REI-BENCH: CAN EMBODIED AGENTS UNDERSTAND VAGUE HUMAN INSTRUCTIONS IN TASK PLANNING?

Anonymous authors
Paper under double-blind review

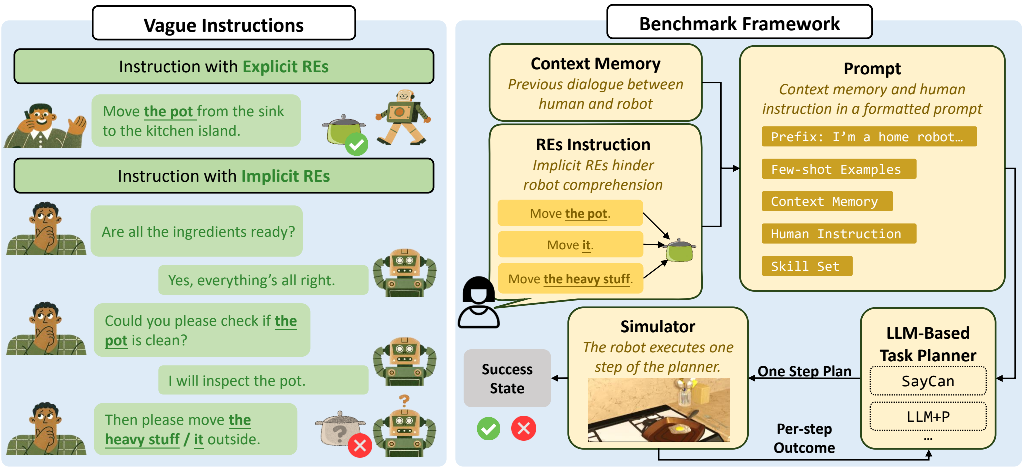

Vague Instructions
Instruction with Explicit REs
Move the pot from the sink to the kitchen island.
Instruction with Implicit REs
Are all the ingredients ready?
Yes, everything's all right.
Could you please check if the pot is clean?
I will inspect the pot.
Then please move the heavy stuff / it outside.
?
Benchmark Framework
Context Memory
Previous dialogue between human and robot
REs Instruction
Implicit REs hinder robot comprehension
Move the pot.
Move it.
Move the heavy stuff.
Prompt
Context memory and human instruction in a formatted prompt
Prefix: I'm a home robot…
Few-shot Examples
Context Memory
Human Instruction
Skill Set
Success State
Simulator
The robot executes one step of the planner.
One Step Plan
Per-step Outcome
LLM-Based Task Planner
SayCan
LLM+P
…

Figure 1: **Left**: Robots using existing LLM-based task planners can understand clear instructions with explicit referring expressions (REs), but they struggle to resolve implicit REs in multi-turn dialogues. **Right**: We propose the REI-Bench framework that aims to study real-world HRI scenarios where coreferential vagueness exists in human instructions.

ABSTRACT

Robot task planning decomposes human instructions into executable action sequences that enable robots to complete a series of complex tasks. Although recent large language model (LLM)-based task planners achieve amazing performance, they assume that human instructions are clear and straightforward. However, real-world users are not experts, and their instructions to robots often contain significant vagueness. Linguists suggest that such vagueness frequently arises from referring expressions (REs), whose meanings depend heavily on dialogue context and environment. This vagueness is even more prevalent among the elderly and children, who are the groups that robots should serve more. This paper studies how such vagueness in REs within human instructions affects LLM-based robot task planning and how to overcome this issue. To this end, we propose the first robot task planning benchmark that systematically models vague REs grounded in pragmatic theory (REI-Bench), where we discover that the vagueness of REs can severely degrade robot planning performance, leading to success rate drops of up to 36.9%. We also observe that most failure cases stem from missing objects in planners. To mitigate the REs issue, we propose a simple yet effective approach: task-oriented context cognition, which generates clear instructions for robots, achieving state-of-the-art performance compared to aware prompts, chains of thought, and in-context learning. By tackling the overlooked issue of vagueness, this work contributes to the research community by advancing real-world task planning and making robots more accessible to non-expert users, e.g., the elderly and children.

1 INTRODUCTION

In recent years, large language models (LLMs) have shown strong capabilities in tackling open-world tasks across diverse domains. Their use in robot planning indicates a promising shift: unlike traditional task planning methods (Hart et al., 1968; Aeronautiques et al., 1998; Chaslot et al., 2008), which are often constrained by specific environments and narrow task domains, LLMs enable robots to tackle tasks outside of conventional planning domains.

Although existing LLM-based task planners have shown remarkable performance (Ahn et al., 2022; Huang et al., 2022a; Wong et al., 2023), they operate under an idealized assumption: human instructions are always clear, complete, and unambiguous. However, in the real world, human language often exhibits vagueness. Figure 1 shows how vague terms like “it” or “heavy stuff” (refer to a pan) can make a kitchen robot grab the wrong item (e.g., a plate or frying pan) instead of the pot. This challenge becomes even more pronounced for individuals with impaired memory or limited expressive abilities (Robinson & Apperly, 2001; So et al., 2010; Hendriks et al., 2014), including young children, older adults, and individuals with Alzheimer’s disease—groups that rely most on robotic assistance.

Linguists suggest that various forms of linguistic vagueness can impact human-robot interaction, including syntactic, scopal, and coreferential vagueness (Li et al., 2024b). Among these, coreferential vagueness is particularly common and impactful in robot task planning, as it can affect the planner’s understanding of noun phrases in instructions and lead to incorrect identification of task objects. The coreferential vagueness arises because humans employ not only explicit referring expressions (REs) (e.g., “pot”), which directly identify their referents, but also implicit REs (e.g., “it” or “this heavy thing”), which require contextual and environmental reasoning to resolve their meaning. Unlike robots, humans naturally use and interpret implicit REs in communication (Drave, 2002; Jucker et al., 2003; Paris et al., 2021; Peter, 2018; Alkhatnai, 2017; Wasow et al., 2005). Linguists have found that about 20% of expressions in news content are descriptive (a kind of implicit REs) (Hervaś & Finlayson, 2010), with this percentage being even higher in everyday life. Bridge inference theory (Clark, 1975) explains the process by which humans resolve implicit REs. When a listener hears an expression like “this heavy stuff,” they naturally identify several possible referents based on contextual memory: the “pot,” the “ingredients,” and the “sink.” Among these, the “pot” best matches the description. This explains why the word “refer” is composed of two parts: the prefix *re-* (“back/again”) and the root *-fer* (“to carry / to bring”), which together embody the core mechanism of linguistic reference —namely, carrying meaning back from previously established contextual information. Inspired by linguistic studies, our research investigates key questions regarding task planners in embodied agents: **Do implicit REs in human instructions impact the performance of LLM-based task planners in robots? How does the frequency of implicit REs influence the success rate of these task planners? What are the underlying causes of this impact, and what strategies can be employed to mitigate it?**

To evaluate the impact of implicit REs on the success rate of planners, we first build the referring expressions instruction (REI) dataset and then propose the first benchmark that systematically models coreferential vagueness grounded in pragmatic theory for robot task planners, namely REI-Bench. In this benchmark, our work systematically models the use of REs in human-robot instructions by defining three levels of referential difficulty, based on the ratio of explicit to implicit REs. As robots need context to understand REs in human instruction, we propose three levels of real-world contexts in multi-turn dialogues with irrelevant or missing information. Combining REs’ difficulties with context memory types, the REI dataset includes nine levels of coreferential vagueness.

