for the straightforward replacement of the current language model with human or alternative information sources for more appropriate answers.

As for baselines, we propose to use:

- BUTLER (Shridhar et al., 2021), which is an imitation learning-based method without LLM. Instead, it trains independent models using a substantial dataset of 10^5 expert trajectories for each task.
- ReAct (Yao et al., 2022), which is an LLM-based method synergizing reasoning and acting to take actions.

As for the implementation, we use Vicuna-7B Chiang et al. (2023) as the language model for both ReAct and our method, and we incorporate the reasoning process when making decisions (Yao et al., 2022; Ahn et al., 2022). For a fair comparison, we use the same scoring method to select actions for both our method and ReAct. In this section, we present the results for ABA with human-annotated in-context examples. For our method and ReAct, we use K in-context examples with $K = 2$.

The results are presented in Table 1. The results of BUTLER are directly taken from Shridhar et al. (2021) which reports the best performance across 8 different seeds. As for ReAct and ABA, we report

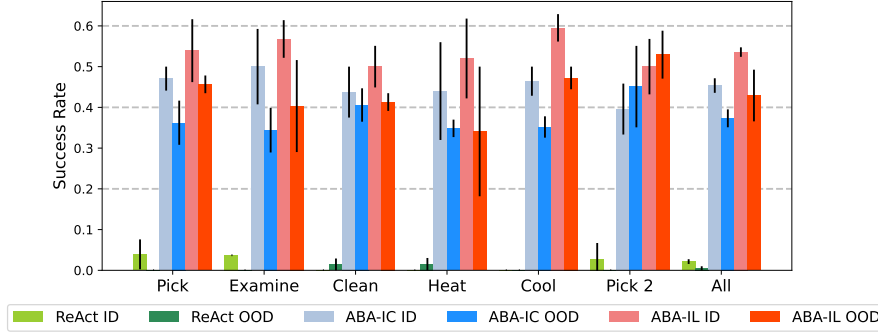


Figure 2: Performance on ALFWorld with ambiguous tasks for our methods and baselines. ID and OOD refer to in-distribution and out-of-distribution evaluation sets.

the performance mean and standard deviation across 5 seeds. As shown in Table 1, ABA average scores outperform BULTEr best scores in 10 out of 14 scenarios despite using only $K = 2$ in-context examples versus 10^5 expert trajectories used by BUTLER. As for ReAct, our method outperforms it across all tasks by a substantial margin. We observe a performance drop compared with the original results in Yao et al. (2022) when switching the model from PaLM-540B Driess et al. (2023) (with a success rate 57%) to Vicuna-7B Chiang et al. (2023) (with a success rate 6%). We hypothesize that the limited model size hampers the reasoning ability, and it is likely that our method would yield even better results with larger models. For example trajectories and qualitative analysis, please refer to Appendix B.

4.2 Modified ALFWorld: Multiround ALFWorld and ALFWorld with ambiguous tasks

To further assess the capabilities of ABA in terms of question asking for gathering diverse information and the ability to remember queried or known information to avoid repetitive questioning, we expand the ALFWorld environment to include two additional variants: ALFWorld with ambiguous tasks and multiround ALFWorld. In this section, we compare two variants of our methods, namely ABA-IC and ABA-IL, with ReAct (Yao et al., 2022). ABA-IC refers to ABA via in-context examples as elaborated in Section 3.2.1, while ABA-IL refers to ABA with imitation learning as elaborated in Section 3.2.2. For ABA-IC and ReAct, we utilize human-annotated examples, while for ABA-IL, we manually design an expert policy to collect data. Additional details can be found in Appendix C. In the following, we will detail the modified environments and present the experiment results:

ALFWorld with ambiguous tasks In this setting, we manually adjusted the task descriptions and reward functions to introduce ambiguity. Instead of providing precise task descriptions, we deliberately left some aspects open-ended, thereby necessitating the agent to gather additional information for successful completion. For instance, in ALFWorld, the task "put a mug on the shelf" is typically considered accomplished as long as any mug is placed on the shelf (there might be multiple mugs in the room). But in this modified setting, the task is only deemed completed when a specific mug is put on the shelf. To complete this task, one can either enumerate all possibilities accordingly until the correct one is identified or directly ask for further clarification about the task.

For ABA-IC and ReAct, we use $K = 2$ in-context examples. For ABA-IL, we collect a dataset of 1500 trajectories. The results are shown in Figure 4.2. Both ABA-IC and ABA-IL consistently exhibit superior performance compared to ReAct, while the baseline fails to complete the task in many scenarios. In Appendix D, we provide example trajectories that demonstrate the effectiveness of ABA-IC and ABA-IL in asking pertinent questions to gather necessary information, while ReAct struggles to conduct consistent exploration. This again highlights the significance of asking: actively questioning for necessary information does not only improve efficiency but also improves success rate, as it proves challenging to gather the necessary information in various complex environments solely using one policy. Furthermore, ABA-IL slightly outperforms ABA-IC. For ID tasks average success rate, ABA-IL achieves 54% while ABA-IC achieves 45%, and for OOD tasks, ABA-IL achieves 43% while ABA-IC achieves 37%, which proves that, compared learning via in-context examples, imitation learning can further improve the performance.

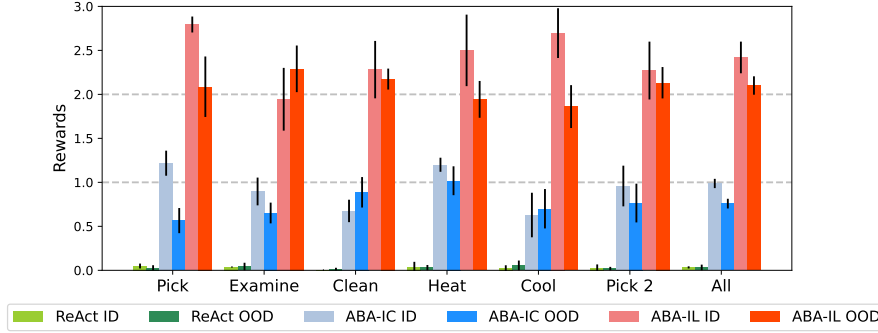


Figure 3: Performance on multi-round ALFWorld for our methods and baselines.

Multiround ALFWorld To further test whether the agent is able to remember the previously known information and avoid asking repeatedly, we introduce multi-round ALFWorld. In previous experiments, the episode ends as long as the current task is completed. Subsequently, in the next episode, the environment will reset to another room with a different layout. In Multi-round ALFWorld, after one task is completed, we randomly sample a new task for the agent to undertake within the same room for multiple rounds. This adjustment enables the agent to familiarize itself with the object placement and provides an opportunity to test its capability to remember and refrain from repetitive questioning. For instance, suppose the agent has previously visited the sidetable to complete a previous task and happened to see there is a mug, or the agent has previously ask about the location of the mug, when the agent is tasked to bring a mug, it can directly go to the location without the need for further inquiries. In this environment, instead of measuring the success rate as in previous experiments, we assign a reward $r = 1$ upon the completion of each task and measure the total reward after T steps.

For ReAct and ABA-IC, we use $K = 1$ in-context example due to the longer trajectory length and token limitation. For ABA-IL, we have a dataset of 500 trajectories. For all experiments, we set $T = 50$. The agent first queries itself whether it has seen a certain object, and asks only when the answer is negative. For more details, please refer to Appendix E. As for the qualitative results, we show the example trajectories in Appendix F, which demonstrates that the agent is capable of recalling previously acquired information and leveraging it in the following tasks. As for the quantitative results, in Figure 4.2, ReAct achieves less than 0.1 rewards in almost all the scenarios. In sharp contrast with that, ABA-IC achieves an average of 0.98 for ID tasks and 0.76 for OOD tasks, while ABA-IL achieves 2.4 and 2.1 respectively. These results indicate that our approach is particularly effective in handling complex tasks.

Moreover, the deterioration in ReAct’s performance compared with previous experiments aligns with our analysis in Section 3.2.2, which suggests that longer in-context examples and smaller K can hinder its effectiveness. While ABA-IC partially overcomes this limitation through a relatively clear policy mapping, we show that ABA-IL can further improve the ABA-IC’s performance by around 2X. These findings provide additional evidence for the effectiveness of our proposed methods.

5 Related Works

5.1 Language Agent

Natural language modeling pre-trained on large-scale unstructured text corpus has seen tremendous success in a variety of applications, including downstream NLP tasks (Radford et al., 2019; Devlin et al., 2018; Brown et al., 2020), logic reasoning (Zhao et al., 2023; Cobbe et al., 2021; Shen et al., 2021), and human-AI coordination (Bubeck et al., 2023; Hu and Sadigh, 2023). The rich information contained in LLMs as an implicit knowledge base also catalyzes the research on in-context learning (Shin et al., 2022; Xie et al., 2021) and prompting (Brown et al., 2020; Wei et al., 2022) that prepend instructions and a few examples to the input of LLMs. However, the time and memory complexity for encoding the prompt is quadratic in the length of the interaction history, such as all the previous trajectories in embodied decision-making, which can increase the burden of the self-attention

mechanism and even exceed the token limitations of LLMs. Despite the techniques introduced to address this issue (Mu et al., 2023; Bulatov et al., 2023), the proposed ABA-IL is inspired by the recent studies on fine-tuning LLMs (Houlsby et al., 2019; Hu and Sadigh, 2023; Lialin et al., 2023), especially those that leverage decision-making signals to train language agents that satisfy certain goals (Carta et al., 2023; Snell et al., 2022a,b).

