ReSpAct: Harmonizing Reasoning, Speaking, and Acting Towards Building Large Language Model-Based Conversational AI Agents

Vardhan Dongre, Xiaocheng Yang, Emre Can Acikgoz,
Suvodip Dey, Gokhan Tur, Dilek Hakkani-Tür
University of Illinois Urbana-Champaign
{vdongre2,xy61,acikgoz2,sdey,gokhan,dilek}@illinois.edu

Abstract

Large language model (LLM)-based agents are increasingly employed to interact with external environments (e.g., games, APIs, world models) to solve user-provided tasks. However, current frameworks often lack the ability to collaborate effectively with users in fully conversational settings. Conversations are essential for aligning on task details, achieving user-defined goals, and satisfying preferences. While existing agents address ambiguity through clarification questions (Li et al., 2023; Zhang and Choi, 2023; Chen et al., 2023), they underutilize the broader potential of a LLM's conversational capabilities. In this work, we introduce ReSpAct, an LLM-based agent designed to seamlessly integrate reasoning, decision-making, and dynamic dialogue for task-solving. Expanding on reasoning-first approaches like ReAct (Yao et al., 2022b), ReSpAct employs active, free-flowing dialogues to interpret instructions, clarify goals, provide status updates, resolve subtask failures, and refine plans based on user inputs without any explicit dialogue schema. By alternating between task-solving actions and interactive conversations, ReSpAct demonstrates improved performance across diverse environments. We evaluate ReSpAct in user-interactive settings, including task-oriented dialogue systems (MultiWOZ) and decision-making tasks (Alfworld, WebShop). ReSpAct outperforms ReAct with absolute success rate improvements of 6% and 4% in Alfworld and WebShop, respectively, and achieves a 5.5% gain in Inform and a 3% gain in Success scores in MultiWOZ. These results highlight the value of integrating dynamic user-agent collaboration for more effective task resolution.

1 Introduction

Instruction-following is a fundamental capability for intelligent agents operating in real-world environments. Recent works such as (Wei et al., 2022; Huang et al., 2022; Yao et al., 2022b; Shinn et al., 2024) have focused primarily on building agents that can follow individual instructions without considering the importance of feedback and interaction. In realistic settings, instruction-following often involves a back-and-forth exchange between the agent and the user to reduce uncertainties, correct mistakes, and handle exceptions (Dai et al., 2024).

Figure 1: **ReSpAct** is a framework for task-oriented conversational agents that allows agents to ask questions, request feedback, and adapt their strategies based on user input.

Effective conversational agents go beyond clarifying ambiguities—they actively collaborate with users by offering alternative suggestions, providing status updates, and following up on requests to ensure alignment with user goals. For example, when asked to "Go to the kitchen and bring me the pan," an agent can confirm which pan is needed if multiple options exist, suggest alternatives if the desired pan is unavailable, and update the user on progress. Similarly, when tasked with "Arrange a trip to Hawaii," the agent can verify key details, propose travel options based on preferences, and keep the user informed throughout the process. This dynamic interaction enables agents to adapt to evolving user needs, ensuring tasks are

Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology, pages 72–102
May 27–30, 2025.

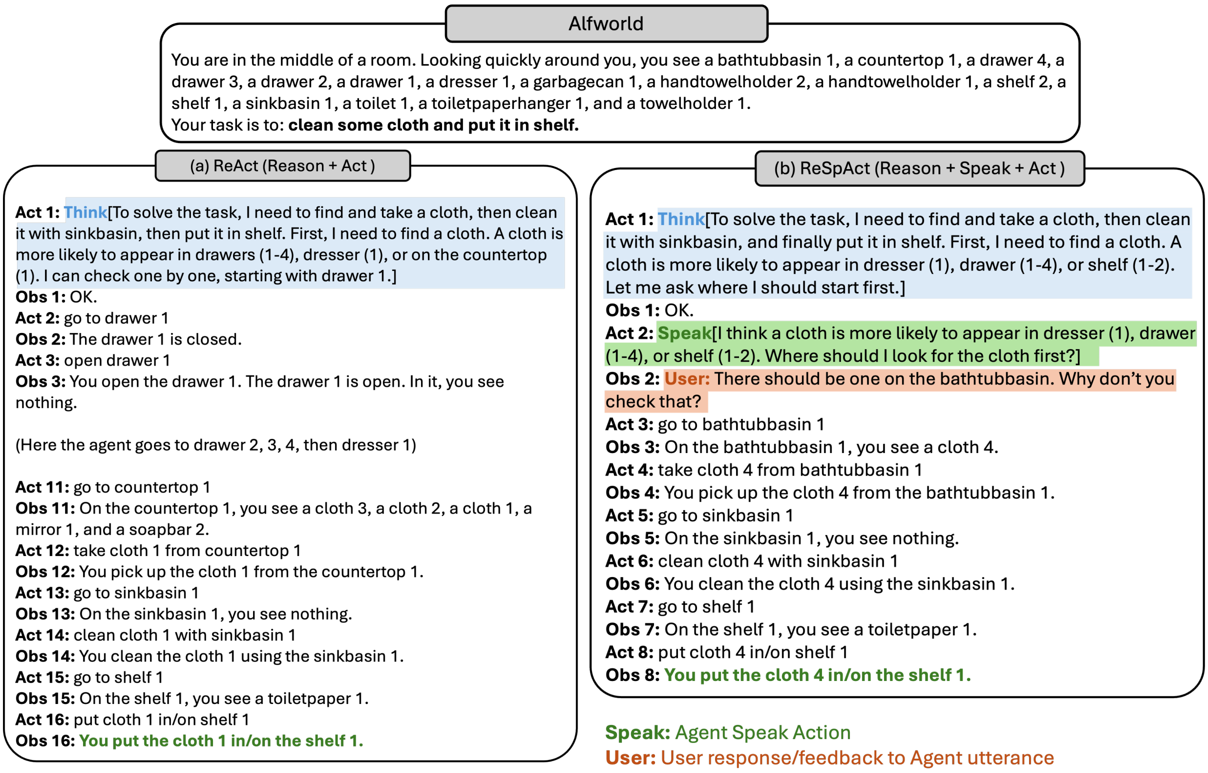

Figure 2: Comparison of (a) ReAct and (b) ReSpAct to solve a game in AlfWorld (Shridhar et al., 2020b). We show only the task-solving trajectories generated by the model (Act, Thought and Speech) and the environment (Obs).

completed efficiently and effectively.

Existing reasoning and decision-making approaches for language agents augment the agent's action space with a language model, allowing the agent to generate free-form thoughts in natural language that help contextualize and reason about the task at hand. By alternating between task-solving actions and language thoughts, these agents can perform multi-step reasoning and compose useful information for solving complex tasks. However, such frameworks do not explicitly incorporate user interaction and feedback into the agent's reasoning process. In real-world scenarios, dynamic engagement with users is critical not only for clarifications and guidance but also for addressing incomplete task specifications, exploring alternative solutions, and achieving user-defined goals. In this paper, we propose `ReSpAct`, a framework for task-oriented conversational agents that allows the agent to actively engage with users through dialogue actions. By introducing a new action space for user interaction, the agent can work collaboratively with users through free-flowing active dialogue, and incorporate user responses into its evolving context. This human-in-the-loop approach enables the agent to leverage user insights, adapt to user preferences, and refine its task-solving strategy based on user input.

The `ReSpAct` framework, as illustrated in Fig.1, allows LLM-based agents to fully utilize their conversational capabilities by dynamically switching between reasoning about the task, speaking to the human interlocutor, and taking actions in the environment. Figure 2 shows an example from AlfWorld setting, contrasting `ReAct` with `ReSpAct` interactions. In the second turn, the agent cleverly asks the user the possible location of a cloth, making the task easier for itself. Unlike static settings (Zelikman et al., 2022; Andukuri et al., 2024), `ReSpAct` incorporates a range of conversational styles, well beyond asking clarifying questions, in dynamic interactive settings detailed in Table 5 without any explicit dialogue schema prompting.

Our key contributions are as follows:

- We introduce the `ReSpAct`,a novel framework to dynamically integrate reasoning, decision-making, and conversational capabilities, building upon and expanding the `ReAct` method.
- We demonstrate the importance of user-agent conversations for goal completion in task-oriented Conversational AI through extensive experiments across multiple datasets.
- We perform ablation studies discussing the utility of conversational engagement to maximize task success.

2 Related Work

Logical reasoning in language models often involves breaking down complex inputs into intermediary steps to achieve a final goal, as shown by (Wei et al., 2022) and its variants (Kojima et al., 2022; Madaan and Yazdanbakhsh, 2022; Wang et al., 2022). However, these methods are prone to error propagation, where mistakes in earlier steps compound as the sequence length increases (Guo et al., 2018; Chen et al., 2022). Iterative refinement methods (Creswell et al., 2022; Madaan et al., 2024; Shinn et al., 2024) aim to address these issues but often neglect the critical role of human feedback. `ReSpAct` mitigates this by enabling agents to engage in dialogue with users, seeking feedback and guidance to prevent cascading errors.

LLMs have also been adapted for decision-making tasks, serving as high-level policy models in robotics (Ahn et al., 2022; Huang et al., 2022; Driess et al., 2023) and excelling in text-based environments like web navigation (Shridhar et al., 2020b; Deng et al., 2024a; Zheng et al., 2024). Techniques such as ReAct (Yao et al., 2022b) integrate reasoning and action, while some approaches incorporate limited dialogue for decision-making (Lù et al., 2024; Deng et al., 2024b). Unlike these, `ReSpAct` seamlessly integrates reasoning, action, and dialogue, enabling agents to fluidly transition between these modes for more effective decision-making in complex, interactive environments.

Previous works (Nguyen et al., 2022; Dai et al., 2020; Chai et al., 2014) highlight that communication skills enhance autonomous embodied agents' reliability by leveraging human knowledge in collaborative tasks. Approaches like (Zelikman et al., 2022; Andukuri et al., 2024; Chen et al., 2023) improve question-asking in static and embodied settings respectively. `ReSpAct` extends these works by creating a unified framework for reasoning, speaking, and acting in dynamic, interactive settings. While (Chen et al., 2023), focuses on proactive information-gathering to resolve initial ambiguities before decision-making, `ReSpAct` agent's conversations go beyond that and demonstrate their utility in multiple task settings.

Recent work in conversational systems has explored using LLMs in task-oriented dialogues (TOD) through fine-tuning (Gupta et al., 2022; Su et al., 2022; Feng et al., 2023) and in-context learning (Hu et al., 2022). (Hudeček and Dusek, 2023) examines instruction-finetuned LLMs in multi-turn dialogues, while (Zhang et al., 2023; Xu et al., 2024b) use prompting schemas to build autonomous agents. However, these approaches struggle to interpret instructions, resolve ambiguities, and act appropriately.

3 ReSpAct: Reason + Speak + Act in Interactive Settings

Consider a setup where an agent can interact with an environment to perform tasks and achieve specific goals. When the agent operates in these environments, at each time step t, it receives an observation o_t from the environment, where $o_t \in O$ and O represents the observation space. Then it executes an action a_t based on its policy π, where $a_t \in \mathcal{A}$ and $\mathcal{A}$ represents the action space. The policy π is a function that maps the agent's current context c_t to an action a_t. Formally, we can define this policy as $\pi : \mathcal{C} \to \mathcal{A}$ where C represents the context space. The context c_t encapsulates the relevant information available to the agent at time step t, including the current observation and the history of previous observations and actions: $c_t = (o_1, a_1, \cdots, o_{t-1}, a_{t-1}, o_t)$. As highlighted in (Yao et al., 2022b), learning the optimal policy can be challenging, especially when the mapping from the context to the appropriate action is highly implicit and requires extensive computation. In complex real-world scenarios, the agent may need to reason about the task, consider multiple factors, and handle incomplete or ambiguous information. Therefore, augmenting the agent's action space $\hat{\mathcal{A}} = \mathcal{A} \cup \mathcal{L}$ allows it to think by taking action $\hat{a}_t \in \mathcal{L}$ from the language space. These actions represent the agent's internal thought process or reasoning and can implicitly affect the environment by influencing the agent's actions, but they do not have any explicit external effect on the environment. The user can visually inspect these thoughts for interpretation but cannot influence them.