Then we evaluate task planners based on the REI dataset, including 6 mainstream LLMs and 4 robot planning frameworks. Because most deployed robot systems currently rely on small-scale language models, our analysis focuses on planners operating within this model regime. The results show that existing planners generally perform poorly in the presence of vagueness, with task success rates dropping between 7.4% and 36.9% in the baseline models. We attempt to mitigate the issue using novel basic NLP methods, including aware prompt (AP) (Gao et al., 2024a), chain-of-thought (CoT) (Wei et al., 2022), and in-context learning (ICL) (Brown et al., 2020), but observe limited gains. Meanwhile, we find that these performance declines in LLMs occur primarily because they devote excessive attention to plan generation while failing to perform their inherent language understanding abilities. This challenges the common assumption that simply embedding an LLM in robot planning is sufficient for the robot to understand human language. Inspired by our observation, we propose a simple yet effective approach, Task-Oriented Context Cognition (TOCC), which decouples task comprehension from the planning decision-making process. Rather than designing a fully fledged solution, we intend to draw attention from the research community to this overlooked challenge and thereby motivate deep exploration.

Our contributions are threefold: (1) We systematically study how the instruction vagueness caused by REs impacts LLM-based robot task planners. To the best of our knowledge, this work provides the first systematic modeling of instruction vagueness in the context of robot task planning. (2) We develop REI-Bench by designing different levels of REs and context memory of human-robot dialogues, studying the success rate of robot tasks with vague instructions. (3) We analyze the underlying reasons for performance degradation and propose a simple yet effective approach, TOCC, to enhance the robustness of planners. Extensive experiments on REI-Bench validate its effectiveness.

2 RELATED WORKS

Embodied Task Planning enables robots to generate action sequences for various applications, including household and industrial tasks. Recent developments in LLM (Brown et al., 2020; Touvron et al., 2023) have led to language-driven planning methods (Kotb et al., 2024; Driess et al., 2023; Brohan et al., 2023; Chen et al., 2023; Liu et al., 2024a; Li et al., 2023; Huang et al., 2023), such as SayCan (Ahn et al., 2022), which leverage affordance functions to generate policies without fine-tuning. However, evaluating these planners becomes difficult due to the high cost of real-world deployment, including expensive robotic hardware and resources. The early studies rely on human evaluation (Ahn et al., 2022; Huang et al., 2022b), which was subjective and inefficient. To address this, automated evaluation with simulators has emerged (Yin et al., 2024; Liu et al., 2024b). Methods such as ProgPrompt (Singh et al., 2023) and LoTA-Bench (Choi et al., 2024) leverage datasets like VirtualHome (Puig et al., 2018) and AI2-THOR (Kolve et al., 2017). Meanwhile, Embodied Agent Interface (Li et al., 2024a) offers a general framework for evaluating LLMs.

Linguistic Vagueness in LLMs has attracted attention among researchers in natural language processing (Liu et al., 2023b; Ortega-Martín et al., 2023). The AmbiEnt benchmark (Liu et al., 2023a) evaluates the ability of LLMs to resolve linguistic ambiguities, revealing significant challenges even for advanced models such as GPT-4. To address these issues, APA (Kim et al., 2024) improves the management of vague queries by LLMs by leveraging their self-assessment of vagueness. In the field of embodied AI, some prior works try to address these ambiguities by having robots ask clarifications or reason about uncertainty (Doğan et al., 2022; Park et al., 2023; Ren et al., 2023). However, these works lack a systematic definition of linguistic vagueness, a comprehensive evaluation of its impact on robotic performance, and effective methods to enhance the planner's own understanding capability. Building upon these developments, we further compare existing datasets and task planning benchmarks with linguistic ambiguity in Table 1.

Table 1: Comparison of REI-Bench with existing datasets and benchmarks.

Benchmark / Method	**Task**	**Planning**	**Systematic Vagueness**	**Multi-turn Context**	**Dataset / Size**
REI-Bench (ours)	Evaluating coreferential vague instructions (RE-based) for robot task planning in AI2-THOR	✓	✓	✓	REI-Dataset / 2.7k instructions (× 9 vagueness levels)
AmbiK (Ivanova et al., 2025)	Ambiguous natural language task instructions for robot planning in a kitchen environment	×	✓	×	AmbiK / 1k ambiguous instructions
CLARA (Park et al., 2023)	Method for LLMs to classify whether the command is certain or not	×	✓	×	SaGC / 105 goals, 5222 tasks
KNOWNO (Ren et al., 2023)	Framework for measuring and aligning the uncertainty of LLM-based planners	✓	×	×	No dataset proposed
DialFRED (Gao et al., 2022)	Questioner performer framework	✓	×	✓	53K task-relevant questions and answers
Asking Clarifications (Xu et al., 2019)	Clarification identification, clarification question generation, and answering for ambiguous language	×	✓	✓	CLAQUA / 40K dialogue words

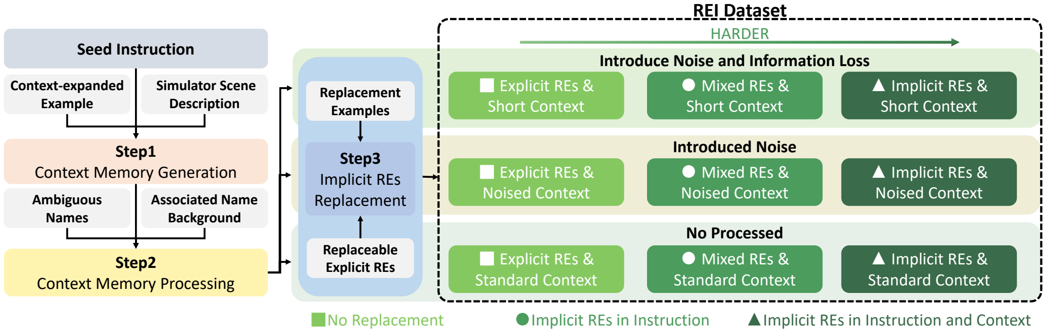

REI Dataset
HARDER
Seed Instruction
Context-expanded Example
Simulator Scene Description
Step1
Context Memory Generation
Ambiguous Names
Associated Name Background
Step2
Context Memory Processing
Replacement Examples
Step3
Implicit REs Replacement
Replaceable Explicit REs
Introduce Noise and Information Loss
Explicit REs & Short Context
Mixed REs & Short Context
Implicit REs & Short Context
Introduced Noise
Explicit REs & Noised Context
Mixed REs & Noised Context
Implicit REs & Noised Context
No Processed
Explicit REs & Standard Context
Mixed REs & Standard Context
Implicit REs & Standard Context
No Replacement
Implicit REs in Instruction
Implicit REs in Instruction and Context

Figure 2: **Data curation pipeline of the REI dataset.** From a seed instruction, we (1) generate context memory; (2) produce three context variants—Standard, Noised, Short; (3) replace explicit REs with implicit ones across varying degrees. This results in subsets reflecting nine levels of coreferential vagueness, determined by RE types (Explicit/Mixed/Implicit) and context variants.