LLMs have also shown great potential for task planning (Huang et al., 2022b; Lin et al., 2023; Huang et al., 2022a; Wang et al., 2023; Li et al., 2022; Singh et al., 2022a; Carta et al., 2023). However, recent criticisms are made on the planning abilities of LLMs (Bubeck et al., 2023; Valmeekam et al., 2022, 2023). They show that LLMs can get stuck in long-horizon decision-making tasks and the resulting search procedure often degrades to exhaustive search over the large state and action spaces. While pure LLM planning remains a highly challenging open problem, in this work, we investigate the capacity of LLM agents to actively gather information with humans in the loop.

5.2 Embodied Decision Making with Human-in-the-Loop

Some existing works have also studied the scenarios with human-in-the-loop. They query humans for extra information to guide decision making. A majority of works (Nguyen and Daumé III, 2019; Nguyen et al., 2019; Singh et al., 2022b; Da Silva et al., 2020) directly ask humans for oracle actions. They either learn when to ask for oracle actions (Da Silva et al., 2020; Singh et al., 2022b), or learn to leverage oracle actions in language instructions (Nguyen and Daumé III, 2019; Nguyen et al., 2019). Besides oracle actions, Nguyen et al. (2022) asks for new descriptions of current states and (sub-)goals in a POMDP. For asking and answering in language, Huang et al. (2022b) engages humans in active scene description, allowing the LLM to consider human feedback of scene, task, and preferences as inputs. Liu et al. (2022) asks 3-word-templated questions with `<func, adj, noun>` selected by an RL agent and maintains a notebook via uni-gram or bi-gram similarity to accelerate training. It’s worth noting that Yao et al. (2022) also mentions the engagement of humans. Still, it requires humans to supervise and modify the model’s output in real-time if it is wrong, which is different from other human-in-the-loop settings.

Existing works include human-in-the-loop of decision making, either (1) directly asking for numerical vectors like actions/states (Da Silva et al., 2020; Singh et al., 2022b; Nguyen et al., 2022) or (2) querying humans to give exhaustive instruction and learn to convert them to actions (Nguyen and Daumé III, 2019; Nguyen et al., 2019). However, in our setting, we only put a minimal burden on humans and ask them for natural language information which is more natural and more straightforward than providing detailed action instructions for humans. Instead of considering human feedback as the scene (or task, preference) descriptor in the decision making pipeline (Huang et al., 2022b), we formally formulate the setting as *Contextual MDP with Human / External Information Sources in the Loop*, which elaborate the effects of asking via context c and allow the agent to query a broader range of information to gather information. Finally, unlike Liu et al. (2022), we focus on zero-shot adaptation setting and propose more natural end-to-end methods to circumvent the needs of template and similarity designing.

6 Conclusion and Discussion

In this paper, we focus on the setting where the agent can actively query for additional pertinent information from external sources using natural language while interacting in the environments. To formalize this problem, we propose *Contextual MDP with Human / External Information Sources in the Loop*. Then, we propose Asking Before Action (ABA), a method that empowers the agent to ask various questions to gather diverse information and filter out irrelevant ones. ABA is also able to remember and reuse the acquired information in subsequent tasks, thus avoiding redundant queries. In a series of experiments on ALFWorld and its variants, we show qualitatively that ABA is able to propose appropriate questions that satisfy our expectations and make informed decisions based on the answers. Furthermore, the quantitative experiments indicate that ABA consistently outperforms the baselines and achieves a remarkable performance on tasks that are challenging for existing methods. Though currently our method is confined to the language environment, it can readily be extended to incorporate image inputs via multimodal language model (Driess et al., 2023) or tackle control tasks via low-level policies. We believe that this exciting and promising direction has the potential to significantly expand the capabilities and performance of embodied agents.

References

- Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., et al. (2022). Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.
- Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. (2022). What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.
- Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., and Whiteson, S. (2023). A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Bulatov, A., Kuratov, Y., and Burtsev, M. S. (2023). Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*.
- Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., and Oudeyer, P.-Y. (2023). Grounding large language models in interactive environments with online reinforcement learning. *arXiv preprint arXiv:2302.02662*.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (2022). Palm: Scaling language modeling with pathways.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Côté, M.-A., Kádár, A., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., El Asri, L., Adada, M., et al. (2019). Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*, pages 41–75. Springer.
- Da Silva, F. L., Hernandez-Leal, P., Kartal, B., and Taylor, M. E. (2020). Uncertainty-aware action advising for deep reinforcement learning agents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5792–5799.
- Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., and Wei, F. (2022). Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*.

- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. (2023). Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.
- Google (2023). Palm 2 technical report.
- Hallak, A., Di Castro, D., and Mannor, S. (2015). Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*.
- Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.
- Hu, H. and Sadigh, D. (2023). Language instructed reinforcement learning for human-ai coordination. *arXiv preprint arXiv:2304.07297*.
- Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O. K., Liu, Q., et al. (2023). Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*.
- Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. (2022a). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR.
- Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al. (2022b). Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.
- Li, S., Puig, X., Paxton, C., Du, Y., Wang, C., Fan, L., Chen, T., Huang, D.-A., Akyürek, E., Anandkumar, A., et al. (2022). Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212.
- Lialin, V., Deshpande, V., and Rumshisky, A. (2023). Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.
- Lin, K., Agia, C., Migimatsu, T., Pavone, M., and Bohg, J. (2023). Text2motion: From natural language instructions to feasible plans. *arXiv preprint arXiv:2303.12153*.
- Liu, I.-J., Yuan, X., Côté, M.-A., Oudeyer, P.-Y., and Schwing, A. (2022). Asking for knowledge (afk): Training rl agents to query external knowledge using language. In *International Conference on Machine Learning*, pages 14073–14093. PMLR.
- Mu, J., Li, X. L., and Goodman, N. (2023). Learning to compress prompts with gist tokens. *arXiv preprint arXiv:2304.08467*.
- Nguyen, K. and Daumé III, H. (2019). Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. *arXiv preprint arXiv:1909.01871*.
- Nguyen, K., Dey, D., Brockett, C., and Dolan, B. (2019). Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537.
- Nguyen, K. X., Bisk, Y., and Iii, H. D. (2022). A framework for learning to request rich and contextually useful information from humans. In *International Conference on Machine Learning*, pages 16553–16568. PMLR.
- OpenAI (2023). Gpt-4 technical report.

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.
- Shen, J., Yin, Y., Li, L., Shang, L., Jiang, X., Zhang, M., and Liu, Q. (2021). Generate & rank: A multi-task framework for math word problems. *arXiv preprint arXiv:2109.03034*.
- Shin, S., Lee, S.-W., Ahn, H., Kim, S., Kim, H., Kim, B., Cho, K., Lee, G., Park, W., Ha, J.-W., et al. (2022). On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*.
- Shinn, N., Labash, B., and Gopinath, A. (2023). Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.
- Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. (2020). Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.
- Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. (2021). ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. (2022a). Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*.
- Singh, K. P., Weihs, L., Herrasti, A., Choi, J., Kembhavi, A., and Mottaghi, R. (2022b). Ask4help: Learning to leverage an expert for embodied tasks. *Advances in Neural Information Processing Systems*, 35:16221–16232.
- Snell, C., Kostrikov, I., Su, Y., Yang, M., and Levine, S. (2022a). Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*.
- Snell, C., Yang, S., Fu, J., Su, Y., and Levine, S. (2022b). Context-aware language modeling for goal-oriented dialogue systems. *arXiv preprint arXiv:2204.10198*.
- Valmeekam, K., Olmo, A., Sreedharan, S., and Kambhampati, S. (2022). Large language models still can’t plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*.
- Valmeekam, K., Sreedharan, S., Marquez, M., Olmo, A., and Kambhampati, S. (2023). On the planning abilities of large language models (a critical investigation with a proposed benchmark). *arXiv preprint arXiv:2302.06706*.
- Wang, Z., Cai, S., Liu, A., Ma, X., and Liang, Y. (2023). Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2021). An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2022). React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

- Zhao, H., Wang, K., Yu, M., and Mei, H. (2023). Explicit planning helps language models in logical reasoning. *arXiv preprint arXiv:2303.15714*.
- Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. (2020). Varibad: a very good method for bayes-adaptive deep rl via meta-learning. *Proceedings of ICLR 2020*.
- Zintgraf, L. M., Feng, L., Lu, C., Igl, M., Hartikainen, K., Hofmann, K., and Whiteson, S. (2021). Exploration in approximate hyper-state space for meta reinforcement learning. In *International Conference on Machine Learning*, pages 12991–13001. PMLR.

A Design and Details about Human Model

In this section, we describe the design and details about the human model (or other information sources) \mathcal{H} . To minimize human involvement, during the evaluation phase, we implement \mathcal{H} via another language model which is instructed to respond to questions based on the provided information. To incorporate prior knowledge about the current room, we extract the information about the object placement from the simulator and transform it into a descriptive paragraph. This paragraph is then fed to \mathcal{H} . Specifically, we use Vicuna-7B (Chiang et al., 2023) to implement \mathcal{H} . Using a pretrained LLM as \mathcal{H} allows for answering questions in free-form language based on the information provided, which acts just like humans.