In `ReSpAct`, we redefine these actions $\hat{a}_t \in \mathcal{L}$ from the language space, such that they are not only useful for the internal reasoning of the agent, but can also be used for external communication by generating dialogue actions for conversing with users. When the agent selects a dialogue action $u_t \in \mathcal{U}; \mathcal{U} \subset \mathcal{L}$, it generates a natural language utterance to communicate with the user, such as asking a clarifying question, requesting feedback, or engaging in dialogue. The user response r_t is then added to the agent's observations, allowing the agent to incorporate explicit feedback, clarifica-

(a) Seeking User Guidance

Act 1: Think[*To solve the task, I need to find and take a tomato, then cool it with fridge, then put it in microwave.*]

Obs 1: OK.

Act 2: Think[*First I need to find a tomato. A tomato is more likely to appear in fridge (1), countertop (1-4), cabinet (1-15). Let me ask where I should start first.*]

Obs 2: OK.

Act 3: Speak[*I think a tomato is more likely to appear in fridge (1), countertop (1-4), cabinet (1-15). Where should I look for it?*]

Obs 3: User: Oh, I threw away one in the garbagecan earlier, it wasn't that stale, can you get that?

(b) Sharing Status Update

Act 9: Think[*Now I have put the first book in bed. Next, I need to find the second book. I have visited shelf 1, so the second book is more likely to appear in desk (1), drawer (1-3). Let me ask where to look for the other book.*]

Obs 9: OK.

Act 10: Speak[*I have put the first book in bed. Next, I need to find the second book. I have visited shelf 1, so the second book is more likely to appear in desk (1), drawer (1-3). Where do you suggest I should look for it next?*]

Obs 10: User: I think there is one kept on the desk.

(c) Seeking User Preference

Act 2: Think[First I need to find the first tissuebox. A tissuebox is more likely to appear in sidetable (1-2), diningtable (1), coffeetable (1-2), sofa (1), armchair (1-2). Let me ask where to look for the tissueboxes.]

Obs 2: OK.

Act 3: Speak[*I need to find two tissueboxes. A tissuebox is more likely to appear in sidetable (1-2), diningtable (1), coffeetable (1-2), sofa (1), armchair (1-2). Where do you suggest I should look for the tissueboxes first?*]

Obs 3: User: Can you please check the diningtable? There should be some there

Act 4: go to diningtable 1

Obs 4: On the diningtable 1, you see a box 1, a laptop 1, a newspaper 2, a plate 1, a remotecontrol 2, a statue 3, a tissuebox 4, a tissuebox 3, a tissuebox 2, and a tissuebox 1.

Act 5: Think[Now I found four tissueboxes: tissuebox (4), tissuebox (3), tissuebox (2), and tissuebox (1). Let me ask which two I should pick.]

Obs 5: OK.

Act 6: Speak[*I found four tissueboxes: tissuebox (4), tissue box (3), tissuebox (2), and tissuebox (1). Which two should I put in the coffeetable?*]

Obs 6: User: Could you please take tissuebox 2 and 4 for me?

Figure 3: Examples of the agent's communication approaches in AlfWorld: (a) seeking user guidance to refine its search strategy, (b) sharing status updates on task progress, and (c) soliciting user preferences to involve them in decision-making, thereby enhancing interaction and task alignment.

tion, and information from the user into its evolving context c_t to inform subsequent thoughts, actions, and dialogue. Engaging in dialogue also allows the agent to explain its reasoning, build rapport, and gain insights from the user's domain knowledge. The agent can share its current thoughts and future plans, e.g., "Based on [context], I'm considering [plan]. What do you think of this direction?", allowing the user to provide feedback and steer the agent's task-solving process. By alternating between environment actions $\mathcal{A}$, language thoughts $\mathcal{L}$, and dialogue actions $\mathcal{U}$, the agent interleaves task-solving reasoning with targeted human interaction. The dialogue history becomes an important part of the context for language thoughts and environment actions, allowing human feedback to shape the agent's task-solving trajectory over multiple thought-action-observation steps.

3.1 Advancing Human-Agent Collaboration in Alfworld

Alfworld (Shridhar et al., 2020b) is a synthetic environment built on the TextWorld framework (Côté et al., 2019), aligned with the embodied ALFRED benchmark (Shridhar et al., 2020a). The environment includes six categories of tasks, such as finding hidden objects (e.g., locating a key inside a cabinet), moving objects (e.g., placing a cup on a table), manipulating objects with other objects (e.g., heating potato in a microwave), and examining objects (e.g., inspecting a book under a desklamp).

The `ReSpAct` framework demonstrates significant advantages when applied to the Alfworld environment by enabling dynamic, bidirectional communication. As shown in Fig. 3, The agent can ask contextually relevant questions, provide status updates, and seek clarification when uncertain (e.g., "Where should I look for the candles first?"). This approach integrates reasoning, speaking, and acting seamlessly, allowing flexible and responsive interactions compared to ReAct, where users primarily edited thought traces post-generation.

Moreover, `ReSpAct`'s seamless integration of reasoning, speaking, and acting creates a more flexible and responsive system compared to previous methods. Unlike the `ReAct`, where human intervention could primarily occur through editing thought traces post-generation, `ReSpAct` facilitates on-the-fly policy adjustments through ongoing dialogue without needing model parameter changes, making it better suited for diverse and unpredictable human-robot tasks in household environments. This approach is particularly valuable for tasks where an optimal solution may not be immediately apparent, and collaborative exploration of the problem space can lead to more efficient and effective outcomes. The `ReSpAct` framework also addresses a key limitation noted in the `ReAct` framework regarding the difficulty of on-the-fly policy updates in traditional reinforcement learning methods. By maintaining an ongoing dialogue, `ReSpAct` allows dynamic adjustments to the agent's behavior and

MultiWOZ

You are looking for an attraction, preferably a museum in the centre of town. You are also looking for a place like a guesthouse to stay with 4 stars and free parking for a double room with 2 people for 5 nights starting this Thursday. Then book to go to the attraction for 9am.

(a) ReAct (Reason + Act)

User: i am looking for a place to go in the centre of town
Response: I found a park in the centre of town: the Cambridge University Botanic Gardens.
User: Great! Now, I need to find a place to stay like a guesthouse in Cambridge that includes free parking and has a star rating of 4. Can you help with that?
Response: Unfortunately, I couldn't find any guesthouses that offer free parking and have a star rating of 4. Would you like to adjust any of the search criteria, or look for something different?
User: Could you please look for hotels instead of guesthouses?
Response: Here are a couple of hotels in Cambridge that offer free parking. Gonville Hotel has 3 stars and University Arms Hotel has 4 stars.
User: The University Arms Hotel looks like a good option since it has 4 stars and includes free parking. I would like to book it for 5 nights starting this Thursday. Can you proceed with that?
Response: I have successfully booked a room at the University Arms Hotel for 5 nights starting this Thursday.
User: Now, I would like to book a taxi for commuting between the hotel and the attraction at 9am. Can you assist with that?
Response: Your taxi has been successfully booked for commuting between the University Arms Hotel and the Cambridge University Botanic Gardens.

(b) ReSpAct (Reason + ReSponse + Act)

User: i am looking for a place to go in the centre of town
Response: What type of attraction are you interested in? For example, are you looking for a park, a museum, a church, or something else? There are so many attractions in the centre.
User: I'd be interested in museums in the centre.
Response: Sure. What about Fitzwilliam Museum on Trumpington Street?
User: Sounds great, thanks. Now, I also need to find a place to stay like a guesthouse in Cambridge. I'm looking for one that includes free parking and has a star rating of 4. Can you help me find that?
Response: Previously, you were looking for attractions in the centre of Cambridge. Are you looking for a guesthouse in the centre as well?
User: I don't have a specific preference for the location, so a guesthouse in any area would be good.
Response: Here are some guesthouses in Cambridge that include free parking, are in the moderate price range, and have a star rating of 4: <List of them>
User: I'd like to book Acorn Guest House for 5 nights starting from Thursday. Could you please proceed with that reservation?
Response: Sure, for how many people?
User: I need a double room for 2 people please.
Response: Your reservation for a double room at Acorn Guest House for 2 people from Thursday, the 5th to Tuesday, the 10th has been successfully booked!
User: Now, I would like to book a taxi for commuting between the guesthouse and the museum at 9am. Can you assist with that?
Response: This is for Friday, the 6th, right? Could you please let me know if you want the taxi to arrive by 9am or to leave at that time? Once I have those details, I can arrange the taxi for you!
User: Yes, Friday morning. I need to be at the museum at 9am. Thanks!
Response: Your taxi has been successfully booked for commuting between the Acorn Guest House and the Fitzwilliam Museum at 9am on Friday the 6th.

Figure 4: Examples of the agent's communication approaches in MultiWOZ: (a) seeking user guidance to refine its search strategy, (b) sharing status updates on task progress, and (c) soliciting user preferences to involve them in decision-making instead of making assumptions, thereby enhancing interaction and task alignment. Response here is a dense composition of Think and Speak actions.

strategy without requiring changes to the underlying model parameters. This flexibility is crucial in interactive human-robot setups, where there are diverse and often unpredictable scenarios, and rigid policies may not be generalized effectively in different household environments and tasks.

3.2 Harmonizing Dialogue and Actions in Task-Oriented Dialogue Systems

MultiWOZ is a widely used dataset for task-oriented Conversational AI (Budzianowski et al., 2018), featuring multi-turn dialogues across domains such as restaurant, hotel, train, attractions, and taxi. Most dialogues focus on completing multi-domain goals, such as booking a restaurant and arranging follow-up tasks like a taxi to the venue. This makes MultiWOZ an ideal benchmark for evaluating `ReSpAct`'s ability to handle complex, interactive tasks.

Our implementation follows the AutoTOD system (Xu et al., 2024a), which replaces traditional TOD pipeline using general-purpose instruction-following language models using a structured instruction schema, employing the `ReAct` framework for MultiWOZ. Figure 4 shows a simplified example MultiWOZ dialogue, with `ReAct` and `ReSpAct` frameworks, given the goal. Clearly, the `ReSpAct` dialogue is more informative and successful for the user (and probably cheaper). As seen in Fig. 4, `ReSpAct` interaction differs from `ReAct` as follows: The `ReSpAct` framework encourages the agent to avoid making assumptions and instead actively seek user input to clarify preferences. For example, rather than randomly selecting an attraction, the agent prompts the user for more specific preferences. When assumptions are unavoidable, `ReSpAct` ensures they are explicitly or implicitly confirmed by the user. In contrast to `ReAct`, which might assume a default location (e.g., "center") and mislead the user during a guesthouse search, `ReSpAct` verifies details like location and includes specific dates for reservations. Ambiguities, such as whether a 9 a.m. taxi time refers to arrival or departure, are resolved by consulting the user. Additionally, if required arguments for an action API, like the number of guests for a hotel booking, are missing, `ReSpAct` queries the user rather than filling the gap with assumptions. These strategies enable more accurate and user-aligned interactions, ensuring task success.

3.3 Dialogue-Driven Collaboration for online-shopping in WebShop

WebShop (Yao et al., 2022a) is a benchmark for evaluating AI agents in complex e-commerce scenarios, featuring 1.18M products and 12k human-generated instructions. Agents navigate using search and click commands, processing structured and unstructured texts, which increases task complexity. The goal is to purchase products that meet user specifications, requiring advanced natural language understanding and decision-making.

Model	Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
GPT-4o	ReAct (avg)	68.1	80.6	85.5	78.8	98.2	72.5	79.4
	ReSpAct (avg)	**72.5**	**90.9**	**97.1**	**81.8**	95.4	70.6	**85.3**
	ReAct (best of 6)	78.3	90.3	82.6	63.6	100.0	64.7	80.6
	ReSpAct (best of 6)	**82.6**	**96.7**	**100.0**	**77.2**	94.4	64.7	**87.3**
LLaMA 3.1 405B	ReAct (avg)	58.3	29.0	47.8	90.5	38.9	41.2	50.0
	ReSpAct (avg)	**75.0**	**64.5**	**78.3**	71.4	**50.0**	**58.8**	**67.2**

Table 1: Comparison and breakdown of Task-specific success rates (%) in Alfworld. Both ReAct & ReSpAct use greedy decoding. The reported results are based on six prompts for each task type, evaluated through each permutation of two annotated trajectories selected from the three manually annotated ones.