3 APPROACH

Existing works Choi et al. (2024) typically evaluate LLM-based robot task planners using clear instructions, whereas human instructions in real-world HRI scenarios are often ambiguous. In natural language, instruction vagueness arises from the one-to-many relationship between a *Signifier* (the symbol itself) and its *Signifieds* (the entities it may represent in the physical world) (Hervaś & Finlayson, 2010). For example, the signifier ‘mouse’ can refer to two signifieds: ‘a rodent’ or ‘a computer input device’. Pragmatics scholar Levinson et al. (Levinson, 1983) further distinguish vagueness into two types: *Referential Expressions (REs)* and *Deictic Expressions (DEs)*. Humans can interpret REs through language context. For example, “Please bring me the mouse and keyboard” (referring to the computer input devices). In contrast, understanding DEs depends on environmental context, such as time, space, and the speaker’s position, for example, “I want the book on the right side”. Given our focus on LLM-based task planners, we confine our discussion to the impact of REs on task planning performance.

The objective of our work is to comprehensively evaluate and analyze how different levels of coreferential vagueness from implicit REs affect the planner’s performance across diverse multi-turn dialogue contexts. To this end, we (1) systematically formalize REs in the HRI context (section 3.1), (2) establish the REI dataset and benchmark to evaluate planners on embodied tasks involving vague instructions (section 3.2), and (3) introduce a simple yet effective solution (section 3.3).

3.1 FORMALIZING VAGUENESS BY IMPLICIT REs AND HUMAN-ROBOT DIALOGUE CONTEXT

Levinson et al. (Levinson, 1983) propose that understanding humans’ intention depends both on *Referring Expressions* and *Context Memory*, where context memory refers to the previous dialogues which provide hints to determine the unique meaning of REs. Here, we systematically model these two concepts into three levels to simulate varying degrees of vagueness.

Referring Expressions. In robot task planning, understanding REs is essential, as they provide key semantic cues for the goals of embodied tasks. Specifically, REs take various forms, including proper nouns (e.g., “apple”), definite noun phrases (e.g., “the apple”), indefinite noun phrases (e.g., “an apple”), pronouns (e.g., “it”, “them”), and attributive expressions (e.g., “sweet fruit”). The first three forms, known as *explicit REs*, have only one potential corresponding object and can be directly understood. In contrast, the latter two forms, known as *implicit REs*, have multiple potential corresponding objects and should be identified by contextual inference. For example, the pronoun “it” simply indicates the referent is an object rather than a person, while “red fruit“ only specifies the type and color of the item, without identifying which specific fruit it is.

To systematically model different forms of REs, we categorize them into three levels (Figure 2). At the “Explicit REs” level, all expressions are preserved as they appear in the original dataset. At the “Mixed” level, explicit REs in the instruction are replaced with implicit ones, while those in the context memory remain unchanged. The planner should refer to explicit REs to infer their implicit counterparts. At the “Implicit REs” level, all explicit REs are replaced with implicit ones, except the first one in context memory, forcing the planner to rely on scene information to identify the referred objects. The latter two levels essentially simulate the vagueness introduced by implicit REs, which are more common in daily human communication.

Context Memory. Linguists argue that the connection between words and objects is constructed by humans within specific contexts (Levinson, 1983). Analogously, different types of context can affect the reasoning capability of a robot due to the inclusion of misleading cues or the omission of semantic information. One common source of misleading cues arises when a single signifier may plausibly refer to multiple, context-dependent entities, creating naming ambiguity. For example, the word "apple" may shift from denoting a fruit to referring to a mobile phone brand when such mentions appear in the discourse. Additionally, semantic omissions can arise from the speaker's health issues or from different stages of cognitive development.

To simulate these scenarios, we deliberately introduce irrelevant information to the context while removing certain cues. Specifically, we define three types of context memory. In the "standard context", all information related to the task in context is provided. In "Noised Context", we introduce the *Ambiguous Name* noise, defined as a character or brand with a name intentionally resembling that of an object in the simulator scene. For example, "Apple" as a brand name is repeatedly mentioned in the dialogue, causing the planner to treat the fruit in the scene as the target. This noise reflects real-world cases where names are the same as objects, and tests the model's ability to correctly identify the intended referent. Moreover, the "Short Context" not only introduces noise but also omits partial task-relevant information, further increasing the difficulty of reasoning.

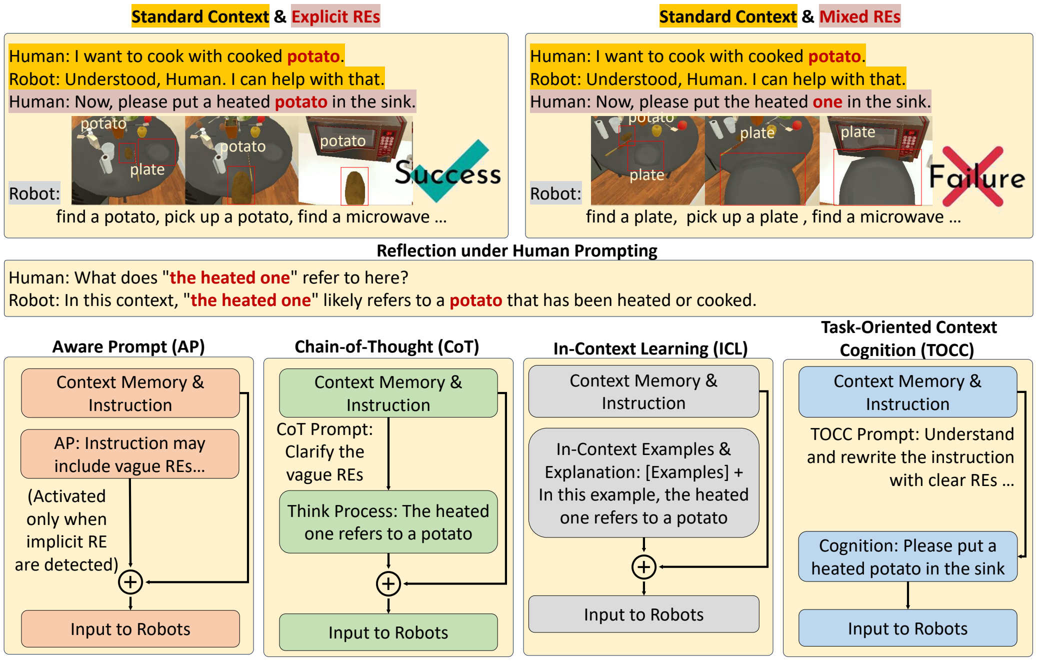
3.2 REI-BENCH DATASET

Existing vague expressions datasets (Marcus et al., 2011; Levesque et al., 2012; Recasens et al., 2010), which are annotated by linguists, do not systematically formulate the position, frequency, and forms of REs, making it infeasible to investigate their impact on task planning. Thus, we establish a comprehensive referring expressions instruction (REI) benchmark via an automatic pipeline to assess the effects of implicit REs on robotic planning tasks.