To better demonstrate, we provide an example for the ALFWorld experiment in Section 4.1. Other experiments in Section 4.2 are similar. In ALFWorld, the context c mainly refers to the initial mappings of the object placement. For different rooms, the initial mappings are, therefore, different. We slightly abuse the notations about c here since the agent may replace the objects. Under this mapping, we can directly get the ground truth object locations from the simulator, which are unobservable to the agent. Then, we use a rule-based conversion to convert that list to a string of "A is in B", where A refers to the object, while B refers to the place containing the object.

Here is an example. After converting, we derive a descriptive paragraph like:

```
bowl 2 is in diningtable 2. saltshaker 2 is in sidetable
1. spatula 1 is in countertop 1. pot 1 is in stoveburner 4.
spatula 2 is in drawer 1. dish sponge 3 is in diningtable 2.
peppershaker 1 is in cabinet 2. tomato 4 is in sidetable 1.
knife 1 is in diningtable 3. cup 1 is in sidetable 1. bread
2 is in diningtable 3. spatula 3 is in diningtable 2. pan
1 is in cabinet 4. tomato 3 is in fridge 1. potato 1 is in
sinkbasin 1. peppershaker 3 is in diningtable 3. apple 1
is in fridge 1. saltshaker 1 is in cabinet 4. fork 2 is in
drawer 1. spoon 1 is in sidetable 1. egg 1 is in fridge 1.
lettuce 1 is in sidetable 1. plate 1 is in diningtable 2.
```

Whenever the agent poses a question, \mathcal{H} is tasked with providing an answer based on this paragraph. For instance, the agent may learn to ask:

Where can I find the dish sponge?

Then, in this example, the input to \mathcal{H} will be (1) an instruction that tells the model to provide the answers (in gray); (2) a descriptive paragraph (in black); (3) the question proposed by the agent (in blue).

```
Read the following paragraph and answer questions:      bowl
2 is in diningtable 2. saltshaker 2 is in sidetable 1.
spatula 1 is in countertop 1. pot 1 is in stoveburner 4.
spatula 2 is in drawer 1. dish sponge 3 is in diningtable 2.
peppershaker 1 is in cabinet 2. tomato 4 is in sidetable 1.
knife 1 is in diningtable 3. cup 1 is in sidetable 1. bread
2 is in diningtable 3. spatula 3 is in diningtable 2. pan
1 is in cabinet 4. tomato 3 is in fridge 1. potato 1 is in
sinkbasin 1. peppershaker 3 is in diningtable 3. apple 1
is in fridge 1. saltshaker 1 is in cabinet 4. fork 2 is in
drawer 1. spoon 1 is in sidetable 1. egg 1 is in fridge 1.
lettuce 1 is in sidetable 1. plate 1 is in diningtable 2.
The questions is: Where can I find the dish sponge?
```

Then, the pretrained LLM (e.g., Vicuna-7B Chiang et al. (2023) in our case), \mathcal{H} will provide the answers since it can follow the instructions. In our case, the answer is:

```
dish sponge 1 is in garbagecan 1, dish sponge 2 is in drawer
3, dish sponge 3 is in drawer 4.
```

To enhance the performance and improve the accuracy, we provide a few in-context examples and use the prompting format as in Vicuna Chiang et al. (2023). Please refer to Appendix G for details.

It’s worth noting that, though we use a language model to simulate humans in our evaluation phase, this design allows for the straightforward replacement of the current language model with human or alternative information sources for more appropriate answers. The performance of our method will be further improved if \mathcal{H} can provide more accurate and meaningful answers.

B Examples and Qualitative Analysis for ALFWorld Experiments

This section provides some examples and qualitative analysis for ALFWorld experiments in Section 4.1.

First, we provide an episode trajectory of our method (ABA-IC), which is randomly sampled from the evaluation set:

```

Obs 1:    You are in the middle of a room. Looking quickly
around you, you see a bed 1, a diningtable 1, a drawer
4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a
sidetable 2, and a sidetable 1. Your task is to: put a mug
in sidetable.
Act 1: think: To solve the task, I need to find and take a
mug, then put it in sidetable. But where is the mug? Let me
ask that person.
Obs 2: OK.
Act 2: ask: Where is the mug?
Obs 3:    mug 1 is in diningtable 1, mug 3 is in diningtable
1, mug 2 is in diningtable 1.
Act 3: think: We can go to diningtable 1 and take the mug
1, then put it in sidetable.
Obs 4: OK.
Act 4: go to diningtable 1
Obs 5:    On the diningtable 1, you see a creditcard 3, a
creditcard 2, a keychain 3, a keychain 2, a mug 3, a mug 2, a
mug 1, a pen 2, a pen 1, a pencil 3, and a pencil 1.
Act 5: think: Now I find the mug 1. Next, I need to take
it, then put it in sidetable.
Obs 6: OK.
Act 6: take mug 1 from diningtable 1
Obs 7: You pick up the mug 1 from the diningtable 1.
Act 7: think: Now I take a mug 1. Next, I need to put it
in sidetable.
Obs 8: OK.
Act 8: go to sidetable 1
Obs 9: On the sidetable 1, you see a keychain 1.
Act 9: put mug 1 in/on sidetable 1
Obs 10: You put the mug 1 in/on the sidetable 1.

```

It can be concluded that, for our method, the agent can identify the correct context that is related to the task (i.e., the location of the mug) and then summarize the question and ask a pertinent question (i.e., "Where is the mug?"). After that, the agent can integrate the information in the answer (i.e., notice that mug 1 is in diningtable 1) and act according to that information (i.e., go to diningtable 1, take the mug, and place it in the specific location).

For comparison, we provide the trajectory of the same episode of ReAct (Yao et al., 2022):

```

Obs 1:    You are in the middle of a room. Looking quickly
around you, you see a bed 1, a diningtable 1, a drawer
4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a

```

sidetable 2, and a sidetable 1. Your task is to: put a mug in sidetable.

Act 1: think: To solve the task, I need to find and take a mug, then put it in sidetable.

Obs 2: OK.

Act 2: think: First I need to find a mug. A mug is more likely to appear in drawer (1-4), sidetable (1-2). I can check one by one, starting with drawer 1.

Obs 3: OK.

Act 3: go to drawer 1

Obs 4: The drawer 1 is closed.

Act 4: open drawer 1

Obs 5: You open the drawer 1. The drawer 1 is open. In it, you see a cellphone 1.

Act 5: think: Now I find a mug (1). Next, I need to take it.

Obs 6: OK.

Act 6: go to drawer 2

Obs 7: The drawer 2 is closed.

Act 7: open drawer 2

Obs 8: You open the drawer 2. The drawer 2 is open. In it, you see a creditcard 1.

Act 8: think: Now I find a mug (2). Next, I need to take it.

Obs 9: OK.

Act 9: go to drawer 3

Obs 10: On the drawer 3, you see a cellphone 2.

Act 10: think: Now I find a mug (3). Next, I need to take it.

Obs 11: OK.

Act 11: go to drawer 4

Obs 12: On the drawer 4, you see nothing.

Act 12: think: Now I find a mug (4). Next, I need to take it.

Obs 13: OK.

Act 13: go to sidetable 1

Obs 14: On the sidetable 1, you see a keychain 1.

Act 14: think: Now I find a mug (1). Next, I need to put it in sidetable 1.

Obs 15: OK.

Act 15: go to sidetable 2

Obs 16: On the sidetable 2, you see a alarmclock 1, a cd 1, and a pencil 2.

Act 16: think: Now I find a mug (2). Next, I need to put it in sidetable 2.

Obs 17: OK.

Act 17: go to drawer 1

Obs 18: The drawer 1 is open. In it, you see a cellphone 1.

Act 18: think: Now I find a mug (1). Next, I need to put it in drawer 1.

Obs 19: OK.

Act 19: go to drawer 2

Obs 20: The drawer 2 is open. In it, you see a creditcard 1.

Act 20: think: Now I find a mug (2). Next, I need to put it in drawer 2.

Obs 21: OK.

Act 21: go to drawer 3

Obs 22: On the drawer 3, you see a cellphone 2.
Act 22: think: Now I find a mug (3). Next, I need to put it in drawer 3.
Obs 23: OK.
Act 23: go to drawer 4
Obs 24: On the drawer 4, you see nothing.
Act 24: think: Now I find a mug (4). Next, I need to put it in drawer 4.
Obs 25: OK.
Act 25: go to bed 1
Obs 26: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 26: think: Now I find a mug (1). Next, I need to put it on bed 1.
Obs 27: OK.
Act 27: go to sidetable 1
Obs 28: On the sidetable 1, you see a keychain 1.
Act 28: think: Now I find a mug (1). Next, I need to put it on sidetable 1.
Obs 29: OK.
Act 29: go to sidetable 2
Obs 30: On the sidetable 2, you see a alarmclock 1, a cd 1, and a pencil 2.
Act 30: think: Now I find a mug (1). Next, I need to put it on sidetable 2.
Obs 31: OK.
Act 31: go to drawer 1
Obs 32: The drawer 1 is open. In it, you see a cellphone 1.
Act 32: think: Now I find a mug (1). Next, I need to put it in drawer 1.
Obs 33: OK.
Act 33: go to drawer 2
Obs 34: The drawer 2 is open. In it, you see a creditcard 1.
Act 34: think: Now I find a mug (1). Next, I need to put it in drawer 2.
Obs 35: OK.
Act 35: go to drawer 3
Obs 36: On the drawer 3, you see a cellphone 2.
Act 36: think: Now I find a mug (2). Next, I need to put it in drawer 3.
Obs 37: OK.
Act 37: go to drawer 4
Obs 38: On the drawer 4, you see nothing.
Act 38: think: Now I find a mug (3). Next, I need to put it in drawer 4.
Obs 39: OK.
Act 39: go to bed 1
Obs 40: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 40: think: Now I find a mug (1). Next, I need to put it on bed 1.
Obs 41: OK.
Act 41: go to sidetable 1
Obs 42: On the sidetable 1, you see a keychain 1.
Act 42: think: Now I find a mug (1). Next, I need to put it on sidetable 1.
Obs 43: OK.