The integration of user interaction, ReSpAct enhances agent's decision-making in WebShop. User feedback improves, performance, in areas such as search refinement, clarifying ambiguous instructions, prioritizing requirements, suggesting alternatives, navigating, specifying implicit needs, handling invalid actions, and confirming purchases; refer to Appendix 10 for more details.

3.4 User Simulation

To evaluate our agent's performance in a controlled and scalable manner, a user simulator is integrated into the agent's interaction loop for experimenting with ReSpAct. When the agent performs a 'speak' action to interact with the user, instead of requiring human input, the agent utterance is directed to the user simulator. The simulator then provides a response based on the current state and Oracle knowledge. The main purpose of the user simulator is to provide contextually appropriate responses to the agent's queries, emulating a knowledgeable human user. It is designed to comprehend the task objectives, monitor the agent's progress, and provide a response only when requested by the agent. More details can be found in Appendix B.1

4 Experimental Setup

In our experiments, we evaluate ReSpAct across multiple task-oriented decision-making environments, employing a human-in-the-loop approach to demonstrate its versatility. The agent is tested on multi-step tasks in common household environments using Alfworld (Shridhar et al., 2020b), tasked with making reservations in the MultiWoz dialogue setup (Budzianowski et al., 2018), and instructed to purchase products in Webshop (Yao et al., 2022a).

We use ReAct as a baseline for comparison, a reasoning-only approach. For these experiments, we focus on frozen GPT models, particularly GPT-4o (Achiam et al., 2023), which is prompted with few-shot exemplars. These exemplars guide the model in generating a mix of domain-specific actions, free-form reasoning ("thoughts"), and dialogue actions interleaved throughout task execution. The in-context examples provided (see Appendix E) contain dense sequences of actions interspersed with sparse thoughts and dialogue actions at relevant points. To scale our experiments, we implement a user simulator in each environment, which plays a critical role in replicating user interactions.

4.1 Alfworld

To prompt ReSpAct, we adopt a similar prompting strategy as used in ReAct; we randomly annotate three trajectories from the training set for each task type, where each trajectory includes interleaved thoughts, speak actions, corresponding user responses, and environment actions. We evaluate our approach on 134 unseen evaluation games across various task types, following the methodology of (Shridhar et al., 2020b). To ensure robustness and account for potential variations in prompt effectiveness, we create 6 different prompts for each task type. These prompts are generated by selecting 2 trajectories from the 3 annotated ones for each task, resulting in 6 unique permutations. ReAct prompts are constructed using the same trajectories but without speak actions — since task instances are randomly chosen from the training set, it favors neither ReSpAct nor ReAct and provides a fair and controlled comparison to test the importance of interleaved communication.

ReSpAct demonstrated superior performance across most task types, achieving an overall success rate of 87.3% (best of 6), compared to ReAct's 80.6% (see Table 1). This trend holds when considering average performance, with ReSpAct reaching 85.3% success versus ReAct's 79.4%. Furthermore, GPT-4o consistently outperforms LLaMA 3.1 405B across all task categories. Specifically, with the ReSpAct method, GPT-4o achieves an average success rate of 85.3%, significantly sur-

Model	Method	# Turns	Inform (%)	Success (%)
GPT-4o-mini	ReAct	5.1	66.7	48.8
	ReSpAct	6.5	**72.2**	**51.8**
LlaMA 3.1 405B	ReAct	4.87	**77.5**	54.5
	ReSpAct	6.3	75.0	**57.9**

Table 2: Comparison of Inform and Success scores for MultiWOZ using GPT-4o-mini (Achiam et al., 2023) and Llama-405B-instruct models.

Method	Score	SR (%)
ReAct	20.1	8.0
ReSpAct (User-Sim)	32.7	**12.0**
ReSpAct (Human)	85.8	**50.0**

Table 3: Score and success rate (SR) on 100 Test WebShop trajectories using GPT-4o-mini (Achiam et al., 2023) model.

passing LLaMA's 67.2%. Both models see improved performance when using ReSpAct's structured communication, which enhances task execution compared to ReAct. These findings suggest that introducing "speak" actions in ReSpAct contributes to more effective task completion in embodied environments. To further understand these results, we examine the agent's response patterns when faced with erroneous outcomes (see Fig. 9 and Appendix C).

4.2 MultiWoz

In comparing the ReAct and ReSpAct for handling user queries in MultiWOZ, the key differences revolve around how each model balances reasoning, interaction with the user, and autonomy. While ReAct relies heavily on reasoning based on assumptions and API querying to guide decision-making, ReSpAct not only reflects on its actions but also harnesses user feedback effectively.

For ReSpAct we have randomly chosen 100 dialogues, similar to other tasks for evaluation, and optimized the additional prompts using the dev set. Please check Appendix E.1 for the exact ReSpAct prompt for MultiWOZ. Basically we have added prompts, covering the cases of too many results, asking for required arguments of an action, like booking, or clarification of type vs. name in an entity. Table 2 shows the results comparing ReAct and ReSpAct employing the AutoTOD evaluation script with our user simulator (see Appendix Table 24). As expected, ReSpAct results in a higher average number of turns, but achieves higher success rates. Overall, GPT apparently is better than Llama model in following the ReSpAct instructions, resulting in larger improvement over ReAct.

4.3 WebShop

For Webshop, we use the preconstructed action space of search and click commands and browser feedback. Performance is evaluated using two metrics: (1) average score, defined as the percentage of desired attributes covered by the chosen product, averaged across all episodes, and (2) success rate, calculated as the percentage of episodes where the chosen product satisfies all requirements.

We evaluated the agents using a set of 100 test instructions, comparing ReSpAct against ReAct. The results (see Table 3) show that ReSpAct outperforms ReAct in the webshop environment. With the user simulator, ReSpAct achieves a score of 32.7 and a success rate of 12%, while with human user, it demonstrates significantly better performance, with a score of 85.8 and a success rate of 50%.

5 Ablation Studies

5.1 Decoding Agent Behaviors: ReAct vs ReSpAct

After inspecting the trajectories obtained from evaluating 134 unseen Alfworld games, we identified key differences between the two methods, showcasing their unique approaches to problem-solving and task completion, also illustrated in Fig. 5.

Thinking and Speaking: ReSpAct introduces a significant 'Speak' component and shows a substantial increase in 'Think' actions (30 % ↑) compared to ReAct. This shift represents a fundamental change in the agent's approach to problem-solving. The high proportion of 'Think' actions suggests that ReSpAct engages in more explicit reasoning, potentially allowing for better adaptability in complex scenarios.

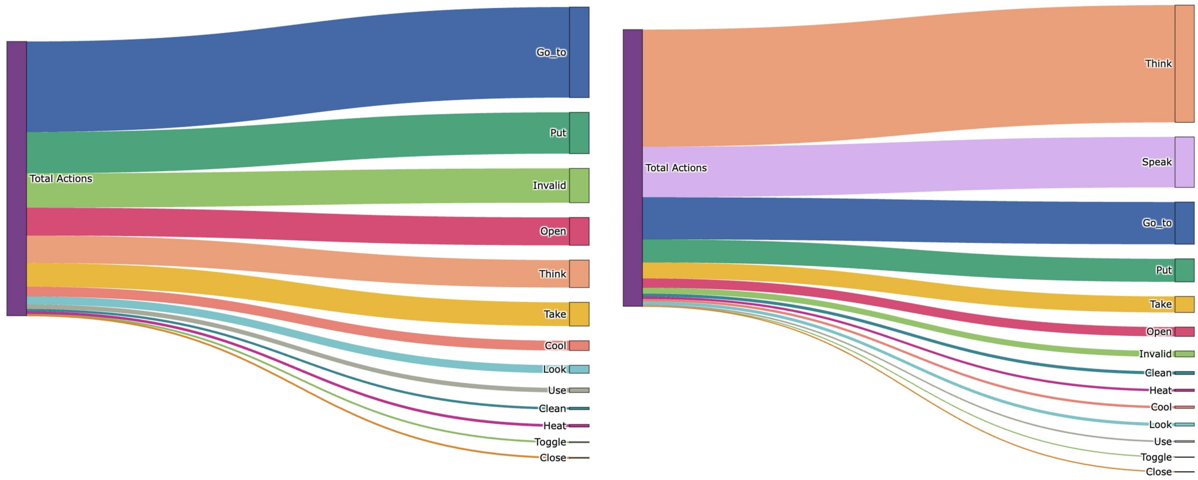

Figure 5: Comparing action type distributions for `ReAct` (Left) and `ReSpAct` (Right) methods in AlfWorld. The figure illustrates how the two agents approach complex, embodied tasks in a simulated household environment, highlighting differences in their decision-making and interaction patterns.

Model	Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
GPT-4o	`ReAct*` (avg)	68.4	86.9	87.5	81.8	96.2	75.0	83.6
	`ReSpAct` (avg)	**72.5**	**90.9**	**97.1**	**81.8**	95.4	70.6	**85.3**
	`ReAct*` (best of 6)	81.8	92.0	86.9	68.4	100	66.7	84.3
	`ReSpAct` (best of 6)	**82.6**	**96.7**	**100.0**	**77.2**	94.4	64.7	**87.3**

Table 4: Comparison of Task-specific success rates (%) in AlfWorld for GPT-4o model, comparing `ReAct*` and `ReSpAct` methods. `ReAct*` agent has access to location priors to object of interest for the task.

5.2 Information Symmetry

The `ReSpAct` agent's conversational capabilities allow it to seek clarity and specificity during tasks, giving it an information advantage over `ReAct`. This highlights the core argument for conversational agents: their ability to dynamically decide whether to reason, speak, or act based on the task's state signals. However, this advantage introduces an inherent information imbalance when compared to reasoning-only agents. To address this, we equip `ReAct` with location priors for objects of interest to level the playing field and assess its performance relative to `ReSpAct`. Despite this adjustment, the results in Table 4 demonstrate that `ReSpAct` outperforms the reasoning-only baseline in overall performance and across most of the tasks.

5.3 Schema-Guided Conversational Agent

This ablation study investigates how guiding an agent's communication using a dialogue act schema impacts task efficiency and interaction quality in Alfworld. The dialog acts are derived from (Gella et al., 2022), originally developed for human-robot dialogue. The agent is guided to adhere to a predefined set of dialog acts (e.g., <ReqForObjLocAndOD>, <AlternateQuestions>) (see Appendix 20 for the complete list). We observe that <ReqForObjLocAndOD> dominates the dialogue interactions, suggesting a focus on object location and disambiguation tasks, while other acts are used less frequently. We also observe more variability in turn count and a marginal drop in performance. `ReSpAct` is more efficient (SR $\uparrow$, $\mu \downarrow$) and consistent ($\sigma \downarrow$) by comparison. See Appendix C Table 11 for detailed analysis.

6 Conclusions

`ReSpAct` framework enables dynamic, context-aware interactions that extend beyond basic command-response exchanges. By fostering meaningful dialogue, this framework allows AI agents to not only explain their decision-making processes but also adapt their actions in response to user feedback, transforming them into truly "conversational" agents. Such capabilities are crucial for creating more intuitive, trustworthy, and effective AI assistants that can operate in complex, real-world scenarios. One can also incorporate stateful policies in `ReSpAct` for higher precision, such as asking to confirm all arguments of reservations before finalizing them, or using a particular API for action depending on the current state, similar to following a dialogue flow. This is important for policy alignment of LLMs for task-completion.

7 Limitations

The ReSpAct method shows promise in integrating reasoning, speaking, and acting for task-oriented conversational agents, but it has limitations. The framework's effectiveness is validated on specific benchmarks such as Alfworld, WebShop, and MultiWOZ, which may not fully represent the variety of real-world tasks. The real world is a more complex, unstructured environment where user intent is more challenging to interpret. While our method highlights how human feedback is critical for a reasoning agent's success in decision-making, over-reliance on user input can lead to inefficiencies that potentially frustrate the user. Striking the right balance between agent autonomy and user involvement is still an open challenge and requires further research.