Specifically, we build the REI-Bench dataset upon ALFRED (Shridhar et al., 2020), a benchmark for embodied household tasks. From ALFRED, we select six tasks (Pick & Place, Stack & Place, Clean & Place, Heat & Place, Cool & Place, Examine in Light) and exclude the task of Pick Two & Place as it cannot be reliably completed by an embodied agent. Furthermore, since tasks that cannot be accomplished even with clear instructions fall outside our scope, we use "LLaMA3.1-8B + SayCan" to perform the six selected tasks in the AI2-THOR (Kolve et al., 2017) simulator, keeping only the successfully executed tasks as seed instructions.

As shown in Figure 2, context memory is generated (Step 1), processed into three types (Step 2), and further transformed to the REI-Bench through implicit REs replacement (Step 3). In step 1, we extend the context of seed instructions by prompting GPT-4o-mini with a template that consists of a context-expanded example and the text-based simulator scene description. The generated data consists of an instruction and a corresponding context memory. Typically, the generated instruction conveys the same task requirement as the seed instruction but in a more natural, human-like form. Meanwhile, the context memory captures the multi-turn human-robot dialogue before the instructions, which may include any objects present in the scene to reflect the complexity of real-world human dialogue.

In step 2, we construct three types of context. The "standard context" retains the context memory generated in step 1 without further processing. For the "Noised Context" type, LLM is prompted to repeatedly insert an ambiguous name into the dialogue without altering its meaning. The ambiguous name is derived by slightly modifying the name of a simulator object (e.g., Rose → Mrs. Rose). To ensure natural adaptation, the LLM is given a brief background prompt for the name (e.g., "Rose is a family member."). In the "short context" setting, partial noun phrases are randomly removed from the context, including those containing task-relevant explicit REs. In Step 3, explicit REs are replaced with implicit ones following the rules outlined in Section 3.1. We adopt a CoT approach to determine which explicit REs can be substituted. Substitution examples are selected from OntoNotes (Pradhan et al., 2013), ensuring that the implicit REs are consistent with natural language usage. To ensure consistency in the number of explicit REs across tasks, we define a counting-based rule for each level of implicit REs and discard any data that violates the rule. Consequently, the REI-Bench consists of


Standard Context & Explicit REs
Standard Context & Mixed REs
potato
plate
Success
Failure
Reflection under Human Prompting
Aware Prompt (AP)
Chain-of-Thought (CoT)
In-Context Learning (ICL)
Task-Oriented Context Cognition (TOCC)
Context Memory & Instruction
AP: Instruction may include vague REs...
(Activated only when implicit RE are detected)
Input to Robots
CoT Prompt: Clarify the vague REs
Think Process: The heated one refers to a potato
In-Context Examples & Explanation: [Examples] + In this example, the heated one refers to a potato
TOCC Prompt: Understand and rewrite the instruction with clear REs ...
Cognition: Please put a heated potato in the sink

Human: I want to cook with cooked potato.
Robot: Understood, Human. I can help with that.
Human: Now, please put a heated potato in the sink.
Robot:
find a potato, pick up a potato, find a microwave ...

Human: I want to cook with cooked potato.
Robot: Understood, Human. I can help with that.
Human: Now, please put the heated one in the sink.
Robot:
find a plate, pick up a plate , find a microwave ...

Human: What does "the heated one" refer to here?
Robot: In this context, "the heated one" likely refers to a potato that has been heated or cooked.

Figure 3: **Addressing implicit referring expressions in task planning.** Top row: LLM succeeds with explicit REs ("potato"), but misidentifies the object with implicit REs ("the heated one"). Middle row: a reflection prompt from humans can guide the LLM to resolve the implicit REs and identify the correct object. Bottom row: Comparison among different prompting methods, including aware prompt (AP), chain-of-thought (CoT), in-context learning (ICL), and our task-oriented context cognition (TOCC).

2,700 examples spanning nine difficulty levels, defined by combinations of varying RE vagueness and context memory conditions. Please refer to Appendix B for detailed prompts of context expansion (B.1), context process (B.2), explicit REs replacement (B.3), and the counting-based rule (B.4).

3.3 TASK-ORIENTED CONTEXT COGNITION

Based on the evaluation of multiple LLM-based robot planners in REI-Bench, we find that most failures result from object omissions, as illustrated in the top row of Figure 3. When explicit REs are presented in the instruction, the robot correctly identifies the task target "potato" and successfully completes the planning. In contrast, when implicit REs, such as "the heated one", are used, robots fail to identify "potato" but a task-irrelevant object "plate" instead. Furthermore, we find that LLMs can resolve implicit REs when prompted explicitly (shown in the middle row of Figure 3). This suggests that LLMs inherently can interpret implicit REs, yet this ability can not fully manifest during planning and requires explicit prompting. As a result, the idealized expectation that embedding an LLM into embodied agents guarantees full comprehension of human language has been challenged.

To this end, we propose to inject explicit prompts into the LLM-based robot planners to alleviate the coreferential vagueness. Specifically, we first evaluate three conventional prompting methods: (1) aware prompt (AP) (Gao et al., 2024a), which explicitly adds a prompt to guide the planner detect potential REs, (2) Chain-of-Thought (CoT) (Wei et al., 2022), which guides the planner to resolve REs step by step before planning, and (3) In-Context Learning (ICL) (Brown et al., 2020), which provides examples of how to infer implicit REs from context. However, AP remains insufficient for handling implicit REs, as the prompt signals vagueness but does not induce the deeper reasoning that LLMs struggle to perform during planning. Meanwhile, both CoT and ICL substantially lengthen prompts, hindering language understanding during planning, particularly when onboard computing resources are limited and only a small language model is available.