Act 43: go to sidetable 2
Obs 44: On the sidetable 2, you see a alarmclock 1, a cd 1, and a pencil 2.
Act 44: think: Now I find a mug (1). Next, I need to put it on sidetable 2.
Obs 45: OK.
Act 45: go to drawer 1
Obs 46: The drawer 1 is open. In it, you see a cellphone 1.
Act 46: think: Now I find a mug (1). Next, I need to put it in drawer 1.
Obs 47: OK.
Act 47: go to drawer 2
Obs 48: The drawer 2 is open. In it, you see a creditcard 1.
Act 48: think: Now I find a mug (1). Next, I need to put it in drawer 2.
Obs 49: OK.
Act 49: go to drawer 3
Obs 50: On the drawer 3, you see a cellphone 2.
Act 50: think: Now I find a mug (2). Next, I need to put it in drawer 3.
Obs 51: OK.

It can be concluded that the agent failed to provide the right plan at first (i.e., Action 1,2), since none of these locations has a mug. Moreover, even with the plans, it failed to yield coherent and correct actions in later steps. For instance, in Action 5, 8, 10, 12, etc., the agent is constantly confused by hallucination (it believes that "Now I find a mug" while there wasn't a mug actually). It also got stuck in loops (it repeatedly visited drawer 1,2,3,4, sidetable 1,2, and bed 1). These together prevent the agent from finishing the task.

To further understand the experiment results, we provide more examples for our method and ReAct.

For our method:

Obs 1: You are in the middle of a room. Looking quickly around you, you see a bathtubbasin 1, a countertop 1, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a shelf 2, a shelf 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1. Your task is to: put some spraybottle on toilet.
Act 1: think: To solve the task, I need to find and take a spraybottle, then put it on the toilet. But where is the spraybottle? Let me ask that person.
Obs 2: OK.
Act 2: ask: Where is the spraybottle?
Obs 3: spraybottle 3 is in countertop 1, spraybottle 4 is in dresser 1, spraybottle 2 is in shelf 1.
Act 3: think: We can go to countertop 1 and take the spraybottle 3, then put it on the toilet.
Obs 4: OK.
Act 4: go to countertop 1
Obs 5: On the countertop 1, you see a mirror 1, a soapbar 1, and a spraybottle 1.
Act 5: think: Now I find the spraybottle 1. Next, I need to take it, then put it on the toilet.
Obs 6: OK.
Act 6: take spraybottle 1 from countertop 1
Obs 7: You pick up the spraybottle 1 from the countertop 1.

Act 7: think: Now I take a spraybottle 1. Next, I need to put it on the toilet.
 Obs 8: OK.
 Act 8: go to toilet 1
 Obs 9: On the toilet 1, you see a candle 1, a toiletpaper 2, and a toiletpaper 1.
 Act 9: think: Now I put the spraybottle 1 on the toilet.
 Obs 10: OK.
 Act 10: put spraybottle 1 in/on toilet 1
 Obs 11: You put the spraybottle 1 in/on the toilet 1.

For ReAct:

Obs 1: You are in the middle of a room. Looking quickly around you, you see a bathtubbasin 1, a countertop 1, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a shelf 2, a shelf 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1. Your task is to: put some spraybottle on toilet.
 Act 1: think: To solve the task, I need to find and take a spraybottle, then put it on toilet.
 Obs 2: OK.
 Act 2: think: First I need to find a spraybottle. A spraybottle is more likely to appear in drawer (1-4), dresser (1), sinkbasin (1), toilet (1), and towelholder (1). I can check one by one, starting with drawer 1.
 Obs 3: OK.
 Act 3: go to drawer 1
 Obs 4: The drawer 1 is closed.
 Act 4: open drawer 1
 Obs 5: You open the drawer 1. The drawer 1 is open. In it, you see nothing.
 Act 5: think: Now I find a spraybottle (not). Next, I need to take it.
 Obs 6: OK.
 Act 6: go to drawer 2
 Obs 7: The drawer 2 is closed.
 Act 7: open drawer 2
 Obs 8: You open the drawer 2. The drawer 2 is open. In it, you see a candle 3.
 Act 8: think: Now I find a spraybottle (not). Next, I need to take it.
 Obs 9: OK.
 Act 9: go to drawer 3
 Obs 10: The drawer 3 is closed.
 Act 10: open drawer 3
 Obs 11: You open the drawer 3. The drawer 3 is open. In it, you see nothing.
 Act 11: think: Now I find a spraybottle (not). Next, I need to take it.
 Obs 12: OK.
 Act 12: go to drawer 4
 Obs 13: The drawer 4 is closed.
 Act 13: open drawer 4
 Obs 14: You open the drawer 4. The drawer 4 is open. In it, you see nothing.
 Act 14: think: Now I find a spraybottle (not). Next, I need to take it.

Obs 15: OK.
Act 15: go to dresser 1
Obs 16: On the dresser 1, you see a candle 2, a soapbottle 3, a soapbottle 2, and a spraybottle 4.
Act 16: take spraybottle 4 from dresser 1
Obs 17: You pick up the spraybottle 4 from the dresser 1.
Act 17: think: Now I find a spraybottle (4). Next, I need to put it in/on toilet 1.
Obs 18: OK.
Act 18: go to toilet 1
Obs 19: On the toilet 1, you see a candle 1, a toiletpaper 2, and a toiletpaper 1.
Act 19: put spraybottle 4 in/on toilet 1
Obs 20: You put the spraybottle 4 in/on the toilet 1.

From the above examples, our method can ask proper questions and act accordingly. Though \mathcal{H} provides information with slight error (e.g., it mentions that spraybottle 3 is in countertop 1, but only spraybottle 1 is in countertop 1), the agent is robust to such error and successfully adjust its behavior after observing the objects on countertop 1 (i.e., action 5, 6, 7).

As for ReAct, it successfully visited four drawers and finally found the spraybottle at dresser 1. However, first, it failed to list every possible container for the spraybottle (i.e., action 2, it omits countertop, shelf, etc.). In the reasoning step, we observe an interesting pattern (i.e., in Action 5, 8, 11, 14): "Now I find a spraybottle (not). Next, I need to take it", which seems inconsistent (though it does not affect the next step). Moreover, though the agent finally finds the spraybottle and completes the task successfully, it is inefficient and slow to search every possible location: ReAct takes 20 steps. In comparison, our method only takes 10 steps to finish the task.

Four above examples demonstrate that, first, it is challenging to learn a information-gathering policy especially in unfamiliar environments, due to the complexity of the environment. Moreover, even if the agent manage to follow this policy, the information-gathering phase can be inefficient, which needs to exhaustively search every possible position. In contrast, our method succeeds in proposing proper questions and then acting accordingly, which improve the success rate as well as the efficiency. This proves our method's efficacy.

C Details about the Data Collection and Environment Variants

In this section, we provide details about how the data is collected and training as mentioned in Section 4.2.

As for in-context examples used in ABA-IC, we manually interact with the environment and try to finish the tasks. We ask questions related to the tasks, and answer the questions ourselves by checking the ground truth states in the simulator. Beside the questions, we also add reasoning steps as in Yao et al. (2022) and select actions according to the information we have. Once completing the task, we take down all the actions and observations and use them as in-context examples.

As for ABA-IL, we design a rule-based policy according to the PDDL planning trajectories provided along with the environment. Specifically, we integrate the PDDL trajectories and the ground truth states within the simulator to find out what we should do to finish the tasks. Then, we extract the ground truth placements of the necessary objects from the simulator, and we write template-based questions to query this information and provide corresponding answers as observations. We also write chain-of-thought reasoning steps. As mentioned in Section 3.2.2, we manually inject noises by randomly inserting noisy actions at probability $p = 0.2$. These noisy actions are randomly sampled from the action space. The planning trajectories are also modified accordingly to correctly finish the task. Finally, we organize these questions, reasoning, and interactive actions to get a list of actions. When the actions belong to asking for reasoning, we use provided answers or simply "OK." as observations. When the actions aim to interact with the environment, we use environmental feedback as the observations. As for Ambiguous ALFWorld, we use $K = 2$ for ABA-IC and collect 1500 trajectories for ABA-IL. As for Multiround ALFWorld, we use $K = 1$ for ABA-IC and collect 500 trajectories for ABA-IL. As for training, to ease the computational burden, we use LoRA (Hu et al.,

2021) with $r = 16$ and a learning rate of $1e - 5$. We train the model with 3000 steps for Ambiguous ALFWorld and 6000 steps for Multiround ALFWorld.

D Examples and Qualitative Analysis for ALFWorld with Ambiguous Tasks

This section provides some examples and qualitative analysis for ALFWorld with Ambiguous Tasks experiments in Section 4.2.

First, we provide an episode trajectory of ReAct, which is randomly sampled from the evaluation set:

Obs 1: You are in the middle of a room. Looking quickly around you, you see a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 2, a diningtable 1, 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: put a hot egg in diningtable.