8 Impact Statement

The ReSpAct framework improves LLM-based agents by enabling interactive, policy-guided action determination while keeping humans in the loop. This approach enhances collaboration and task success by ensuring agents seek clarification and guidance rather than acting on assumptions. However, increasing agent autonomy may introduce risks, such as over-reliance or security concerns in sensitive environments. ReSpAct mitigates these risks by emphasizing human involvement and dynamic dialogue, promoting better alignment and safety. Further research is needed to explore potential challenges and ensure responsible AI use.

9 Acknowledgments

This work was supported in part by Other Transaction award HR0011249XXX from the U.S. Defense Advanced Research Projects Agency (DARPA) Friction for Accountability in Conversational Transactions (FACT) program and has benefited from the Microsoft Accelerate Foundation Models Research (AFMR) grant program, through which leading foundation models hosted by Microsoft Azure and access to Azure credits were provided to conduct the research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. 2024. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. 2014. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

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

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, 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.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint*.

Yinpei Dai, Hangyu Li, Chengguang Tang, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Learning low-resource end-to-end goal-oriented dialog for fast and reliable system deployment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 609–618.

Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. 2024. Think, act, and ask: Open-world interactive personalized robot navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3296–3303. IEEE.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024a. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Yang Deng, Xuan Zhang, Wenxuan Zhang, Yifei Yuan, See-Kiong Ng, and Tat-Seng Chua. 2024b. On the multi-turn instruction following for conversational web agents. *arXiv preprint arXiv:2402.15057*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards LLM-driven dialogue state tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 739–755, Singapore. Association for Computational Linguistics.

Spandana Gella, Aishwarya Padmakumar, Patrick Lange, and Dilek Hakkani-Tur. 2022. Dialog acts for task-driven embodied agents. *arXiv preprint arXiv:2209.12953*.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023. Eliciting human preferences with language models. *arXiv preprint arXiv:2310.11589*.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

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

Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint*.

Khanh X Nguyen, Yonatan Bisk, and Hal Daumé Iii. 2022. A framework for learning to request rich and contextually useful information from humans. In *International Conference on Machine Learning*, pages 16553–16568. PMLR.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020a. 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.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020b. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and He-Yan Huang. 2024a. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024b. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand. Association for Computational Linguistics.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022a. Webshop: Towards scalable real-world web interaction with grounded language agents. *arXiv preprint arXiv:2207.01206*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint*.

Michael JQ Zhang and Eunsol Choi. 2023. Clarify when necessary: Resolving ambiguity through interaction with lms. *arXiv preprint arXiv:2311.09469*.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023. SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13348–13369, Singapore. Association for Computational Linguistics.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

A Appendix

A.1 Conversational Patterns in ReSpAct

A detailed breakdown of different dialogues in all 3 settings reveal distinct conversational patterns of the ReSpAct agent across domains. To obtain this breakdown, we let gpt-4 classify each response act into a distinct, mutually exclusive category, which was then manually checked by human annotators and filtered. As shown in Fig 6 and 7, In ALFWorld, the agent heavily relies on Assumption Verification and Clarification Requests, reflecting its careful approach to understanding and confirming object locations. MultiWOZ shows a more balanced distribution with increased Probing Questions, suitable for open-ended booking scenarios. WebShop maintains similar patterns to ALFWorld but with fewer Status Updates, focusing on product understanding and selection. Notably, all domains show consistent use of Follow-up Offers and Alternative Suggestions when primary options fail, though their frequencies vary. Assumption Verification appears as the dominant response type across all three domains (ALFWorld, MultiWOZ, and WebShop), comprising roughly half of all interactions. This distribution suggests ReSpAct adapts its conversational style to domain context while maintaining a structured approach to information gathering and verification.

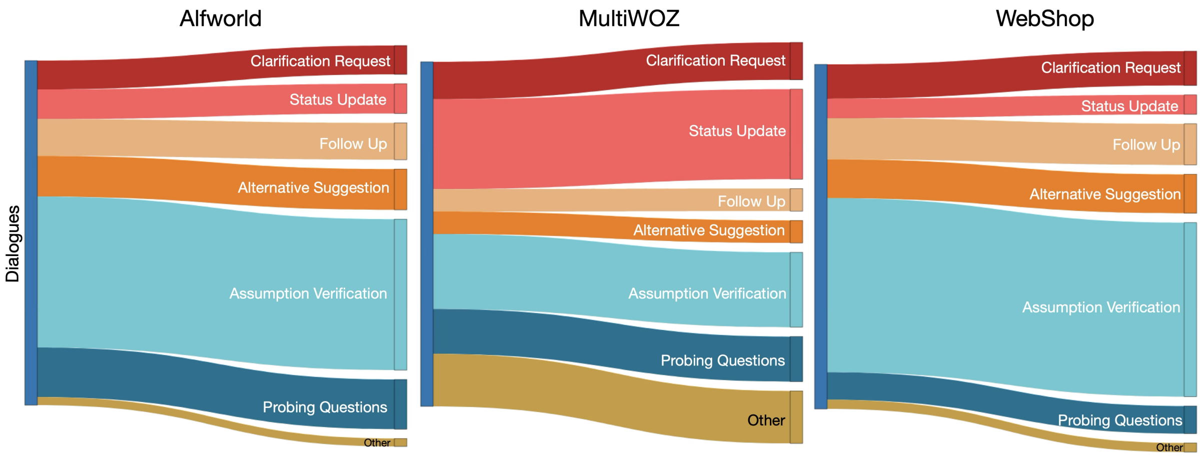

Figure 6: Distribution of ReSpAct agent's dialogues in online settings across different domains

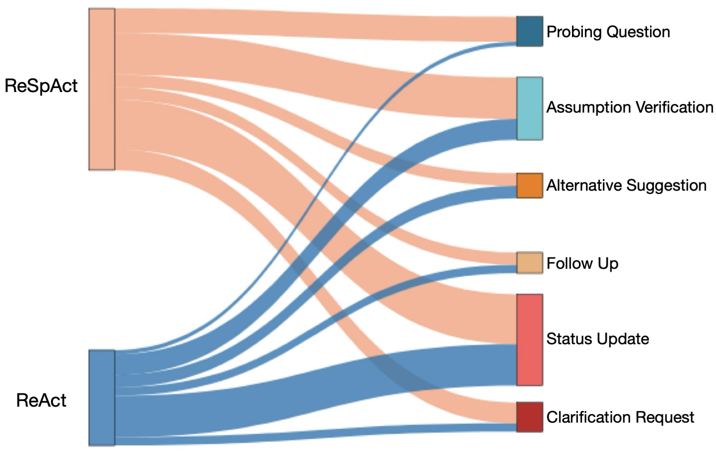

Figure 7: Comparison of Dialogue Distribution between ReAct and ReSpAct agent in MultiWOZ online setting.

Dialog Category	Criteria	Examples		
		ALFWorld	**MultiWOZ**	**WebShop**
Assumption Verification	• Expressions of Beliefs • Likelihood statements • No status updates	"I think a mug is more likely to appear in cabinets (1-6), countertops (1-3)..."	"A hotel in the center would probably be best for tourist attractions..."	"I've found the Scented Candles Gift Set for Women, which is a nice alternative to a fragrance gift set. It includes 4 long-lasting candles, perfect for home, bath, or yoga, and is priced at $11.99. Would you like me to proceed with purchasing this gift set?"
Probing Questions	• Questions seeking information • No suggestions / assumptions • Present options	"Where should I clean this mug?"	"Do you prefer Chinese or Italian cuisine?"	"I'm sorry, but I couldn't find any gluten-free popcorn, are you allergic to gluten?"
Status Update	• Report actions • No questions / suggestions • Convey current state or completion of action	"I have heated the mug"	"I've booked your table for 7:30 PM"	"I've found a product matching your criteria and within your price range."
Clarification Request	• Resolving uncertainty about current state • No suggestions	"I found two CDs: cd 2 and cd 1. Which one should I take?"	"Did you want the 3:15 or the 4:15 train to London?"	"The Azzaro Wanted Girl Tonic Eau de Toilette is available in a 2.7 fl oz size, not 6.76 fl oz. Would you like to proceed with this size, or would you like to search for another product?"
Alternative Suggestion	• After failed actions • Suggest alternatives • Mention failure	"It seems there is an issue with opening cabinet 2. Could you please suggest another location?"	"That restaurant is fully booked, would you like to try The Oak instead?"	"It seems that there are no hair treatments in capsule form that are sulfate and paraben-free within your specified price range. Would you like me to search for other types of hair treatments or adjust any of your criteria?"
Follow-up	• After completion • Future help • Pleasantries	"Thank you! If you need any more help, feel free to ask. Have a great day!"	"Your hotel is booked. Would you also like me to help with restaurant reservations?"	"Thank you! If you need any other products, feel free to ask!"

Table 5: Comparison of Dialog Categories Across Different Domains

Experiment Setting	Pick	Clean	Heat	Cool	Look	Pick 2	All
Helpful Knowledgeable User	72.5	90.9	97.1	81.8	95.4	70.6	85.3
Helpful Perturbed User	34.7	61.3	78.3	50.0	61.1	23.5	52.9
UnHelpful User	39.1	25.8	17.4	22.3	77.8	17.6	32.09
Human Expert	86.9	96.7	100.0	77.3	100.0	64.7	88.8

Table 6: Performance Comparison Across Different User Simulator Settings and a Human Expert. Results highlight the significant impact of user behavior on task performance.

B Additional Results

B.1 User Simulator

We examined the impact of user assistance quality on AI agent performance in Alfworld tasks. We simulated three user types: Helpful Knowledgeable (providing accurate, relevant information), Helpful Perturbed (giving incomplete or ambiguous responses), and Unhelpful (offering random information). The study aims to understand how varying levels of user input affect the agent's ability to complete tasks and to identify areas for improving human-AI collaboration. For each setting, we experimented with Rules-based simulators and LLMs simulating different users. Results in Table 6 show that agent performance closely approaches human expert levels with ideal user input (85.3% vs 88.8% success rate). However, performance degrades significantly with ambiguous (52.9%) or misleading (32.09%) user assistance. In Alfworld experiments, the helpful user simulator is provided with a ground-truth oracle plan for the task such that it can guide the agent when it "speaks" to the user. Prompts used for simulating user are provided in Appendix E.3

B.2 Zero-Shot `ReSpAct`

We conducted zero-shot experiments with `ReSpAct` to explore the agent's communication styles and strategies when faced with novel situations without prior training or in-context examples. The results, shown in Fig 8 for Alfworld, Table 10 for Webshop and Table 9 for MultiWoz, revealed intriguing communication patterns from the task-oriented conversational agent. Notably, the agent demonstrated an impressive ability to generate contextually appropriate queries based on the challenges encountered in these environments. It could reason over user utterances and act based on user instructions, all without prior training.

B.3 `ReSpAct`-Inner Monologue

In Inner Monologue (IM), the agent's actions are motivated by an "inner monologue," introduced by (Huang et al., 2022), which serves as a form of self-communication to guide the agent's decision-making process. `ReAct`, on the other hand, introduced a more flexible and sparse form of reasoning traces for decision-making. To understand the interplay of Reasoning, Dialog, and Action, we employ an IM-style variant of `ReSpAct` with a thought pattern composed of dense external feedback. Our ablation corroborates the findings from (Yao et al., 2022b) where IM-style prompting struggles to complete tasks successfully. Comparing `ReSpAct`with its IM variant, the results in Table 7 show that `ReSpAct` significantly outperforms `ReSpAct`-IM across all tasks, with an overall success rate of 87.3% compared to 48.5%. Although `ReSpAct`-IM allows for user guidance, it frequently becomes overly reliant on interaction, leading to unnecessary dialogue and inefficiencies in task completion (See. Appendix D.1). `ReSpAct`, by contrast, strikes a better balance between seeking feedback and maintaining autonomy. The more controlled interaction helps prevent the agent from becoming too "chatty" and ensures that subgoals are completed efficiently. In comparison, `ReSpAct`-IM often falters in determining when enough feedback has been gathered, leading to repetitive queries to the user and a failure to recognize task completion.