Consequently, we propose a simple yet effective method, task-oriented context cognition (TOCC) to tackle the REs. As shown in the bottom row of Figure 3, TOCC decouples the REs interpretation step from the planning process, avoiding the LLM from devoting excessive attention to planning

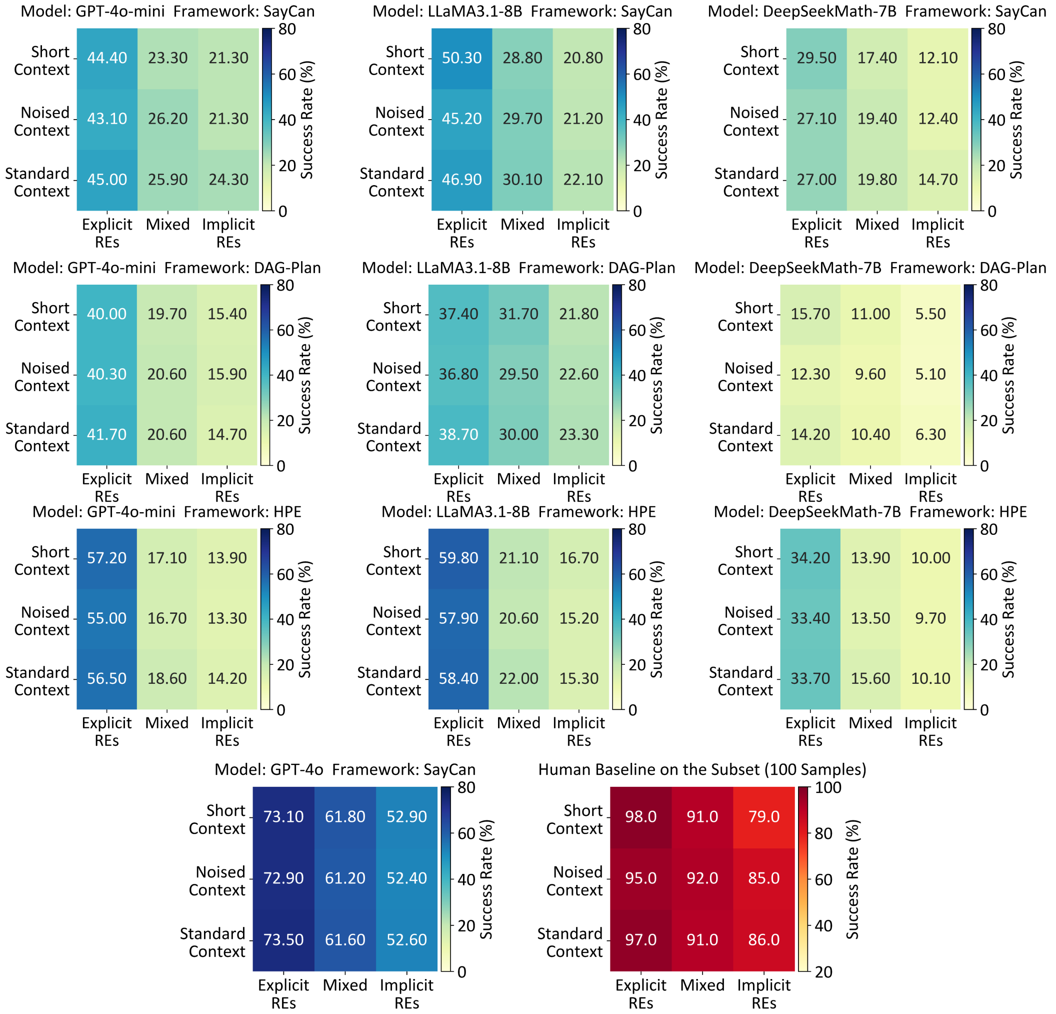


Figure 4: Success rate (%) of three task planner frameworks, SayCan, DAG-Plan, and HPE, using three LLMs (GPT-4o-mini, LLaMA3.1-8B, DeepSeekMath-7B), together with an additional "GPT-4o + SayCan" planner and a human baseline on the REI dataset. Explicit, Mixed, and Implicit REs denote three levels of implicit REs in human instructions, and Standard, Noised, and Short Contexts represent three context memory types.

within a single generation step. By resolving the vague REs and rephrasing the human instruction in a more concise form before planning, TOCC demonstrates superior performance compared to existing methods. Implementation details and the exact variants used for AP and CoT are provided in Appendix C.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

We evaluate two state-of-the-art LLM-based embodied task planning frameworks, SayCan (Ahn et al., 2022), DAG-Plan (Gao et al., 2024b), and hierarchical task planning and execution (HPE) (Han et al., 2025), on our REI-Bench dataset. Due to the deployment requirements on mobile robots, which must be lightweight and open-source for on-robot adaptation, we focus solely on relatively small language models. For each task planner, we evaluate six LLMs, including GPT-4o-mini (Achiam et al., 2023), LLaMA3.1-8B (Grattafiori et al., 2024), Ministral-8B (Jiang et al., 2024), Gemma2-9B (Team et al., 2024), DeepSeek-Math-7B (Shao et al., 2024), and Qwen2.5-7B (Bai et al., 2023), which form a comprehensive benchmark consisting of 12 planners in total. To ensure balanced task coverage

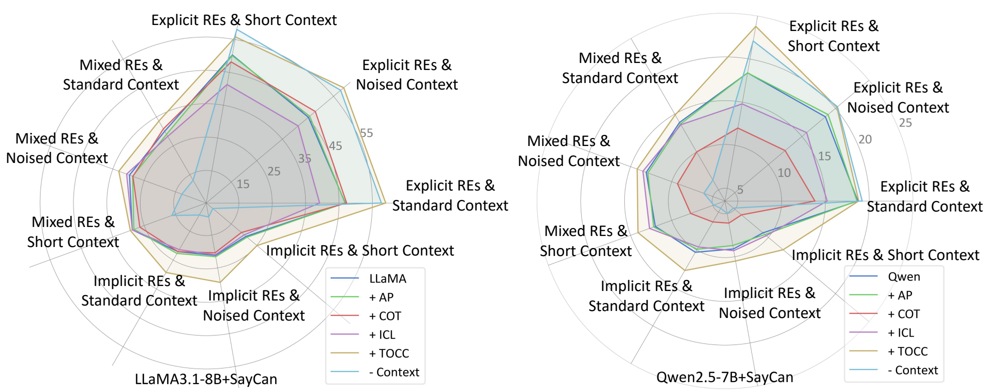


Figure 5: Success rates (%) of various prompting methods applied to LLaMA 3.1-8B and Qwen2.5-7B models with SayCan framework on the REI dataset.

during evaluation, we first construct a 1,000-task subset of REI-Bench via stratified sampling over task categories, and use this subset for all planner evaluations. To further compare the performance of LLM-based task planners with humans, we invite human volunteers to conduct the same planning tasks on a randomly sampled subset of the REI-Bench.