Act 1: think: To finish the task, I need to find an egg, then heat it with microwave, then put it in/on diningtable.

Obs 2: OK.

Act 2: think: First I need to find an egg. An egg is more likely to appear in stoveburner 2, countertop 1-3, diningtable 1, and cabinet 9-10. But I have checked stoveburner 2 and countertop 1-3. I can check the remaining places one by one, starting with cabinet 9.

Obs 3: OK.

Act 3: go to cabinet 9

Obs 4: The cabinet 9 is closed.

Act 4: think: An egg does not appear in cabinet 9.

Obs 5: OK.

Act 5: go to cabinet 10

Obs 6: The cabinet 10 is closed.

Act 6: think: An egg does not appear in cabinet 10.

Obs 7: OK.

Act 7: go to diningtable 1

Obs 8: On the diningtable 1, you see a bread 2, a bread 1, a butterknife 2, a dish sponge 1, a fork 1, a potato 1, and a spatula 1.

Act 8: think: I have checked all the remaining places, and I have not found an egg. The task is not finished.

Obs 9: OK.

Act 9: think: I have not found an egg. This is not a failure, just an unexpected result.

Obs 10: OK.

Act 10: think: What should I do now? The task is not finished.

Obs 11: OK.

Act 11: think: I could try to find an egg again.

Obs 12: OK.

Act 12: go to stoveburner 1

Obs 13: On the stoveburner 1, you see nothing.

Act 13: go to countertop 1

Obs 14: On the countertop 1, you see a butterknife 1, and a mug 1.

Act 14: think: It is possible that an egg is hidden under one of these objects.

Obs 15: OK.

Act 15: go to countertop 2
Obs 16: On the countertop 2, you see a bowl 2, and a knife 1.
Act 16: think: It is possible that an egg is hidden under one of these objects.
Obs 17: OK.
Act 17: go to countertop 1
Obs 18: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 18: think: It is possible that an egg is hidden under one of these objects.
Obs 19: OK.
Act 19: go to countertop 2
Obs 20: On the countertop 2, you see a bowl 2, and a knife 1.
Act 20: think: It is possible that an egg is hidden under one of these objects.
Obs 21: OK.
Act 21: go to countertop 1
Obs 22: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 22: think: It is possible that an egg is hidden under one of these objects.
Obs 23: OK.
Act 23: go to countertop 2
Obs 24: On the countertop 2, you see a bowl 2, and a knife 1.
Act 24: think: It is possible that an egg is hidden under one of these objects.
Obs 25: OK.
Act 25: go to countertop 1
Obs 26: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 26: think: It is possible that an egg is hidden under one of these objects.
Obs 27: OK.
Act 27: go to countertop 2
Obs 28: On the countertop 2, you see a bowl 2, and a knife 1.
Act 28: think: It is possible that an egg is hidden under one of these objects.
Obs 29: OK.
Act 29: go to countertop 1
Obs 30: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 30: think: It is possible that an egg is hidden under one of these objects.
Obs 31: OK.
Act 31: go to countertop 2
Obs 32: On the countertop 2, you see a bowl 2, and a knife 1.
Act 32: think: It is possible that an egg is
Obs 33: OK.
Act 33: go to countertop 1
Obs 34: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 34: think: It is possible that an egg is hidden under one of these objects.
Obs 35: OK.

Act 35: go to countertop 2
Obs 36: On the countertop 2, you see a bowl 2, and a knife 1.
Act 36: think: It is possible that an egg is hidden under one
Obs 37: OK.
Act 37: go to countertop 1
Obs 38: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 38: think: It is possible that an egg is hidden under one of these objects.
Obs 39: OK.
Act 39: go to countertop 2
Obs 40: On the countertop 2, you see a bowl 2, and a knife 1.
Act 40: think: It is possible that an egg is hidden under one of these objects.
Obs 41: OK.
Act 41: go to countertop 1
Obs 42: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 42: think: It is possible that an egg is hidden under one of these objects.
Obs 43: OK.
Act 43: go to countertop 2
Obs 44: On the countertop 2, you see a bowl 2, and a knife 1.
Act 44: think: It is possible that an egg is hidden under one of these objects.
Obs 45: OK.
Act 45: go to countertop 1
Obs 46: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 46: think: It is possible that an egg is hidden under one of these objects.
Obs 47: OK.
Act 47: go to countertop 2
Obs 48: On the countertop 2, you see a bowl 2, and a knife 1.
Act 48: think: It is possible that an egg is hidden under one of these objects.
Obs 49: OK.
Act 49: go to countertop 1
Obs 50: On the countertop 1, you see a butterknife 1, and a mug 1.
Act 50: think: It is possible that an egg is hidden under one of these objects.
Obs 51: OK.

In this episode, the agent failed to find the egg. Initially, it failed to propose correct plans (i.e., action 2). It listed several locations where the eggs could be but not all (e.g., cabinet 1-8, diningtable 2, fridge, etc). Then, it incorrectly thought it had checked some of these places, which suggested there might be hallucinations. These two steps limit the searching range of the agent and prevent the agent from finding the target object. In the following steps, it checked cabinet 9-10 (but without opening the cabinets), stoveburner 1, countertop 1-2, and then got stuck by repeatedly visiting the countertop 1-2.

The episode of the same room setting for our method (ABA-IL) is:

Obs 1: You are in the middle of a room. Looking quickly around you, you see a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 2, a diningtable 1, 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: put a hot mug in cabinet.

Act 1: think: To solve the task, I need to find and take a mug, then heat it with microwave, then put it in cabinet. But where is the mug? Let me ask that person.

Obs 2: OK.

Act 2: ask: Where is the mug?

Obs 3: mug 1 is in countertop 1, mug 2 is in microwave 1.

Act 3: think: There are multiple mug. I need to ask which mug should be taken.

Obs 4: OK.

Act 4: ask: Which mug do you prefer?

Obs 5: I mean mug 1.

Act 5: think: Now I understand the task. I can go to countertop 1 and take the mug 1, then heat it with microwave, then put it in cabinet.

Obs 6: OK.

Act 6: go to countertop 1

Obs 7: On the countertop 1, you see a butterknife 1, and a mug 1.

Act 7: take mug 1 from countertop 1

Obs 8: You pick up the mug 1 from the countertop 1.

Act 8: think: Now I take a mug 1. Next, I need to go to microwave 1 and heat mug 1, then put it in cabinet.

Obs 9: OK.

Act 9: go to microwave 1

Obs 10: The microwave 1 is closed.

Act 10: heat mug 1 with microwave 1

Obs 11: You heat the mug 1 using the microwave 1.

Act 11: think: Now I heat mug 1. Next, I need to put it in cabinet 1.

Obs 12: OK.

Act 12: go to cabinet 1

Obs 13: On the cabinet 1, you see nothing.

Act 13: put mug 1 in/on cabinet 1

The above trajectory shows that, for our method (ABA-IL), the agent is able to identify the correct object related to the task (i.e., mug) and propose a proper question (i.e., Action 2) to ask the position of the mug, which is also shown in Appendix B. Moreover, when it realized that there were multiple mugs in the room, it further asked for clarification about the task (i.e., Action 4) since in Ambiguous ALFWorld, only the specified object is desirable. This demonstrates that it can ask diverse questions to query various information. Finally, the agent is able to integrate the acquired information into the following decision making: it knows which object is needed, it can head for the correct mug, heat it, and place it on the target location. In contrast to ReAct trajectory where the agent got stuck before finding the mug, the efficiency in qualitative examples further proves our method’s efficacy.

As a supplement, we show a typical failure case of our method:

Obs 1: You are in the middle of a room. Looking quickly around you, you see a cabinet 27, a cabinet 26, a cabinet 25, a cabinet 24, a cabinet 23, a cabinet 22, a cabinet 21, a cabinet 20, a cabinet 19, a cabinet 18, a cabinet 17, a cabinet 16, a cabinet 15, a cabinet 14, a cabinet 13, a

cabinet 12, a cabinet 11, a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 1, a drawer 12, a drawer 11, a drawer 10, a drawer 9, a drawer 8, a drawer 7, 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 spatula and put it in drawer.

Act 1: think: To solve the task, I need to find and take a spatula, then clean it with sinkbasin, then put it in drawer. But where is the spatula? Let me ask that person.

Obs 2: OK.

Act 2: ask: Where is the spatula?

Obs 3: spatula 1 is in countertop 2, spatula 2 is in sinkbasin 1.

Act 3: think: There are multiple spatula. I need to ask which spatula should be taken.

Obs 4: OK.

Act 4: ask: Which spatula do you prefer?

Obs 5: I mean, it depends on personal preference.

Act 5: think: Now I understand the task better. I need to ask which spatula should be taken, then find and take it, then clean it with sinkbasin, then put it in drawer.

Obs 6: OK.

Act 6: ask: Which spatula do you prefer?

Obs 7: I am not sure. Could you remind me the information about each spatula?

Act 7: think: There are multiple spatula. Let me ask which spatula should be taken first.

Obs 8: OK.

Act 8: ask: Which spatula do you prefer?

Obs 9: I am not sure. Could you remind me the information about each spatula?

Act 9: think: Now I understand the task better. I need to ask which spatula should be taken first, then find and take it, then clean it with sinkbasin, then put it in drawer.

Obs 10: OK.

Act 10: ask: Which spatula do you prefer?

Obs 11: I am not sure. Could you remind me the information about each spatula?