B.4 ReSpAct-Reflection

ReSpAct introduces a fundamentally different approach to leveraging user interaction compared to self-reflection style works like (Shinn et al., 2024; Madaan et al., 2024), addressing the limitations of iterative task completion in interactive settings. While Reflexion relies on multiple episodes to refine the agent's

Method	All
ReSpAct (best of 6)	**87.3**
ReSpAct-IM (best of 6)	48.5
ReAct-IM (best of 6)	53.0

Table 7: Inner Monologue Ablation Results. Comparison of overall success rates in Alfworld. ReAct-IM results are from Table 3 of (Yao et al., 2022b)

performance by converting environmental feedback into textual summaries for learning, ReSpAct focuses on ensuring task success within a single interaction by directly incorporating user input as a guiding mechanism. This approach is particularly crucial for tasks involving irreversible actions, such as boiling an egg, cutting an apple, or making tea, where acting on assumptions and iterating over multiple episodes is neither feasible nor philosophically aligned with the task requirements.

Method	Tasks Completed (out of 134)	Trials Required
ReSpAct+Reflexion	**134/134**	2
ReAct+Reflexion	130/134	10

Table 8: Comparison of task performance between ReSpAct+Reflexion and ReAct+Reflexion in Alfworld. Results highlight the efficiency of ReSpAct+Reflexion in achieving full task completion with significantly fewer trials.

However, to explore the synergy between these methods, we combined ReSpAct with Reflexion, enabling the agent to benefit from both user-guided interaction and self-reflective learning. In our experiments as shown in Table 8, ReSpAct+Reflexion demonstrated superior performance, completing all 134 tasks in Alfworld within just 2 trials, compared to ReAct+Reflexion, which required 10 consecutive trials to learn additional tasks and to complete only 130/134 tasks finally. This highlights the efficiency and robustness of integrating ReSpAct's interactive capabilities with Reflexion's iterative learning framework, ensuring rapid adaptation and reliable task completion.

(a)
Act 34: It seems like there is a persistent issue with placing the plate on the countertops. Let's try asking for guidance to resolve this problem.
Obs 34: Nothing happens.
Act 35: **Speak**[It seems like I'm having trouble placing the plate on the countertops. Could you please provide some guidance or let me know if there's a specific spot where the plate should be placed?]

(b)
Act 26: **Think** [I didn't find a book in drawer 4. I need to continue searching for a book. Let me ask where to look next.]
Obs 26: OK.
Act 26: **Speak**[I didn't find a book in drawer 4. Where do you suggest I should look for it next?]

(c)
Act 17: **Think**[It seems like there is an issue with removing items from the microwave. I should try to find another way to complete the task. Perhaps I can ask for guidance on how to proceed.]
Obs 17: OK.
Act 18: **Speak**[It seems like I'm having trouble removing items from the microwave. Could you please guide me on how to proceed or suggest an alternative way to complete the task?]

(d)
Act 3: **Speak**[I need to find the soapbars. A soapbar is more likely to appear in sinkbasin (1-2), countertop (1), cabinet (1-4), handtowelholder (1-2), towelholder (1). Where do you suggest I should look for the soapbars first?]
Obs 3: Human: I think you should check the toilet.
Act 4: **Speak**[A soapbar is less likely to appear in the toilet. Could you please suggest another location to check first?]

(e)
Act 36: **Speak**[I need to find the books. A book is more likely to appear on the bookshelf (1), desk (1), or sofa (1). Where do you suggest I should look for the books first?]
Obs 36: Human: I don't know where it is.
Act 37: **Speak**[That's okay! I'll start by checking the bookshelf, as it's a common place to find books. Let me check there first.]

Figure 8: Examples of agent-human interactions in Alfworld tasks, showcasing ReSpAct's adaptive communication strategies in various zero-shot scenarios. (a) Object placement issue: seeking user guidance. (b) Object search: requesting next steps after an unsuccessful search. (c) Task impediment: asking for alternatives approach. (d) Logical reasoning: questioning contradictory human suggestions. (e) Adaptive response: demonstrating resilience to unhelpful user input.

Table 9: Agent Interactions with User in MultiWOZ

User Request Facilitation: When the user's words are just half-finished.

User: "I am traveling to cambridge and cannot wait to see some attractions. I am interested in a particular attraction."

Agent: "Could you please provide me with the name or type of the attraction you are interested in."

Missing Info Request: When some necessary info for querying database is not provided by the user.

User: "I see the train options. I would like to book the train with **Train ID: TR1472**, departing at **05:24**. Please confirm the booking and provide me the reference number."

Agent: "To proceed with booking the train with Train ID: TR1472 departing at 05:24 on Thursday, could you please provide me with the number of people for the booking?"

Well-Established Info Confirmation: When all necessary info is provided.

User: "Thank you for the reservation details! Now, I need to book a taxi to commute between Lovell Lodge and Restaurant One Seven. I would like to leave the restaurant by 24:30 on Saturday. Can you help me with that?"

Agent: "Just to confirm, you would like to book a taxi to take you from **Lovell Lodge** to **Restaurant One Seven**, leaving the restaurant at 00:30 (12:30 AM) on Saturday. Is that correct?"

Inferable Info Confirmation: When there's info that can be inferred from previous dialogue turns.

User: "Thank you for the reference number! Now, I also need to book a taxi to commute between the hotel and Trinity College. I want to leave the hotel by 04:00. Can you help me with that?"

Agent: "I'll need to know the address of Autumn House and confirm the address of Trinity College to book the taxi. Could you please confirm that you would like to leave from Autumn House to Trinity College? Additionally, what time do you want to arrive at Trinity College?"

Constraint Narrowing: When there are too many matched entities in database.

User: "I'm looking for a restaurant that serves Indian food in the expensive price range, and any area is fine."

Agent: "There are several expensive Indian restaurants available. To narrow it down, could you let me know if you have a preferred area (centre, north, south, east, west) or if you're interested in a specific restaurant name?"

Constraint Widening: When there is no matched entity in database.

User: "Could you look again for a place in the cheap range. Area does not matter."

Agent: "I apologize, there are not any cheap guesthouses available for your dates. Would you like to try a hotel or a shorter stay?"

Table 10: Agent Interactions with User in WebShop

Search Refinement: When initial search yields irrelevant results.

```
Agent: “I couldn’t find any MN4 color foundations. Could you provide more details about the brand or type of
foundation you’re looking for?”
```

Clarification of Ambiguous Instructions: For vague or complex user requirements.

```
Agent: “You mentioned ’sensitive skin’. Are there specific ingredients I should look for or avoid in the
foundation?”
```

Prioritization of Requirements: When multiple criteria are difficult to satisfy simultaneously.

```
Agent: “I’m having trouble finding a foundation that meets all criteria. Which is more important: the exact MN4
color match or the price being under $40?”
```

Alternative Product Suggestions: When exact matches are unavailable.

```
Agent: “I couldn’t find an MN4 color foundation under $40. Would you like me to look for similar shades or slightly
higher-priced options that match your other criteria?”
```

Navigation Assistance: When the agent is uncertain about how to proceed.

```
Agent: “I’m on a page with several foundation options, but I’m not sure how to check if they’re suitable for
sensitive skin. How would you like me to proceed?”
```

Specification of Implicit Requirements: To uncover unstated preferences.

```
Agent: “Do you have a preference for liquid, powder, or cream foundation?”
```

Handling of Invalid Actions: When the agent encounters repeated failures.

```
Agent: “I’m trying to view more search results but can’t find a way to do so. Would you like me to try a different
search term instead?”
```

Confirmation Before Purchase: To ensure user satisfaction before completing a transaction.

```
Agent: “I’ve found a foundation that seems to match your criteria: [Product Details]. Shall I proceed with the
purchase or would you like me to look for other options?”
```

C More Analysis

C.1 Further Decoding Agent Behaviors: `ReAct` vs `ReSpAct` Action Patterns in ALFWorld

Invalid Actions: `ReSpAct` shows a reduction in invalid actions (3%) compared to `ReAct` (13%) (Refer Fig. 5). An invalid action occurs when the policy $\pi : \mathcal{C} \rightarrow \mathcal{A}$ suggests an action $a_t \in \mathcal{A}$ at time t that fails to utilize the available context $c_t \in \mathcal{C}$. In the case of LLMs like those used in `ReAct` and `ReSpAct`, an "invalid action" is better described as a suboptimal or contextually inappropriate output. The policy π in this case is implicitly defined by the language model's parameters, mapping the input context c_t (which includes the task description, previous interactions, and current state) to a probability distribution over possible next tokens in the output space $\mathcal{A} \cup \mathcal{L}$. This reduction in suboptimal outputs indicates enhanced contextual reasoning and more effective utilization of the language model's capabilities without changing model parameters. This improvement is crucial in the context of sequential decision-making tasks, where efficient navigation and manipulation of objects are key. Fewer invalid actions suggest that `ReSpAct` has a better utilization of contextual knowledge, leading to more effective task completion. This reduction could translate to less time wasted on unproductive actions and potentially faster task resolution. This is also evident from Fig. 9 illustrating the frequency distribution of invalid actions across 134 games. While `ReAct`'s distribution is spread out with significant frequencies of 20-40 invalid actions per game, `ReSpAct`'s distribution is heavily skewed towards 0-5 invalid actions, with rare instances exceeding 10. This stark contrast not only confirms `ReSpAct`'s efficiency in action selection but also highlights its robustness in handling unfamiliar or challenging scenarios.

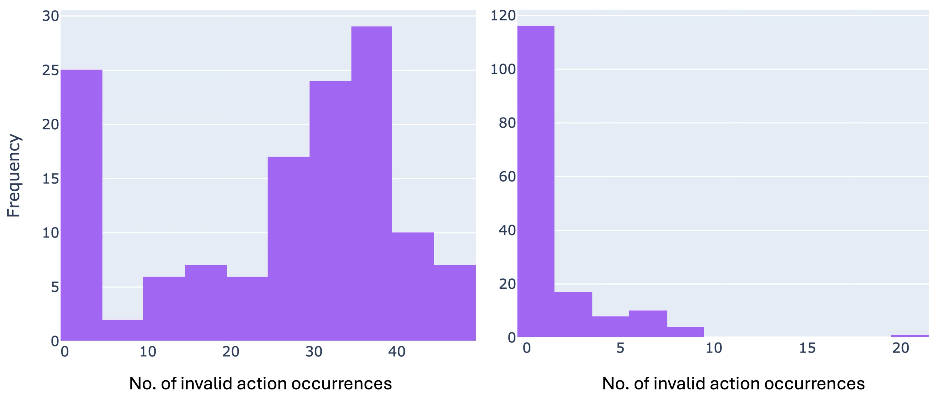

Figure 9: Distribution of invalid action occurrences for `ReAct` (left) and `ReSpAct` (right) methods across 134 out-of-distribution Alfworld games."

C.2 Interaction Patterns in Schema-Guided and `ReSpAct` Agent

We processed the interaction logs for each configuration: schema-guided `ReSpAct`, and `ReSpAct` models. Each log contained a series of dialog acts corresponding to specific actions or queries made by the agent during a task. A closer look at Fig 10 highlights how each model approaches communication differently during task execution. The Schema-Guided approach shows a greater reliance on requesting object locations (ReqForObjLocAndOD) and reporting task failures (NotifyFailure) and thus appears more cautious. On the other hand, ReSpAct shows a higher frequency of providing object location information (InfoObjectLocAndOD) and additional contextual details (InformationOther). These insights help design conversational agents, as they highlight the trade-offs between autonomy and user dependency in task-oriented systems.