4.2 BENCHMARK RESULTS OF LLM-BASED TASK PLANNERS

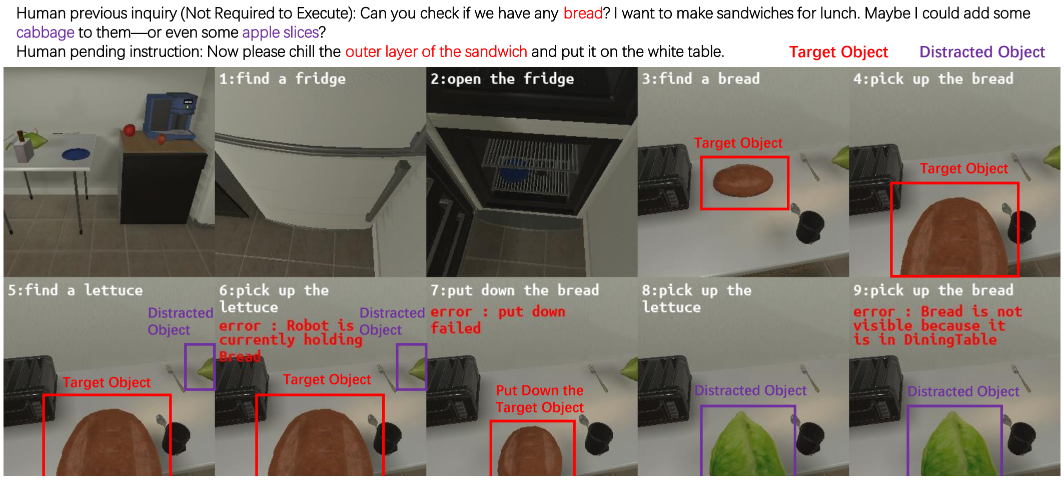
We evaluate the planning performance of all benchmark models on the REI dataset. Due to page limitations, we present only six benchmark results in Figure 4. Please refer to the appendix A.3 for additional results.

LLM-based planners struggle to handle embodied tasks in multi-turn dialogue. We use the instruction portion (excluding the context memory) of the “Explicit REs + Standard Context” type of data as a baseline, for which the “LLaMA3.1-8B+SayCan” planner achieves a 57.7% success rate. However, as shown in the middle of the top row in Figure 4, multi-turn dialogues in “Standard Context” cause the success rate of “LLaMA3.1-8B+SayCan” model to drop significantly from 57.7% to 46.90%, even without implicit REs. In contrast, humans achieve a 97.0% success rate under the same setting. This performance gap highlights the limitations of existing LLM-based planners in handling natural, multi-turn human conversations.

The performance of LLM-based planners consistently decreases as implicit REs increase, while remaining less affected by context memory noise. With the increase of implicit REs, all benchmark planners demonstrate consistent performance degradation across “Standard”, “Noised”, and “Short” context memory. Take “LLaMA3.1-8B+SayCan” (middle of the top row in Figure 4) as an example, the success rate drops 16.8% / 15.5% / 21.5% at the “Mixed REs” level and further decreases 8.0% / 8.5% / 8.0 % at the “implicit REs” level. The consistent declines in LLMs’ ability demonstrate that existing LLM-based planners cannot effectively handle the vagueness of implicit REs within human instructions for embodied tasks. In addition, compared to introducing multi-turn dialogues, adding naming ambiguity noise (“Noised Context”) and further omitting partial task-relevant information (“Short Context”) has little impact on performance. These observations suggest that existing LLM-based planners perform poorly when faced with multi-turn dialogues and implicit REs. However, such challenges are ubiquitous in human–robot interaction, especially when engaging with the elderly or children, and thus must be addressed.

4.3 ABLATION STUDY ON DIFFERENT PROMPTING METHODS

We compare four prompting methods that can mitigate the impact of implicit REs for LLM-based task planners: AP, CoT, ICL, and TOCC. As shown in Figure 5, by directly prompting the LLM-based planners that human instructions contain potential vagueness, AP improves performance in most scenarios. However, in some of the “Explicit REs”, the performance decreases inversely. We deduce that these APs may lead the planner to hallucinate and detect vagueness even when instructions are clear. Meanwhile, ICL provides examples to help the planner infer the meaning of implicit



Human previous inquiry (Not Required to Execute): Can you check if we have any bread? I want to make sandwiches for lunch. Maybe I could add some cabbage to them—or even some apple slices?

Human pending instruction: Now please chill the outer layer of the sandwich and put it on the white table.

Figure 6: Failure example on "Mixed REs & Short Context", using LLaMA3.1-8B+SayCan. Due to the distracted object, the planner mistakenly put down the target object.

REs from context. However, ICL leads to performance degradation in almost all categories. We deduce that the small onboarding LLM-based planners possess limited capability in learning from the provided examples. Furthermore, CoT first guides the planner to autonomously analyse whether the instructions contain implicit REs and then perform planning based on the analysis, resulting in greater improvement. Building on the autonomous analysis from CoT, TOCC enables the planner to provide more refined and task-relevant instructions by correcting implicit REs and reorganizing language. Planning is then performed based on the enhanced instruction. By decoupling RE resolving from planning, TOCC obtains the best performance, with an average success rate improvement of 6.5% on "LLaMA3.1-8B+SayCan". We also provide results using only instructions as input. Under the Explicit REs type, the planner achieved performance comparable to TOCC. However, under the Mixed REs and Implicit REs types, planner performance dropped sharply. Compared to the original instructions, the enhanced instructions from our TOCC achieve improved planning performance. Moreover, the results are consistent with the pragmatic theory that context is indispensable for interpreting implicit REs. Please refer to appendix A.4 for completed ablation results.

4.4 ANALYSIS OF LLM-BASED PLANNER ERRORS

In this section, we review error cases made by LLM-based planners in processing implicit REs. As shown in Figure 6, the planner was uncertain whether "outer layer of the sandwich" referred to bread or lettuce, revealing a limitation of existing planners. Additional cases are provided in appendix A.6.

Table 2: Error rates (%) for the object omission and execution error types in different benchmark models. Results under the "Standard Context" for three types of implicit REs are reported.

Model	Implicit REs Level	Error Type		Overall Error Rate
		Object Omission Rate	Execution Error Rate	
GPT-4o-mini	Explicit REs	7.1	47.9	55.0
	Mixed	37.0 (+29.9)	37.1 (−10.8)	74.1 (+19.1)
	Implicit REs	46.2 (+39.1)	29.5 (−18.4)	75.7 (+20.7)
LLaMA3.1-8B	Explicit REs	22.6	30.5	53.1
	Mixed	38.8 (+16.2)	31.1 (−0.6)	69.9 (+16.8)
	Implicit REs	53.9 (+31.3)	24.0 (−6.5)	77.9 (+24.8)
Deepseek-8B	Explicit REs	28.6	44.4	73.0
	Mixed	40.8 (+12.2)	39.4 (−5.0)	80.2 (+7.2)
	Implicit REs	57.8 (+29.2)	27.5 (−16.9)	85.3 (+12.3)

For an in-depth analysis, we divide the task planning errors into two categories: object omission and execution error. Object omission refers to the planner not correctly identifying the target object in human instructions. As shown in Figure 3, the planner wrongly identifies the referring expression "one" to "plate" as a typical object. In addition, an execution error occurs when the planner identifies the correct object but cannot generate the complete sequence of actions to achieve the goal. We

Table 3: Error rates (%) for the object omission and execution error types under different prompting methods (for LLaMA3.1-8B with "Standard Context").