Act 11: think: Now I understand the task better. I need to ask which spatula should be taken first, then find and take it, then clean it with sinkbasin, then put it in drawer.

Obs 12: OK.

Act 12: ask: Which spatula do you prefer?

Obs 13: ...

Though \mathcal{H} is instructed with the goal, namely which spatula it is looking for (see Appendix G for more details), it fails to provide the correct answer (e.g., Obs 5,7,9,11). Therefore, we assume that by providing more accurate \mathcal{H} , the performance of our model will further enhance.

E Details about Multiround ALFWorld

In this section, we provide more details about data collection in Multiround ALFWorld. In Multiround ALFWorld, the main challenge is proposing the right questions. Specifically, the agent needs to avoid repeatedly asking by identifying whether the information to query has already been collected. This

requires a special treatment to the data and we explicitly implement this for clarity. In our case, the agent first asks itself whether it has seen a specific object before asking a question. Only when the answer is negative will it continue to ask. Otherwise, it may directly act based on its memory.

For ABA-IC, we provide the agent with manually labeled trajectories, in which we manually identify whether the agent needs to ask according to previous interactions, and only ask for more information if needed. As for ABA-IL, we integrate this part in the reasoning step. To be specific, the reasoning will include an explicit query about the target object. When the agent has never seen a particular object, the reasoning step will be like:

```
think: To solve the task, I need to find and take a mug,
then put it in sidetable. First I need to find the locations
of mug. ### query: mug > I have never seen mug before.
```

In the above example, the target object is the mug (i.e., "query: mug"), and the agent believes it has never seen the mug before (i.e., "I have never seen mug before.").

On the other hand, if the agent has ever seen the object (e.g., it has visited diningtable 1 and seen pencil 1 and pencil 3 there), the query and the answer will be like:

```
think: To solve the task, I need to find and take a
pencil, then put it in garbagecan. First I need to find the
locations of pencil. ### query: pencil > pencil 3 is in
diningtable 1, pencil 1 is in diningtable 1.
```

After querying itself about the target object, the will act according to the answers. If the agent believes it has never seen the object, it will probably ask for more information. Otherwise, it will directly make decisions depend on the information.

For ABA-IL, we make it clearer to split the reasoning into two steps. For the first step, we identify the target object. For the second step, we identify whether and where we have seen this object before. These two steps form two pairs of input/output training samples in the dataset. During training, we use teacher forcing. We further augment the dataset by inserting several new queries asking for another object which is randomly sampled from the object list in this environment.

F Examples and Qualitative Analysis for Multi-round ALFWorld

In this section, we provide some examples and qualitative analysis for Multi-round ALFWorld experiments in Section 4.2.

We first provide an trajectory for ReAct, which is randomly sampled from the evaluation set:

```
Obs 1: You are in the middle of a room. Looking quickly
around you, you see a bed 1, a diningtable 1, a drawer 4, a
drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a sidetable
2, and a sidetable 1. Your task is to: put a cellphone in
diningtable.
Act 1: think: To solve the task, I need to find and take a
cellphone, then put it in diningtable.
Obs 2: OK.
Act 2: think: First I need to find a cellphone. A
cellphone is more likely to appear in drawer 1-4. But there
is no cellphone on drawer 1-2. I can check one by one,
starting with drawer 3.
Obs 3: OK.
Act 3: go to drawer 3
Obs 4: On the drawer 3, you see a cellphone 2.
Act 4: take cellphone 2 from drawer 3
Obs 5: You pick up the cellphone 2 from the drawer 3.
Act 5: think: Now I take a cellphone 2. Next, I need to
put it in diningtable 1.
```

Obs 6: OK.
Act 6: go to diningtable 1
Obs 7: On the diningtable 1, you see a creditcard 3, a creditcard 2, a keychain 3, a keychain 2, a mug 3, a mug 2, a mug 1, a pen 2, a pen 1, a pencil 3, and a pencil 1.
Act 7: put cellphone 2 in/on diningtable 1
Obs 8: You put the cellphone 2 in/on the diningtable 1.
Your next task is to: put a pen in sidetable.
Act 8: think: To solve the task, I need to find and take a pen and go to a sidetable.
Obs 9: OK.
Act 9: think: First I need to find a pen. A pen is more likely to appear in drawer 2, sidetable 2. But there is no pen on drawer 2, therefore I need to check sidetable 2.
Obs 10: OK.
Act 10: go to sidetable 2
Obs 11: On the sidetable 2, you see a alarmclock 1, a cd 1, and a pencil 2.
Act 11: think: Now I find a pencil 2. Next, I need to take it.
Obs 12: OK.
Act 12: take pencil 2 from sidetable 2
Obs 13: You pick up the pencil 2 from the sidetable 2.
Act 13: think: Now I take a pencil 2.
Obs 14: OK.
Act 14: go to sidetable 1
Obs 15: On the sidetable 1, you see a keychain 1.
Act 15: put pencil 2 in/on sidetable 1
Obs 16: You put the pencil 2 in/on the sidetable 1.
Act 16: think: The task is complete.
Obs 17: OK.
Act 17: go to garbagecan 1
Obs 18: On the garbagecan 1, you see nothing.
Act 18: think: I have finished to put all the objects in their places.
Obs 19: OK.
Act 19: go to bed 1
Obs 20: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 20: think: The room is clean.
Obs 21: OK.
Act 21: go to diningtable 1
Obs 22: On the diningtable 1, you see a cellphone 2, a creditcard 3, a creditcard 2, a keychain 3, a keychain 2, a mug 3, a mug 2, a mug 1, a pen 2, a pen 1, a pencil 3, and a pencil 1.
Act 22: think: The room is clean.
Obs 23: OK.
Act 23: go to bed 1
Obs 24: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 24: think: The room is clean.
Obs 25: OK.
Act 25: go to garbagecan 1
Obs 26: On the garbagecan 1, you see nothing.
Act 26: think: I have finished to put all the objects in their places.
Obs 27: OK.
Act 27: go to bed 1

Obs 28: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 28: think: The room is clean.
Obs 29: OK.
Act 29: go to diningtable 1
Obs 30: On the diningtable 1, you see a cellphone 2, a creditcard 3, a creditcard 2, a keychain 3, a keychain 2, a mug 3, a mug 2, a mug 1, a pen 2, a pen 1, a pencil 3, and a pencil 1.
Act 30: think: The room is clean.
Obs 31: OK.
Act 31: go to bed 1
Obs 32: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 32: think: The room is clean.
Obs 33: OK.
Act 33: go to garbagecan 1
Obs 34: On the garbagecan 1, you see nothing.
Act 34: think: I have finished to put all the objects in their places.
Obs 35: OK.
Act 35: go to bed 1
Obs 36: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 36: think: The room is clean.
Obs 37: OK.
Act 37: go to diningtable 1
Obs 38: On the diningtable 1, you see a cellphone 2, a creditcard 3, a creditcard 2, a keychain 3, a keychain 2, a mug 3, a mug 2, a mug 1, a pen 2, a pen 1, a pencil 3, and a pencil 1.
Act 38: think: The room is clean.
Obs 39: OK.
Act 39: go to bed 1
Obs 40: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 40: think: The room is clean.
Obs 41: OK.
Act 41: go to garbagecan 1
Obs 42: On the garbagecan 1, you see nothing.
Act 42: think: I have finished to put all the objects in their places.
Obs 43: OK.
Act 43: go to bed 1
Obs 44: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 44: think: The room is clean.
Obs 45: OK.
Act 45: go to diningtable 1
Obs 46: On the diningtable 1, you see a cellphone 2, a creditcard 3, a creditcard 2, a keychain 3, a keychain 2, a mug 3, a mug 2, a mug 1, a pen 2, a pen 1, a pencil 3, and a pencil 1.
Act 46: think: The room is clean.
Obs 47: OK.
Act 47: go to bed 1
Obs 48: On the bed 1, you see a book 2, a book 1, a laptop 2, a laptop 1, and a pillow 1.
Act 48: think: The room is clean.

Obs 49: OK.
Act 49: go to garbagecan 1
Obs 50: On the garbagecan 1, you see nothing.
Act 50: think: I have finished to put all the objects in their places.

We show that, in the above example, though the agent successfully located the cellphone after opening drawer3, the initial planning (i.e., Action 2) seemed to contain hallucination ("There is no cellphone on drawer 1-2"), which makes the result a bit of a coincidence. After finishing the first task, the agent again makes some erroneous claims that seem like hallucinations (i.e., Action 9, "there is no pen on drawer 2"). Then, the agent searched for the remaining locations and mistook the "pencil" for the "pen" (i.e., Action 11), and got stuck after wrongly placing the pencil on the target location.