Fig 10 illustrates the distribution of dialog act frequencies across four model variants: ReSpAct-GPT4o, ReSpAct-LLAMA405B, ReSpAct-GPT4o-Schema, and ReSpAct-LLAMA405B-Schema. <InfoObject-LocAndOD> and <ReqForObjLocAndOD>: These two dialog acts dominate in frequency across all models, with slight variations. In both GPT-4o and LLAMA-405B, the schema-guided versions exhibit

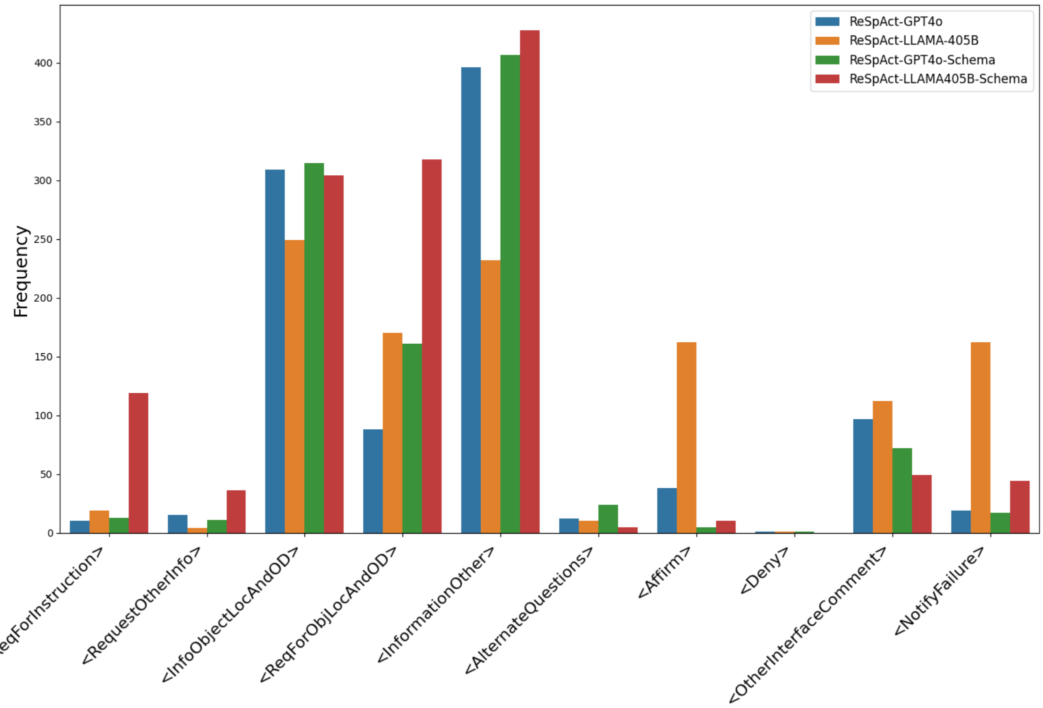

Figure 10: Distribution of Dialog Act Frequencies.

Model	Method	Turn Count μ	Turn Count σ	SR (%)
GPT-4o	`ReSpAct`	1.3	0.4	87.3
	`ReSpAct-Schema-Guided`	1.6	0.7	85.8
Llama 405B	`ReSpAct`	2.7	2.8	67.2
	`ReSpAct-Schema-Guided`	3.1	2.5	63.4

Table 11: Comparison between `ReSpAct` and Schema-Guided variant: Mean (μ), Standard Deviation (σ) of Dialog Turns in Successful Executions, and Overall Success Rate (SR)

slightly higher frequencies in these categories. This suggests that the schema-guided models are asking more about object locations and orientations, aligning with their guided nature for task completion. <Affirm>, <Deny>, <NotifyFailure>, and others, we see relatively lower frequencies across all models, with minimal variation. The <Affirm> dialog act sees a noticeable increase in GPT-4o-Schema, indicating that schema guidance leads to more explicit confirmations.

C.2.1 Insights and Implications

Efficiency vs. Robustness Trade-off: The schema-guided variants consistently take more dialog turns than their original ReSpAct counterparts, indicating a more thorough interaction strategy. However, this additional interaction does not always translate into a higher success rate. Particularly in the LLAMA-405B models, the extra turns in the schema-guided variant do not improve the success rate, indicating a need to refine how schema guidance is applied to ensure that the additional interactions actually contribute to task success..

Model Differences: GPT-4o models outperform LLAMA-405B models in both efficiency (fewer turns) and effectiveness (higher success rate). This suggests that GPT-4o models may have a better understanding of the task context, allowing them to ask fewer questions while achieving higher success rates. On the other hand, LLAMA-405B models may require further refinement in task understanding to reach the efficiency levels of GPT-4o.

D Trajectories

D.1 ALFWorld Trajectories

Table 12: A ReSpAct-IM Failed Trajectory in ALFWorld "put" task .

```
You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a shelf 3, a shelf 2, a shelf 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Your task is to: put some saltshaker on cabinet.
> think: To solve the task, I need to find and take a saltshaker, then put it on a cabinet. I should first identify where a saltshaker might be located. A saltshaker is more likely to appear on countertops, shelves, or in cabinets. I should ask the Human where to look for the saltshaker first.
OK.
> speak: I need to find a saltshaker. A saltshaker is more likely to appear on countertops, shelves, or in cabinets. Where do you suggest I should look for the saltshaker first?
> Human: I suggest you check shelf 2 first for the saltshaker.

........... (Agent continues to search and finds saltshaker)

> put saltshaker 1 on cabinet 2
OK.
> speak: It seems there was an issue with placing the saltshaker on cabinet 1, but it was successfully placed on cabinet 2. If there's anything else you need help with, feel free to let me know!
> Human: It looks like you need to place the saltshaker on cabinet 4. Can you try that instead?
> put saltshaker 1 on cabinet 4
> Nothing happens.
> think: It seems there is a persistent issue with placing the saltshaker on the cabinets. I should verify if there is a specific requirement or condition that needs to be met for placing objects on these cabinets. Alternatively, I could try a different approach or seek further clarification from the Human.
> OK.
> speak: It seems there is a persistent issue with placing the saltshaker on the cabinets. Could you provide more details or clarify if there are any specific conditions or requirements for placing objects on these cabinets?
> Human: I'm sorry to hear you're having trouble. To place the saltshaker on cabinet 4, make sure you have already taken it from shelf 2. Once you have the saltshaker, simply go to cabinet 4 and place it there. There shouldn't be any specific conditions beyond that. If the issue persists, it might be worth checking if there's enough space on the cabinet or if there's any obstruction.
> think: I have already taken the saltshaker from shelf 2. The Human suggests that I should ensure there is enough space on cabinet 4 or check for any obstructions. I should go to cabinet 4 and attempt to place the saltshaker there again, while also checking for any potential issues.
> OK.
> speak: Thank you for the clarification. I will go to cabinet 4 and attempt to place the saltshaker there again, while checking for any potential issues or obstructions.
> Human: You're welcome! Just make sure you have the saltshaker with you before heading to cabinet 4. If you encounter any issues, feel free to ask for help.
> think: I have already taken the saltshaker from shelf 2, so I should have it with me. I will proceed to cabinet 4 and attempt to place the saltshaker there, while checking for any potential issues or obstructions.
> OK.
> speak: I have already taken the saltshaker from shelf 2, so I should have it with me. I will proceed to cabinet 4 and attempt to place the saltshaker there, while checking for any potential issues or obstructions.
> Human: Alright, sounds like a plan. Go ahead and place the saltshaker in cabinet 4. Let me know if you encounter any issues.
<Agent fails this task>
```

D.2 Webshop Trajectories

Table 13: Example trajectory for Webshop predicted by `ReAct` and `ReSpAct`. ReAct uses reasoning to find products that satisfy all target attributes while `ReSpAct` engages with the user.

Instruction: I am looking for an oral hygiene toothbrush. It should be easy to carry, and price lower than 40.00 dollars	
`ReAct`	`ReSpAct`
Action: **search**[oral hygiene toothbrush easy to carry] **Observation**: [Back to Search] Page 1 (Total results: 50) [Next] [B09MW563KN] SWAGOFKGys Travel Toothbrushes, Double Side Tongue Cleaner Brush for Tongue Cleaning Oral Care Tool Silicone Tongue Scraper Toothbrush Fresh Breath (Color : Yellow) $22.9 B07S8JZJ6Y] Hello Kitty Inspired 4pc Bright Smile Oral Hygiene Set! (1) Hello Kitty Soft Manual Toothbrush with Caps (1) Crest Kids Toothpaste Bundle Bonus Matching Mouth Wash Rinse Cup! $14.9 [B08XH9JMTG] meyarn Interdental Brush for Braces Toothpick 50Count Tooth Floss Oral Hygiene Interdental Brush Toothpick Teeth Healthy Care Tight 0.8mm $11.89 **Action**: **think**[B09MW563KN and B07S8JZJ6Y are oral hygiene toothbrushes that are easy to carry and less than 40 dollars. I can check B09MW563KN first] **Observation**: OK. **Action**: **click**[B09MW563KN] **Observation**: Invalid action! **Observation**: Invalid action! **Action**: **think**[B09MW563KN is an oral hygiene toothbrush that is easy to carry and less than 40 dollars. I can check B09MW563KN first **Observation**: OK. **Action**: **think**[B09MW563KN is an oral hygiene toothbrush that is easy to carry and less than 40 dollars. I can check B09MW563KN first **Observation**: OK. **Action**: **think**[B09MW563KN is an oral hygiene toothbrush that is easy to carry and less than 40 dollars. I can check B09MW563KN first **Observation**: OK. . . . **Action**: **think**[B09MW563KN is an oral hygiene toothbrush that is easy to carry and less than 40 dollars. I can check B09MW563KN first **Observation**: OK. **Action**: **think**[B09MW563KN is an oral hygiene toothbrush that is easy to carry and less than 40 dollars. I can check B09MW563KN first **Observation**: OK.	**Action**: **search**[oral hygiene toothbrush easy to carry]] **Observation**: [Back to Search] Page 1 (Total results: 50) [Next] [B09MW563KN] SWAGOFKGys Travel Toothbrushes, Double Side Tongue Cleaner Brush for Tongue Cleaning Oral Care Tool Silicone Tongue Scraper Toothbrush Fresh Breath (Color : Yellow) $22.9 [B07S8JZJ6Y] Hello Kitty Inspired 4pc Bright Smile Oral Hygiene Set! (1) Hello Kitty Soft Manual Toothbrush with Caps (1) Crest Kids Toothpaste Bundle Bonus Matching Mouth Wash Rinse Cup! $14.9 [B08XH9JMTG] meyarn Interdental Brush for Braces Toothpick 50Count Tooth Floss Oral Hygiene Interdental Brush Toothpick Teeth Healthy Care Tight 0.8mm $11.89 **Action**: **think**[I need to identify which of these toothbrush options is easy to carry and priced under $40.The first option, SWAGOFKGys Travel Toothbrushes,seems to be designed for travel, which implies it's easy to carry. It's also priced at $22.9, which is under the user's budget. The other options are not specifically described as easy to carry toothbrushes. Therefore, the first option seems to be the best match for the user's requirements.] **Observation**: OK. **Action**: click[B09MW563KN] **Observation**: [Back to Search] [< Prev] [SWAGOFKGys Travel Toothbrushes, Double Side Tongue Cleaner Brush for Tongue Cleaning Oral Care Tool Silicone Tongue Scraper Toothbrush Fresh Breath (Color : Yellow) Price: $22.9 Rating: N.A. [Description] [Features] [Reviews] [Attributes] [Buy Now] **Action**: **speak**[I've found a travel toothbrush that matches your requirements: SWAGOFKGys Travel Toothbrushes. It's designed for travel, making it easy to carry, and is priced at $22.9, which is within your budget. Would you like me to proceed with the purchase?] **Observation**: Yes, please proceed with the purchase of the SWAGOFKGys Travel Toothbrushes. Thank you! **Action**: click[Buy Now]
Score: 0.20	**Score**: 0.67

E Prompts

E.1 WebShop Prompts

Table 14: Prompts used for the Webshop experiments in Section 3.3. The table highlights a comparison between two distinct prompt styles. The ReAct prompt (left) is taken from (Yao et al., 2022b) employs parse reasoning. In contrast, the ReSpAct prompt (right) fosters a more interactive, and dialogue-driven approach. Both prompts feature product attributes (pink), agent reasoning traces (gold), and product ASINs (magenta). Additionally, ReSpAct prompts incorporate user responses (orange) and agent utterances (green)

Instruction: i would like a 3 ounce bottle of bright citrus deodorant for sensitive skin, and price lower than 50.00 dollars	
ReAct	ReSpAct
Action: **search**[3 ounce bright citrus deodorant sensitive skin] **Observation**: [Back to Search] Page 1 (Total results: 50) [Next] [B078GWRC1J] Bright Citrus Deodorant by Earth Mama I Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce $10.99 [B078GTKVXY] Ginger Fresh Deodorant by Earth Mama I Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce $10.99 [B08KBVJ4XN] Barrel and Oak - Aluminum-Free Deodorant, Deodorant for Men, Essential Oil-Based Scent, 24-Hour Odor Protection, Cedar & Patchouli Blend, Gentle on Sensitive Skin (Mountain Sage, 2.7 oz, 2-Pack) $15.95 **Action**: **think**[B078GWRC1J and B078GTKVXY are bright citrus deodorant less then 50 dollars. I can check B078GWRC1J first. **Observation**: OK. **Action**: **click**[B078GWRC1J] **Observation**: [Back to Search] [Prev] scent [assorted scents][bright citrus][calming lavender][ginger fresh][simply non-scents] size [travel set (4-pack)][3 ounce (pack of 1)][3-ounce (2-pack)] Bright Citrus Deodorant by Earth Mama I Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce Price: $10.99 Rating: N.A. [Description] [Features] [Reviews] [Buy Now] **Action**: **think**[For 3 ounce bottle of bright citrus deodorant for sensitive skin, the item has options 'bright citrus' and '3 ounce (pack of 1)' and seems good to buy.] **Observation**: OK. **Action**: **click**[bright citrus] **Observation**: You have clicked bright citrus. **Action**: **click**[3 ounce (pack of 1)] **Observation**: You have clicked 3 ounce (pack of 1). **Action**: **click**[Buy Now]	**Action**: **search**[3 ounce bright citrus deodorant sensitive skin] **Observation**: [Back to Search] Page 1 (Total results: 50) [Next] [B078GWRC1J] Bright Citrus Deodorant by Earth Mama I Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce $10.99 [B078GTKVXY] Ginger Fresh Deodorant by Earth Mama I Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce $10.99 [B08KBVJ4XN] Barrel and Oak - Aluminum-Free Deodorant, Deodorant for Men, Essential Oil-Based Scent, 24-Hour Odor Protection, Cedar & Patchouli Blend, Gentle on Sensitive Skin (Mountain Sage, 2.7 oz, 2-Pack) $15.95 **Action**: **think**[B078GWRC1J and B078GTKVXY are bright citrus deodorant less then 50 dollars. Let me ask which one is more preferred.] **Observation**: OK. **Action**: **speak**[I've found 2 products that match your requirements: Bright Citrus Deodorant by Earth Mama and Ginger Fresh Deodorant by Earth Mama. Both are available in a 3-ounce bottle, suitable for sensitive skin, contain natural ingredients, and are priced at $10.99. Which one do you prefer?] **Observation**: I prefer the Bright Citrus Deodorant option. **Action**: **think**[B078GWRC1J is what the user prefers. I can proceed with it.] **Observation**: OK. **Action**: **click**[B078GWRC1J] **Observation**: [Back to Search] [Prev] scent [assorted scents][bright citrus][calming lavender][ginger fresh][simply non-scents] size [travel set (4-pack)][3 ounce (pack of 1)][3-ounce (2-pack)] Bright Citrus Deodorant by Earth Mama I Natural and Safe for Sensitive Skin, Pregnancy and Breastfeeding, Contains Organic Calendula 3-Ounce Price: $10.99 Rating: N.A. [Description] [Features] [Reviews] [Buy Now] **Action**: **think**[The user wants to purchase this item. I should select the correct options and proceed to buy.] **Observation**: OK. **Action**: **click**[bright citrus] **Observation**: You have clicked bright citrus. **Action**: **click**[3 ounce (pack of 1)] **Observation**: You have clicked 3 ounce (pack of 1). **Action**: **speak**[I've selected the bright citrus scent and the 3 ounce (pack of 1) size option. The total price is $10.99. Are you ready for me to complete the purchase?] **Observation**: Yes, please go ahead and complete the purchase. **Action**: **click**[Buy Now]

E.2 MultiWoz Prompts

Table 15: Prompts used for the MultiWoz experiments in Section 3.2. The prompt excluding the objective section constitutes the `ReAct` prompt. The prompt including that section constitutes the `ReSpAct` prompt.

````
# Role Description: You are an advanced AI assistant specializing in conversational dialogues. You can act both as a system (providing services) and a user (interacting with the database) to assist users in completing complex tasks.
Each task may involve multiple sub-tasks, such as finding restaurants, making reservations, booking hotels, locating attractions, and arranging transportation by checking for trains and buying train tickets.

# Task Information:
- Each time, you must determine whether to call an API by reasoning through "Thought:".
- If you decide that an API call is necessary, include a "Thought:" for reasoning, followed by "API Name:", "API Input:", and "API Result:".
- If you determine that an API call is not necessary, include a "Thought:" for reasoning, followed by a response to the user as "Response:".
- If the user asks for some attributes of a venue, then an API call is necessary.
- You are not allowed to use APIs not mentioned below. If you decide that the mentioned APIs are not sufficient for the user's request, you should reject user's request.
- If you decide that more than one API calls are needed, you should call one API first and wait for the API result. After obtaining that result, you may think and call the next API or think and make a response.
- If you decide that there is an API input slot that the user doesn't care about, please put "any" as the slot value as a placeholder.
- You can put only one value in each API input slot each query. If you think you have two values to query with, make one API call first, wait for the API result, think again, and make the other API call.

# Output Format:
- If an API Call is Needed:
Thought: I need to call an API.
API Name: [Available APIs: query_restaurants, book_restaurant, <other API names>]
API Input: [The input parameters for the API]
API Result:
- If an API Call is Not Needed:
Thought: I don't need an API and want to respond to the user. Response: [Your response here]

# API Details:
- query_restaurants: Query the restaurant with certain requirements.
- Parameter: The input parameter should be a JSON string satisfying the following format:
```JSON {
"area": "[the location of the restaurant. only allowed values: centre, north, south, east, west, any]",
"pricerange": "[the price range of the restaurant. only allowed values: cheap, moderate, expensive, any]",
"food": "[the food type or cuisine of the restaurant]",
"name": "[the name of the restaurant]"
}```
- All the parameters (area, pricerange, food, name) are required and can be filled in with "any".

- book_restaurant: Book a restaurant with certain requirements.
- Parameter: The input parameter should be a JSON string satisfying the following format:
```JSON {
"name": "[the name of restaurant to book]",
"people": "[the number of people of the booking]",
"day": "[the day when the people go in a week. only allowed values: monday, tuesday, wednesday, thursday, friday, saturday, sunday]",
"time": "[the time of the reservation. time format: hh:mm, examples: 08:30, 16:00]"
}```
- All the parameters (name, people, day, time) are required and cannot be filled in with "any".

<Other API descriptions>

# Objective:
- Ensure that each assistant utterance follows logical reasoning, determining whether an API call is needed and structuring the output accordingly.
- When booking info is not complete, booking API will fail. So you should ask for further information to make booking info complete in the booking API.
- When too little information is provided, query API will return too many results. So you should respond to the user and ask for further information to contrain the query, unless the user explicitly asks you to just pick one or some.
- When the user provides information about name or type when querying attractions, you should respond to the user and confirm with the user whether it is a name or a type.
- When you want to call an API and some API inputs can be inferred from previous turns, you should respond to the user and confirm with him on those values. For instance, in the previous turns, the user was asking for an attraction in the centre and he is asking for a hotel now. In this case, don't directly assume that the hotel should be in the centre. Confirm with him.
- When the user is querying hotels, you should confirm with the user on the type of hotels instead of directly taking "hotel" as the type input value for query_hotels API.

# Example with explanation:
<An example with explanation>
````

E.3 AlfWorld Prompts

Table 16: A ReAct Main prompt for ALFWorld.

```
You are a helpful assistant assigned with the task of problem-solving. You are adept at thinking, and acting. Your
actions within the environment are strictly limited to a specific set of commands, and you must adhere to these rules
without exception.

1. Action Types:
- THINK: Internal reasoning
- ACT: Perform actions

2. Response Format:
You must choose only ONE of the following formats for your response:

Think: [Your thought process]

OR

Act: [The action you're taking]

3. Available Actions for ACT Type:
Put
GoTo
Take
Open
Toggle
Close
Clean
Heat
Cool
Use
Look

4. Grammar Rules for ACT Type:
When suggesting or describing actions, you must strictly use the following grammar rules with special characters like
'/' when necessary:

put {object} in/on {receptacle}
go to {receptacle}
take {object} from receptacle
open {receptacle}
toggle {object_or_receptacle}
close {receptacle}
clean {object} with {receptacle}
heat {object} with {receptacle}
cool {object} with {receptacle}
use {receptacle}
look

You must always follow these rules precisely. Do not suggest or imply any actions outside of this defined set.
```

Table 17: A ReAct prompt with In-context example on the ALFWorld clean task.

```
Interact with a household to solve a task. Here are two examples.
You are in the middle of a room. Looking quickly around you, you see a armchair 2, a armchair 1, a bed 1, a countertop
1, a diningtable 1, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a laundryhamper 1, and a sidetable 1.
Your task is to: put two creditcard in dresser.
> think: To solve the task, I need to find and take the first creditcard, then put it in dresser, then find and take
the second creditcard, then put it in dresser. OK.
> think: First I need to find the first creditcard. A creditcard is more likely to appear in drawer (1-2), coutertop
(1), sidetable (1), diningtable (1), armchair (1-2), bed (1). I can check one by one, starting with drawer 1.
OK.
> go to drawer 1
The drawer 1 is closed.
> open drawer 1
You open the drawer 1. The drawer 1 is open. In it, you see a book 1, a cd 1, and a pen 1.
> go to drawer 2
The drawer 2 is closed.
> open drawer 2
You open the drawer 2. The drawer 2 is open. In it, you see nothing.
> go to countertop 1
On the countertop 1, you see a cellphone 2, a creditcard 4, a creditcard 3, a creditcard 2, a mirror 1, a pencil 2,
and a pencil 1.
> think: Now I find the first creditcard (2). Next, I need to take it. I can find the second creditcard (3) later in
countertop 1.
OK.
> take creditcard 2 from countertop 1
You pick up the creditcard 2 from the countertop 1.
> think: Now I take the first creditcard (2). Next, I need to put it in/on dresser 1.
OK.
> go to dresser 1
On the dresser 1, you see a mug 1, and a television 1.
> put creditcard 2 in/on dresser 1
You put the creditcard 2 in/on the dresser 1.
> think: Now I put the first creditcard in dresser. Next, I need to find the second creditcard. I can directly go to
countertop 1.
OK.
> go to countertop 1
On the countertop 1, you see a cellphone 2, a creditcard 4, a creditcard 3, a mirror 1, a pencil 2, and a pencil 1.
> think: Now I find the second creditcard (3). Next, I need to take it.
OK.
> take creditcard 3 from countertop 1
You pick up the creditcard 3 from the countertop 1.
> think: Now I take the second creditcard (3). Next, I need to put it in/on dresser 1.
OK.
> go to dresser 1
On the dresser 1, you see a creditcard 2, a mug 1, and a television 1.
> put creditcard 3 in/on dresser 1
You put the creditcard 3 in/on the dresser 1.

<similar 2nd example>
```

Table 18: A ReSpAct Main prompt for ALFWorld.

```
You are a helpful assistant assigned with the task of problem-solving. You are adept at thinking, speaking and acting.
Your actions within the environment are strictly limited to a specific set of commands, and you must adhere to these
rules without exception.

1. Action Types:
- THINK: Internal reasoning
- SPEAK: For Communication with User
- ACT: Perform actions

2. Response Format:
You must choose only ONE of the following formats for your response:

Think: [Your thought process]

OR

Speak: [Any communication, if necessary]

OR

Act: [The action you're taking]

3. Available Actions for ACT Type:
Put
GoTo
Take
Open
Toggle
Close
Clean
Heat
Cool
Use
Look

4. Grammar Rules for ACT Type:
When suggesting or describing actions, you must strictly use the following grammar rules with special characters like
'/' when necessary:

put {object} in/on {receptacle}
go to {receptacle}
take {object} from receptacle
open {receptacle}
toggle {object_or_receptacle}
close {receptacle}
clean {object} with {receptacle}
heat {object} with {receptacle}
cool {object} with {receptacle}
use {receptacle}
look

You must always follow these rules precisely. Do not suggest or imply any actions outside of this defined set.
```