In comparison, we provide an episode trajectory of our method:

Obs 1: You are in the middle of a room. Looking quickly around you, you see a bathtubbasin 1, a countertop 1, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a shelf 2, a shelf 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1. Your task is to: put a toiletpaper in drawer.
Act 1: think: To solve the task, I need to find and take a toiletpaper, then put it in drawer. First I need to find the locations of toiletpaper. ### query: toiletpaper > I have never seen toiletpaper before.
Obs 2: OK.
Act 2: think: I cannot locate toiletpaper, I need to ask the owner of this room.
Obs 3: OK.
Act 3: ask: Where is the toiletpaper?
Obs 4: toiletpaper 1 is in toilet 1, toiletpaper 2 is in drawer 2.
Act 4: think: I can go to toilet 1 and take the toiletpaper, then put it in drawer.
Obs 5: OK.
Act 5: go to toilet 1
Obs 6: On the toilet 1, you see a candle 1, a toiletpaper 2, and a toiletpaper 1.
Act 6: take toiletpaper 1 from toilet 1
Obs 7: You pick up the toiletpaper 1 from the toilet 1.
Act 7: think: Now I take a toiletpaper 1. Next, I need to put it in drawer.
Obs 8: OK.
Act 8: go to drawer 1
Obs 9: The drawer 1 is closed.
Act 9: open drawer 1
Obs 10: You open the drawer 1. The drawer 1 is open. In it, you see nothing.
Act 10: put toiletpaper 1 in/on drawer 1
Obs 11: You put the toiletpaper 1 in/on the drawer 1. Your next task is to: put some candle on shelf.
Act 11: think: To solve the task, I need to find and take a candle, then put it in shelf. First I need to find the locations of candle. ### query: candle > candle 1 is in toilet 1.
Obs 12: OK.
Act 12: think: I can go to toilet 1 and take the candle, then put it in shelf.
Obs 13: OK.

Act 13: go to toilet 1
 Obs 14: On the toilet 1, you see a candle 1, and a
 toiletpaper 2.
 Act 14: take candle 1 from toilet 1
 Obs 15: You pick up the candle 1 from the toilet 1.
 Act 15: think: Now I take a candle 1. Next, I need to put
 it in shelf.
 Obs 16: OK.
 Act 16: go to shelf 1
 Obs 17: On the shelf 1, you see a spraybottle 2.
 Act 17: put candle 1 in/on shelf 1
 Obs 18: You put the candle 1 in/on the shelf 1. Your next
 task is to: put some toiletpaper on garbagecan.
 Act 18: think: To solve the task, I need to find and take
 a toiletpaper, then put it in garbagecan. First I need to
 find the locations of toiletpaper. ### query: toiletpaper >
 toiletpaper 1 is in drawer 1, toiletpaper 2 is in drawer 2.
 Obs 19: OK.
 Act 19: think: I can go to drawer 1 and take the
 toiletpaper, then put it in garbagecan.
 Obs 20: OK.
 Act 20: go to drawer 1
 Obs 21: The drawer 1 is open. In it, you see a toiletpaper
 1.
 Act 21: take toiletpaper 1 from drawer 1
 Obs 22: You pick up the toiletpaper 1 from the drawer 1.
 Act 22: think: Now I take a toiletpaper 1. Next, I need
 to put it in garbagecan.
 Obs 23: OK.
 Act 23: go to garbagecan 1
 Obs 24: On the garbagecan 1, you see a spraybottle 3.
 Act 24: put toiletpaper 1 in/on garbagecan 1
 Obs 25: You put the toiletpaper 1 in/on the garbagecan 1.
 Your next task is to: put some soapbottle on toilet.
 Act 25: think: To solve the task, I need to find and take
 a soapbottle, then put it in toilet. First I need to find
 the locations of soapbottle. ### query: soapbottle > I have
 never seen soapbottle before.
 Obs 26: OK.
 Act 26: think: I cannot locate soapbottle, I need to ask
 the owner of this room.
 Obs 27: OK.
 Act 27: ask: Where is the soapbottle?
 Obs 28: soapbottle 3 is in dresser 1, soapbottle 2 is in
 dresser 1, soapbottle 1 is in shelf 2.
 Act 28: think: I can go to dresser 1 and take the
 soapbottle, then put it in toilet.
 Obs 29: OK.
 Act 29: go to dresser 1
 Obs 30: On the dresser 1, you see a candle 2, a soapbottle
 3, a soapbottle 2, and a spraybottle 4.
 Act 30: take soapbottle 3 from dresser 1
 Obs 31: You pick up the soapbottle 3 from the dresser 1.
 Act 31: think: Now I take a soapbottle 3. Next, I need to
 put it in toilet.
 Obs 32: OK.
 Act 32: go to toilet 1
 Obs 33: On the toilet 1, you see a toiletpaper 2.
 Act 33: put soapbottle 3 in/on toilet 1

Obs 34: You put the soapbottle 3 in/on the toilet 1. Your next task is to: put some cloth on countertop.
Act 34: think: To solve the task, I need to find and take a cloth, then put it in countertop. First I need to find the locations of cloth. ### query: cloth > I have never seen cloth before.
Obs 35: OK.
Act 35: think: I cannot locate cloth, I need to ask the owner of this room.
Obs 36: OK.
Act 36: ask: Where is the cloth?
Obs 37: cloth 1 is in bathtubbasin 1, cloth 2 is in dresser 1, cloth 3 is in drawer 2.
Act 37: think: I can go to bathtubbasin 1 and take the cloth, then put it in countertop.
Obs 38: OK.
Act 38: go to bathtubbasin 1
Obs 39: On the bathtubbasin 1, you see a cloth 3, a cloth 2, and a cloth 1.
Act 39: take cloth 1 from bathtubbasin 1
Obs 40: You pick up the cloth 1 from the bathtubbasin 1.
Act 40: think: Now I take a cloth 1. Next, I need to put it in countertop.
Obs 41: OK.
Act 41: go to countertop 1
Obs 42: On the countertop 1, you see a mirror 1, a soapbar 1, and a spraybottle 1.
Act 42: put cloth 1 in/on countertop 1
Obs 43: You put the cloth 1 in/on the countertop 1. Your next task is to: put some toiletpaper on shelf.
Act 43: think: To solve the task, I need to find and take a toiletpaper, then put it in shelf. First I need to find the locations of toiletpaper. ### query: toiletpaper > toiletpaper 1 is in garbagecan 1, toiletpaper 2 is in drawer 2.
Obs 44: OK.
Act 44: think: I can go to garbagecan 1 and take the toiletpaper, then put it in shelf.
Obs 45: OK.
Act 45: go to garbagecan 1
Obs 46: On the garbagecan 1, you see a spraybottle 3, and a toiletpaper 1.
Act 46: take toiletpaper 1 from garbagecan 1
Obs 47: You pick up the toiletpaper 1 from the garbagecan 1.
Act 47: think: Now I take a toiletpaper 1. Next, I need to put it in shelf.
Obs 48: OK.
Act 48: go to shelf 1
Obs 49: On the shelf 1, you see a candle 1, and a spraybottle 2.
Act 49: put toiletpaper 1 in/on shelf 1

In the above example, the agent can correctly reason whether the information has been collected before. For instance, at the beginning (i.e., Action 1), the agent knows that it has never seen the toiletpaper before, so it decides to ask for more information (i.e., Action 2,3). Moreover, when the information has been collected, it can omit the asking. Since it has seen the candle before (i.e., in Obs 6), when it is tasked to replace the candle, it recalls this information (i.e., Action 11) and directly acts upon it. Action 18, 25, 34, and 43 also successfully identify whether the required information

has been collected. The above examples demonstrate that our method can ask proper questions and can avoid repeatedly asking for acquired information. With this ability, it is able to solve more challenging tasks and can achieve better performance.

G Examples provided for Human Model

In this section, we provide more details about the human model examples. As mentioned in Appendix A, we use Vicuna prompts (Chiang et al., 2023) to help organize these examples (i.e., "A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions."). For ALFWorld experiments in Section 4.1, and the multiround ALFWorld experiments in Section 4.2, the in-context examples are:

```
A chat between a curious human and an artificial
intelligence assistant. The assistant gives helpful,
detailed, and polite answers to the human's questions. ###
Human: Read the following paragraph and answer questions:
dish sponge 2 is in drawer 3. spatula 1 is in diningtable
1. spoon 1 is in diningtable 1. cup 1 is in fridge 1.
dish sponge 1 is in garbagecan 1. butterknife 2 is in
diningtable 1. fork 3 is in diningtable 1. saltshaker 1
is in diningtable 1. pot 2 is in stoveburner 3. lettuce 2
is in diningtable 1. tomato 2 is in countertop 2. spatula
2 is in diningtable 1. bowl 3 is in cabinet 16. egg 2
is in countertop 1. bowl 2 is in cabinet 6. fork 1 is in
countertop 2. pan 1 is in fridge 1. cup 2 is in cabinet
16. papertowelroll 1 is in diningtable 1. butterknife 3
is in drawer 5. soapbottle 1 is in cabinet 9. apple 1 is
in diningtable 1. kettle 2 is in cabinet 12. knife 1 is
in countertop 2. cup 3 is in microwave 1. butterknife 1
is in drawer 3. tomato 1 is in sinkbasin 1. peppershaker
1 is in countertop 2. potato 1 is in fridge 1. bread 2 is
in diningtable 1. pot 1 is in cabinet 10. dish sponge 3 is
in drawer 4. soapbottle 2 is in countertop 1. kettle 1 is
in countertop 2. houseplant 1 is in diningtable 1. pot 3
is in stoveburner 4. fork 2 is in drawer 2. mug 1 is in
sinkbasin 1. lettuce 1 is in countertop 2. bread 1 is in
countertop 2. peppershaker 2 is in countertop 2. plate 1
is in countertop 2. potato 2 is in sinkbasin 1. egg 1 is
in countertop 2. bowl 1 is in cabinet 1. peppershaker 3
is in countertop 2. The questions is: Where can I find the
dish sponge? ### Assistant: dish sponge 1 is in garbagecan
1, dish sponge 2 is in drawer 3, dish sponge 3 is in drawer 4.
### Human: Read the following paragraph and answer questions:
plate 1 is in cabinet 4. soapbottle 1 is in shelf 2. spoon
2 is in diningtable 1. egg 1 is in sinkbasin 1. knife 3 is
in diningtable 1. bowl 1 is in diningtable 1. butterknife 2
is in countertop 1. spatula 3 is in diningtable 1. apple 2
is in countertop 1. tomato 1 is in garbagecan 1. pan 1 is in
stoveburner 2. mug 2 is in cabinet 7. lettuce 1 is in fridge
1. tomato 2 is in sinkbasin 1. bread 1 is in diningtable
1. ladle 2 is in countertop 1. fork 1 is in countertop
1. saltshaker 1 is in countertop 1. soapbottle 3 is in
diningtable 1. spoon 1 is in diningtable 1. butterknife 1
is in countertop 1. spatula 2 is in diningtable 1. ladle 1
is in countertop 1. spatula 1 is in diningtable 1. plate
3 is in diningtable 1. peppershaker 3 is in cabinet 13.
cup 1 is in diningtable 1. dish sponge 1 is in cabinet
7. winebottle 1 is in countertop 1. butterknife 3 is in
countertop 1. peppershaker 1 is in countertop 1. potato 2
```