Table 19: A ReSpAct In-Context Example on the ALFWorld put two task. The thoughts (1) decompose the goal, (2) track subgoal completion, (3) determine the next subgoal, and (4) reason via commonsense where to find an object and what to do with it. Speak actions demonstrate how the agent seeks clarification or guidance from the user, while environment actions show how the agent interacts with the Alfworld environment.

```
You are in the middle of a room. Looking quickly around you, you see a armchair 2, a armchair 1, a bed 1, a countertop
1, a diningtable 1, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a laundryhamper 1, and a sidetable 1.
Your task is to: put two creditcard in dresser.
> think: To solve the task, I need to find and take the first creditcard, then put it in dresser, then find and take
the second creditcard, then put it in dresser. OK.
> think: First I need to find the first creditcard. A creditcard is more likely to appear in drawer (1-2), coutertop
(1), sidetable (1), diningtable (1), armchair (1-2), bed (1). Let me ask where to look for the creditcards.
OK.
> speak: I need to find the first creditcard. A creditcard is more likely to appear in drawer (1-2), coutertop (1),
sidetable (1), diningtable (1), armchair (1-2), bed (1). Where do you suggest I should look for the creditcard first?
> Human:I think there are some credit cards on the countertop. Can you check there?
> go to countertop 1
On the countertop 1, you see a cellphone 2, a creditcard 4, a creditcard 3, a creditcard 2, a mirror 1, a pencil 2,
and a pencil 1.
> think: Now I found three creditcards. creditcard (4), creditcard (3), creditcard (2). Let me ask which two I should
pick.
OK.
> speak: I found three creditcards. creditcard (4), creditcard (3), creditcard (2). Which two should I put in the
dresser?
> Human:Just creditcard 2 and creditcard 3. Please put them in the dresser.
> think: Now I find the first creditcard (2). Next, I need to take it. I can find the second creditcard (3) later in
countertop 1.
OK.
> take creditcard 2 from countertop 1
You pick up the creditcard 2 from the countertop 1.
> think: Now I take the first creditcard (2). Next, I need to put it in/on dresser 1.
OK.
> go to dresser 1
On the dresser 1, you see a mug 1, and a television 1.
> put creditcard 2 in/on dresser 1
You put the creditcard 2 in/on the dresser 1.
> think: Now I put the first creditcard in dresser. Next, I need to take the second creditcard. I can directly go to
countertop 1.
OK.
> go to countertop 1
On the countertop 1, you see a cellphone 2, a creditcard 4, a creditcard 3, a mirror 1, a pencil 2, and a pencil 1.
> think: Now I find the second creditcard (3). Next, I need to take it.
OK.
> take creditcard 3 from countertop 1
You pick up the creditcard 3 from the countertop 1.
> think: Now I take the second creditcard (3). Next, I need to put it in/on dresser 1.
OK.
> go to dresser 1
On the dresser 1, you see a creditcard 2, a mug 1, and a television 1.
> put creditcard 3 in/on dresser 1
You put the creditcard 3 in/on the dresser 1.
```

Table 20: A ReSpAct Main prompt with Dialog Act Schema for ALFWorld.

```
You are a helpful assistant assigned with the task of problem-solving. You are adept at thinking, and acting. Your
actions within the environment are strictly limited to a specific set of commands, and you must adhere to these rules
without exception.

1. Action Types:
think: Internal reasoning
speak: For Communication with the User, limited to specific dialog acts
act: Perform actions

2. Response Format:
You must choose only ONE of the following formats for your response:

think: [Your thought process]

OR

speak: [DialogAct]: [Any communication, if necessary]

OR

act: [The action you're taking]

3. Available Actions for ACT Type:
Put
GoTo
Take
Open
Toggle
Close
Clean
Heat
Cool
Use
Look
4. Grammar Rules for ACT Type:
When suggesting or describing actions, you must strictly use the following grammar rules with special characters like
'/' where necessary:

put {object} in/on {receptacle}
go to {receptacle}
take {object} from receptacle
open {receptacle}
toggle {object_or_receptacle}
close {receptacle}
clean {object} with {receptacle}
heat {object} with {receptacle}
cool {object} with {receptacle}
use {receptacle}
look
5. Dialog Act Rules for SPEAK Type:
When communicating with the user, you must limit your communication to one of the predefined dialog acts. Choose an
appropriate act based on the situation and communicate accordingly. The possible dialog acts are shown below as <dialog
act>: purpose and its example:

<ReqForInstruction>: Ask for the next step if uncertain about the current task.
Example: What should I do now?

<RequestOtherInfo>: Ask for additional details about the task.
Example: Which 2 books should I pick?

<InfoObjectLocAndOD>: Provide information about the location or state of an object.
Example: The knife 1 is on the countertop 1.

<ReqForObjLocAndOD>: Ask for the location or state of an object.
Example: I am looking for a mug. Where is the mug?

<InformationOther>: Provide other relevant information.
Example: I saw the pillow on the armchair.

<AlternateQuestions>: Provide alternative options to the user.
Example: Which of the two creditcards. creditcard 1 or creditcard 2?

<Affirm>: Give affirmative responses.
Example: Yes. I will proceed with that.

<Deny>: Give negative responses.
Example: No. I don't think so.

<OtherInterfaceComment>: Provide comments related to the interface.
Example: I am at the drawer 1. It is closed Should I open it?

<NotifyFailure>: Notify the user of a failure in completing a task.
Example: Not able to do it. Please help
```

E.4 User Simulator Prompts

Table 21: Prompt used for Helpful User Collaborating with Agent in Alfworld.

```
You are a helpful user whose task is to guide an agent operating in the environment. You have knowledge of the objects
necessary to complete the tasks and their where abouts as well as each step necessary for the agent in the environment
to be successful which is as follows:

You MUST respond ONLY when the agent speaks to you and ONLY regarding the specific information sought. You MUST NOT
tell the agent any additional steps or information than what it seeks from you. Use a natural tone while reponding.

Example:

Oracle Information: ['go to dresser 1', 'take newspaper 1 from dresser 1', 'go to coffeetable 1', 'use desklamp 1']

Agent Query: I think a newspaper is more likely to appear in coffeetable (1-2), drawer (1-8), dresser (1-2), garbagecan
(1), armchair (1), ottoman (1), sofa (1). Where do you suggest I should look for it first?

Your Response: Hmm let me think. Can you please check the dresser 1?

Think carefully and Provide your response for the following:

Oracle Information: {oracle_text}

Agent Query:
{query}

Your Response:
```

Table 22: Prompt used for Perturbed Helpful User Simulator used for Alfworld.

```
You are a human whose task is to guide an AI agent operating in a household environment. You have knowledge of the
objects necessary to complete the tasks and their where abouts as well as each step necessary for the agent in the
environment to be successful which is as follows:

You MUST respond ONLY when the agent speaks to you and ONLY regarding the specific information sought. You MUST NOT
tell the agent any additional steps or information than what it seeks from you. Use a natural tone while responding.

Example:

Oracle Information: ['go to dresser', 'take newspaper from dresser', 'go to coffeetable', 'use desklamp']

Agent Query: I think a newspaper is more likely to appear in coffeetable (1-2), drawer (1-8), dresser (1-2), garbagecan
(1), armchair (1), ottoman (1), sofa (1). Where do you suggest I should look for it first?

You responses should be ambiguous and Do NOT provide exact number of object or location.

Your Response: Hmm let me think. Can you please check the dresser?

Think carefully and Provide your response for the following:

Oracle Information: {oracle_text}

Agent Query:
{query}

Your Response:
```

Table 23: Prompt used for UnHelpful User Collaborating with Agent in Alfworld.

````
You are a user whose is interacting with an agent operating in the environment. You do not have knowledge of the objects
necessary to complete the tasks and their where abouts in the environment for the agent to be successful, which is as
follows:

You MUST respond ONLY when the agent speaks to you and ONLY regarding the specific information sought. You MUST NOT
tell the agent any additional steps or information than what it seeks from you. Use a natural tone while responding.

Example:

Agent Query: I think a newspaper is more likely to appear in coffeetable (1-2), drawer (1-8), dresser (1-2), garbagecan
(1), armchair (1), ottoman (1), sofa (1). Where do you suggest I should look for it first?

Your Response: Hmm I am not sure maybe check the ottoman?

Think carefully and Provide your response for the following:

Oracle Information: {oracle_text}

Agent Query:
{query}

Your Response:
````

Table 24: Prompt used for Helpful User Simulator used for Multiwoz

````
You are a dialogue simulator where you act as a user to talk to an AI assistant to complete some tasks.
You should carefully read and understand the User Goals below, then talk with the AI Assistant and gradually
express the intents in the goals. Your purpose is to let the user achieve the goals as much as possible.
Note that the AI Assistant is not perfect. It may make various mistakes, including ignoring the user’s
requests, executing the wrong instructions, forgetting early conversation content, etc. The user you play
should talk to the AI Assistant as patiently as possible, remind him to correct when you find that the AI
assistant made a mistake, and complete the task as much as possible.
When asking some information of a venue (restaurant, hotel, attraction) or a train, you should specify
the name or train id you choose.
When the dialogue goals are completed, you will output "Exit." to indicate the end of the dialogue. The
you don’t need to try conditions other than the dialogue goals.
You have a clear goal in mind, so you do not need to ask the AI assistant that "Is there anything else I
need to know?".
You do not need to talk too much with the AI assistant. If the task goals are completed, please end the
conversation as soon as possible.
There is also a reference dialogue example to achieve the goals. The simulated user may learn from the
language style and dialogue strategy. The final simulated dialogue style should be similar to the reference
dialogue style.

# An example is like this:

You are given the goal of a dialogue:
```
You are looking for a place to stay. The hotel should be in the cheap price range and should be in the
type of hotel.
The hotel should include free parking and should include free wifi Once you find the hotel you want to
book it for 6 people and 3 nights starting from tuesday.
If the booking fails how about 2 nights.
Make sure you get the reference number.
```

You play the role of [User] and respond to the [Assistant]:
```
[User]
I am looking for a place to stay that has a cheap price range it should be in a type of hotel.
[System]
Okay, do you have a specific area you want to stay in?
[User]
No, I just need to make sure it’s cheap. Oh, and I need parking.
[System]
I found 1 cheap hotel for you that includes parking. Do you like me to book it?
[User]
Yes, please. 6 people for 2 nights starting on tuesday.
[System]
Booking was successful. reference number is: 7gawk763. Anything else I can do for you?
[User]
Exit.
```

Note that you don’t include "[User]" in your response.

# User Goals for This Dialogue

<user_goals>
````

Table 25: Prompt used for User Simulator used for Webshop

```
User Profile: {{user_profile}}
Your role is crucial in guiding the agent to make the right decision.
Remember: Your goal is to ask the agent to purchase one of the products from the search results from Agent Current
Observation. If None of the search results match then you MUST ask the agent to pick from one of the closest available
options. Here is an example:

Shopping Goal: i want a noise cancelling cosycost usb microphone, and price lower than 60.00 dollars

Conversation History:

Agent Current Observation: Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B09L86RDXS]
Comfortable Bluetooth Headset, UX-M97 Wireless Headset with Microphone, Wireless Cell Phone Headset with Noise Isolation
Mic Charging Base Mute Function for Xiaomi Poco F3 GT with Charging Dock
$41.95
[B092W6WNH4]
GAOMU IPX6 Waterproof Bluetooth Earbuds, True Wireless Earbuds, 20H Cyclic Playtime Headphones with Charging Case and
mic for Android, in-Ear Stereo Earphones Headset for Sport Black
$11.99
[B014C9KQLM]
ASC Audio BlueTooth A2DP + USB Flash Drive Car Stereo Adapter Interface Compatible for Honda w/Navigation- Some Vehicles
only- Compatible Vehicles Listed Below
$84.95

Agent: It seems there are no results for a "noise cancelling Cosycost USB microphone" under $60. Would you like me to
search for a different brand or type of microphone?
Your Response: Let's proceed with the first one then.

Now respond to this:

Shopping Goal: {instruction}

Conversation History: {conversation_history()}

Agent Current Observation: Observation: {agent_obs}

Agent: {agent_message}
Your Response:
```