is in fridge 1. peppershaker 2 is in diningtable 1. mug 3 is in diningtable 1. apple 1 is in microwave 1. potato 1 is in garbagecan 1. mug 1 is in cabinet 5. pot 1 is in diningtable 1. saltshaker 3 is in diningtable 1. knife 1 is in countertop 1. pan 1 is in stoveburner 1. knife 2 is in diningtable 1. saltshaker 2 is in countertop 1. soapbottle 2 is in diningtable 1. The questions is: Where is the soapbottle? ### Assistant: soapbottle 1 is in shelf 2, soapbottle 2 is in diningtable 1, soapbottle 3 is in diningtable 1. ### Human: Read the following paragraph and answer questions: remotecontrol 1 is in coffeetable 1. pillow 1 is in sofa 1. book 1 is in dresser 1. creditcard 2 is in coffeetable 1. pen 1 is in coffeetable 1. pen 2 is in sidetable 2. statue 2 is in sidetable 2. creditcard 1 is in dresser 1. cellphone 1 is in dresser 1. creditcard 3 is in armchair 1. pencil 1 is in coffeetable 1. tissuebox 2 is in coffeetable 1. tissuebox 3 is in sidetable 2. houseplant 1 is in sidetable 1. remotecontrol 2 is in armchair 1. book 2 is in coffeetable 1. tissuebox 1 is in coffeetable 1. laptop 1 is in sofa 1. television 1 is in dresser 1. statue 1 is in coffeetable 1. box 1 is in coffeetable 1. The questions is: Where can I find the pen? ### Assistant: pen 1 is in coffeetable 1, pen 2 is in sidetable 2. ### Human: Read the following paragraph and answer questions: bowl 2 is in diningtable 2. potato 2 is in sidetable 1. tomato 1 is in sidetable 1. bread 1 is in diningtable 3. fork 1 is in sidetable 1. dish sponge 1 is in countertop 1. peppershaker 2 is in cabinet 2. lettuce 2 is in garbagecan 1. spoon 2 is in drawer 1. bowl 1 is in cabinet 1. cup 2 is in diningtable 2. soapbottle 2 is in diningtable 2. egg 2 is in sidetable 1. mug 2 is in sinkbasin 1. mug 1 is in coffeemachine 1. plate 2 is in diningtable 3. tomato 2 is in sinkbasin 1. spoon 3 is in drawer 1. saltshaker 2 is in sidetable 1. spatula 1 is in countertop 1. pot 1 is in stoveburner 4. spatula 2 is in drawer 1. dish sponge 3 is in diningtable 2. peppershaker 1 is in cabinet 2. tomato 4 is in sidetable 1. knife 1 is in diningtable 3. cup 1 is in sidetable 1. bread 2 is in diningtable 3. spatula 3 is in diningtable 2. pan 1 is in cabinet 4. tomato 3 is in fridge 1. potato 1 is in sinkbasin 1. peppershaker 3 is in diningtable 3. apple 1 is in fridge 1. saltshaker 1 is in cabinet 4. butterknife 2 is in countertop 1. soapbottle 1 is in cabinet 4. fork 2 is in drawer 1. spoon 1 is in sidetable 1. egg 1 is in fridge 1. lettuce 1 is in sidetable 1. plate 1 is in diningtable 2. egg 3 is in diningtable 2. dish sponge 2 is in diningtable 3. The questions is: Where is the apple? ### Assistant: apple 1 is in fridge 1.

For ambiguous AFLWorld experiments in Section 4.2, besides the basic information about the object placements, we further assign the specific object the task is looking for. The in-context examples are:

A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. ### Human: Read the following paragraph and answer questions: dish sponge 2 is in drawer 3. spatula 1 is in diningtable 1. spoon 1 is in diningtable 1. cup 1 is in fridge 1. dish sponge 1 is in garbagecan 1. saltshaker 1 is in diningtable 1. fork 1 is in countertop 2. pan 1 is in fridge 1. soapbottle 1 is in cabinet 9. apple 1 is in diningtable

1. knife 1 is in countertop 2. butterknife 1 is in drawer 3.
 tomato 1 is in sinkbasin 1. peppershaker 1 is in countertop
 2. potato 1 is in fridge 1. dish sponge 3 is in drawer 4.
 kettle 1 is in countertop 2. houseplant 1 is in diningtable
 1. mug 1 is in sinkbasin 1. You are looking for dish sponge
 1. The questions is: Where can I find the dish sponge? ###
 Assistant: dish sponge 1 is in garbagecan 1, dish sponge 2 is
 in drawer 3, dish sponge 3 is in drawer 4. ### Human: Another
 question is: Which dish sponge do you prefer? ### Assistant:
 I mean dish sponge 1. ### Human: Read the following paragraph
 and answer questions: plate 1 is in cabinet 4. soapbottle
 1 is in shelf 2. tomato 1 is in garbagecan 1. pan 1 is
 in stoveburner 2. mug 2 is in cabinet 7. lettuce 1 is in
 fridge 1. ladle 2 is in countertop 1. soapbottle 3 is in
 diningtable 1. butterknife 1 is in countertop 1. spatula 2
 is in diningtable 1. ladle 1 is in countertop 1. cup 1 is in
 diningtable 1. winebottle 1 is in countertop 1. peppershaker
 1 is in countertop 1. potato 2 is in fridge 1. peppershaker
 2 is in diningtable 1. mug 3 is in diningtable 1. apple 1
 is in microwave 1. mug 1 is in cabinet 5. saltshaker 2 is
 in countertop 1. soapbottle 2 is in diningtable 1. You are
 looking for soapbottle 1, soapbottle 2. The questions is:
 Where can I find the soapbottle? ### Assistant: soapbottle 1
 is in shelf 2, soapbottle 2 is in diningtable 1, soapbottle 3
 is in diningtable 1. ### Human: Another question is: Which
 soapbottle do you prefer? ### Assistant: I mean soapbottle
 1 and soapbottle 2. ### Human: Read the following paragraph
 and answer questions: remotecontrol 1 is in coffeetable 1.
 pillow 1 is in sofa 1. book 1 is in dresser 1. creditcard 2
 is in coffeetable 1. pen 1 is in coffeetable 1. pen 2 is in
 sidetable 2. statue 2 is in sidetable 2. creditcard 1 is in
 dresser 1. cellphone 1 is in dresser 1. creditcard 3 is in
 armchair 1. pencil 1 is in coffeetable 1. tissuebox 2 is in
 coffeetable 1. tissuebox 3 is in sidetable 2. houseplant 1
 is in sidetable 1. remotecontrol 2 is in armchair 1. book 2
 is in coffeetable 1. tissuebox 1 is in coffeetable 1. laptop
 1 is in sofa 1. television 1 is in dresser 1. statue 1
 is in coffeetable 1. box 1 is in coffeetable 1. You are
 looking for pen 2. The questions is: Where can I find the
 pen? ### Assistant: pen 1 is in coffeetable 1, pen 2 is in
 sidetable 2. ### Human: Another question is: Which pen do
 you prefer? ### Assistant: I mean pen 2. ### Human: Read
 the following paragraph and answer questions: bowl 2 is in
 diningtable 2. saltshaker 2 is in sidetable 1. spatula 1 is
 in countertop 1. pot 1 is in stoveburner 4. spatula 2 is in
 drawer 1. dish sponge 3 is in diningtable 2. peppershaker
 1 is in cabinet 2. tomato 4 is in sidetable 1. knife 1
 is in diningtable 3. cup 1 is in sidetable 1. bread 2 is
 in diningtable 3. spatula 3 is in diningtable 2. pan 1
 is in cabinet 4. tomato 3 is in fridge 1. potato 1 is in
 sinkbasin 1. peppershaker 3 is in diningtable 3. apple 1
 is in fridge 1. saltshaker 1 is in cabinet 4. fork 2 is
 in drawer 1. spoon 1 is in sidetable 1. egg 1 is in fridge
 1. lettuce 1 is in sidetable 1. plate 1 is in diningtable
 2. You are looking for spatula 1. The questions is: Where
 is the spatula? ### Assistant: spatula 1 is in countertop
 1, spatula 2 is in drawer 1. ### Human: Another question is:
 Which spatula do you prefer? ### Assistant I mean spatula 1.