Method	Implicit REs Level	Error Type		Overall Error Rate
		Object Omission Rate	Execution Error Rate	
LLaMA3.1-8B	Explicit REs	22.6	30.5	53.1
	Mixed	38.8	31.1	69.9
	Implicit REs	53.9	24.0	77.9
+ AP	Explicit REs	22.7 (+0.1)	30.5	53.2 (+0.1)
	Mixed	31.3 (−7.5)	39.7	71.0 (+1.1)
	Implicit REs	49.9 (−4.0)	27.4	77.3 (−0.6)
+ CoT	Explicit REs	22.5 (−0.1)	30.2	52.7 (−0.4)
	Mixed	34.9 (−3.9)	34.2	69.1 (−0.8)
	Implicit REs	47.6 (−6.3)	30.3	77.9 (+0)
+ ICL	Explicit REs	22.7 (+0.1)	38.1	60.8 (+7.7)
	Mixed	32.7 (−6.1)	39.0	71.7 (+1.8)
	Implicit REs	49.9 (−4.0)	28.7	78.6 (+0.7)
+ TOCC	Explicit REs	**16.8** (−5.8)	24.2	41.0 (−12.1)
	Mixed	**28.5** (−10.3)	37.9	66.4 (−3.5)
	Implicit REs	**40.1** (−13.8)	30.6	70.7 (−7.2)
- Context	Explicit REs	17.2 (−5.4)	25.1	42.3 (−10.8)
	Mixed	81.6 (+42.8)	5.3	86.9 (+17)
	Implicit REs	85.1 (+31.2)	5.5	90.6 (+12.7)

summarize the error rate related to object omission and execution error for different benchmark models in Table 2. For simplicity, only the results under the "Standard Context" are reported. Please refer to the appendix A.5 for more results under other context memory types. It can be seen that the overall error rate increases as the level of implicit REs grows. However, the error rates for object omission and execution error show opposite trends: the former increases significantly, while the latter decreases as the level of implicit REs grows. This indicates that the main cause of task planning errors is that the implicit REs induce the task planner to overlook target objects in HRI, thus making it unable to generate a task plan correctly. Furthermore, as shown in Table 3, our TOCC effectively reduces both the overall error rate and the object omission error rate across all three levels of implicit REs. These results demonstrate that TOCC effectively guides task planners to focus on target objects in human instructions, thereby enhancing robustness to finish instructions with vague REs.

5 CONCLUSION

We study how coreferential vagueness in human instruction affects robot task planning. By REI-Bench, we systematically simulate real-world language vagueness by categorizing REs and context memory. Extensive experiments show that implicit REs significantly reduce planning success rates. We explore the underlying reason and introduce the TOCC method, which effectively mitigates the negative effect of coreferential vagueness on robot task planner performance.

While our work discusses the impact of coreferential vagueness, human language vagueness is pervasive, and other forms of linguistic vagueness, such as deictic expressions, syntactic vagueness, and scopal vagueness, remain largely unexplored in the context of robot task planning. Furthermore, to isolate the effect of REs, we filter the dataset by selecting tasks that LLMs can complete under clear instructions. As a result, the dataset consists of simple, short-horizon, single-objective tasks. As future LLM-based planners become capable of solving more complex, clear-instruction tasks, we plan to extend our analysis to long-horizon scenarios. Moreover, experiments in the AI2-THOR simulator provide initial results of the planner's semantic understanding capabilities. They do not capture multimodal information, including visual and spatial perception, which is required for robots to interpret other types of vague instructions. Thus, our future work will focus on investigating the impact of deictic expressions on VLM-based task planners and validating the findings through experiments with physical robots.

REFERENCES

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

Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins Sri, Anthony Barrett, Dave Christianson, et al. Pddl— the planning domain definition language. *Technical Report, Tech. Rep.*, 1998.

Michael Ahn, Anthony Brohan, Noah Brown, and ... Do as i can, not as i say: Grounding language in robotic affordances, 2022. arXiv preprint arXiv:2204.01691.

Mubarak Alkhatnai. Vague language and its social role. *Theory and Practice in Language Studies*, 7 (2):122–127, 2017.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen Technical Report, 2023. arXiv preprint arXiv:2309.16609.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. arXiv preprint arXiv:2307.15818.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. Monte-carlo tree search: A new framework for game ai. In *AAAI*, pp. 216–217, 2008.

Yaran Chen, Wenbo Cui, Yuanwen Chen, Mining Tan, Xinyao Zhang, Dongbin Zhao, and He Wang. Robogpt: An intelligent agent for making embodied long-term decisions for daily instruction tasks, 2023. arXiv preprint arXiv:2311.15649.

Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. Lota-bench: Benchmarking language-oriented task planners for embodied agents, 2024. arXiv preprint arXiv:2402.08178.

Herbert H Clark. Bridging. *Theoretical issues in natural language processing/Association for Computing Machinery*, 1975.

Fethiye Irmak Doğan, Ilaria Torre, and Iolanda Leite. Asking follow-up clarifications to resolve ambiguities in human-robot conversation. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 461–469. IEEE, 2022.

Neil Drave. Vaguely speaking: A corpus approach to vague language in intercultural conversations. In *New frontiers of corpus research*, pp. 25–40. Brill, 2002.

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

Jin Gao, Lei Gan, Yuankai Li, Yixin Ye, and Dequan Wang. Dissecting dissonance: Benchmarking large multimodal models against self-contradictory instructions. In *ECCV*, pp. 404–420. Springer, 2024a.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056, 2022.

Zeyu Gao, Yao Mu, Jinye Qu, Mengkang Hu, Shijia Peng, Chengkai Hou, Lingyue Guo, Ping Luo, Shanghang Zhang, and Yanfeng Lu. Dag-plan: Generating directed acyclic dependency graphs for dual-arm cooperative planning. *arXiv preprint arXiv:2406.09953*, 2024b.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The LLaMA 3 Herd of Models, 2024. arXiv preprint arXiv:2407.21783.

Songhao Han, Boxiang Qiu, Yue Liao, Siyuan Huang, Chen Gao, Shuicheng Yan, and Si Liu. Robocerebra: A large-scale benchmark for long-horizon robotic manipulation evaluation. *arXiv preprint arXiv:2506.06677*, 2025.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Petra Hendriks, Charlotte Koster, and John CJ Hoeks. Referential choice across the lifespan: Why children and elderly adults produce ambiguous pronouns. *Language, cognition and neuroscience*, 29(4):391–407, 2014.

Raquel Hervaś and Mark Alan Finlayson. The prevalence of descriptive referring expressions in news and narrative. In *ACL*. © Association for Computational Linguistics, 2010.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world, 2023. arXiv preprint arXiv:2311.12871.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *ICML*, pp. 9118–9147, 2022a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models, 2022b.

Anastasia Ivanova, Bakaeva Eva, Zoya Volovikova, Alexey Kovalev, and Aleksandr Panov. Ambik: Dataset of ambiguous tasks in kitchen environment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33216–33241, 2025.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts, 2024. arXiv preprint arXiv:2401.04088.

Andreas H Jucker, Sara W Smith, and Tanja Lüdge. Interactive aspects of vagueness in conversation. *Journal of pragmatics*, 35(12):1737–1769, 2003.

Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang-goo Lee, and Taeuk Kim. Aligning Language Models to Explicitly Handle Ambiguity, 2024. arXiv preprint arXiv:2404.11972.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Mostafa Kotb, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. Qt-tdm: Planning with transformer dynamics model and autoregressive q-learning. *IEEE Robotics and Automation Letters*, 2024.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 552–561, 2012.

Stephen C Levinson. *Pragmatics*. Cambridge University Press, 1983.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making, 2024a. arXiv preprint arXiv:2410.07166.

Margaret Y Li, Alisa Liu, Zhaofeng Wu, and Noah A Smith. A Taxonomy of Ambiguity Types for NLP, 2024b. arXiv preprint arXiv:2403.14072.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators, 2023. arXiv preprint arXiv:2311.01378.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. We're Afraid Language Models Aren't Modeling Ambiguity, 2023a. arXiv preprint arXiv:2304.14399.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. *arXiv preprint arXiv:2304.14399*, 2023b.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation, 2024a. arXiv preprint arXiv:2406.04339.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *ECCV*, pp. 386–403. Springer, 2024b.

Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, et al. Ontonotes: A large training corpus for enhanced processing. In Joseph Olive, Caitlin Christianson, and John McCary (eds.), *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, pp. 27–48. Springer, 2011.

Miguel Ortega-Martín, Óscar García-Sierra, Alfonso Ardoiz, Jorge Álvarez, Juan Carlos Armenteros, and Adrián Alonso. Linguistic ambiguity analysis in chatgpt, 2023. arXiv preprint arXiv:2302.06426.

Pierre-Henri Paris, Syrine El Aoud, and Fabian Suchanek. The vagueness of vagueness in noun phrases. In *AKBC*, 2021.

Jeongeun Park, Seungwon Lim, Joonhyung Lee, Sangbeom Park, Minsuk Chang, Youngjae Yu, and Sungjoon Choi. Clara: classifying and disambiguating user commands for reliable interactive robotic agents. *IEEE Robotics and Automation Letters*, 9(2):1059–1066, 2023.

Mcgee Peter. Vague language as a means of avoiding controversy. *Training, Language and Culture*, 2(2):40–54, 2018.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pp. 143–152, 2013.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, pp. 8494–8502, 2018.

Marta Recasens, Lluís Màrquez, Emili Sapena, et al. Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pp. 1–8, 2010.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

EJ Robinson and Ian A Apperly. Children's difficulties with partial representations in ambiguous messages and referentially opaque contexts. *Cognitive Development*, 16(1):595–615, 2001.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. arXiv preprint arXiv:2402.03300.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, pp. 10740–10749, 2020.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *ICRA*, pp. 11523–11530. IEEE, 2023.

Wing Chee So, Özlem Ece Demir, and Susan Goldin-Meadow. When speech is ambiguous, gesture steps in: Sensitivity to discourse-pragmatic principles in early childhood. *Applied psycholinguistics*, 31(1):209–224, 2010.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving Open Language Models at a Practical Size, 2024. arXiv preprint arXiv:2408.00118.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models, 2023. arXiv preprint arXiv:2302.13971.

Thomas Wasow, Amy Perfors, and David Beaver. The puzzle of ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe*, pp. 265–282, 2005.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Lionel Wong, Jiayuan Mao, Pratyusha Sharma, Zachary S. Siegel, Jiahai Feng, Noa Korneev, Joshua B. Tenenbaum, and Jacob Andreas. Learning adaptive planning representations with natural language guidance, 2023.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. Asking clarification questions in knowledge-based question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1618–1629, 2019.

Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. Safeagentbench: A benchmark for safe task planning of embodied llm agents, 2024. arXiv preprint arXiv:2412.13178.

APPENDICES

Within this supplementary material, we elaborate on the following aspects:

- **Appendix A:** Supplementary Experiment Results
  - **A.1:** Framework of Planners
  - **A.2:** Language Models List and Sampled-Subset Task-Type Distribution
  - **A.3:** Supplementary Benchmark Results
  - **A.4:** Supplementary Result of Ablation Study on Method
  - **A.5:** Supplementary Result of LLM-based Planner Errors
  - **A.6:** Supplementary of Task Planning Cases
- **Appendix B:** REI Dataset Construction
  - **B.1:** Context Memory Generation
  - **B.2:** Context Memory Processing
  - **B.3:** Implicit REs Replacement
  - **B.4:** Data Filtering
- **Appendix C:** Prompts and Implementation Details of Prompting Methods
  - **C.1:** AP and Gated AP Variant
  - **C.2:** Chain-of-Thought
  - **C.3:** In-Context Learning
  - **C.4:** Task-Oriented Context Cognition
- **Appendix D:** Use of Large Language Models

A SUPPLEMENTARY EXPERIMENT RESULTS

A.1 FRAMEWORK OF PLANNERS

SayCans is a method designed to help robots understand and carry out human instructions expressed in natural language. It breaks down a complex instruction into smaller steps suggested by the LLM, and then evaluates whether each step is possible in the real world with affordance-based value function (a separate model trained on data about how robots interact with their environment). With the emergence of more capable language tools, we employ Guidance (Choi et al., 2024) to replace the affordance value function with LLM-based feasibility assessment, which allows selecting a skill in one generation pass and significantly reduces experiment time.

LLM+P is a hybrid framework that integrates large language models (LLMs) with classical symbolic planners to achieve robust and interpretable task planning from natural language instructions. The process typically involves three steps: First, the LLM translates high-level natural language commands into formal representations such as PDDL (Planning Domain Definition Language). Second, a classical planner, such as Fast Downward, computes a valid or optimal plan based on the generated problem and domain definitions. Finally, the LLM translated the resulting plan, a sequence of low-level actions, back into natural language, making it more interpretable for users.

DAG-Plan is a planning framework in which an LLM generates a Directed Acyclic Graph (DAG) of sub-tasks rather than a linear sequence. Each node represents a symbolic high-level action, and edges denote dependency relations. This structure makes the plan explicitly hierarchical and ensures that prerequisite conditions are satisfied before execution. By modeling sub-task dependencies, DAG-Plan improves robustness on multi-object and multi-step tasks.

HPE (Hierarchical Planning with Episodic memory) equips an LLM planner with a lightweight memory bank that stores key contextual information throughout reasoning. Instead of relying solely on the immediate prompt, the model retrieves and updates this memory to maintain coherence over long horizons. However, HPE still relies on manually structured memory representations for context management, which can restrict the LLM's ability to extract subtle contextual cues, particularly those needed for resolving implicit referring expressions.

A.2 LANGUAGE MODELS LIST AND SAMPLED-SUBSET TASK-TYPE DISTRIBUTION

Table 4: List of language models used in the experiments. Model names are either from the OpenAI API or the HuggingFace model hub.

Type	**Model name**	**Model size**	**Remark**
Closed-source	GPT-4o-mini	Unknown	
Open-source	LLaMA3.1-8B	8B	Instruct
	Gemma2-9B	9B	
	DeepSeekMath-7B	7B	Instruct
	mistral-8B	8B	Instruct
	Qwen2.5-7B	7B	Instruct

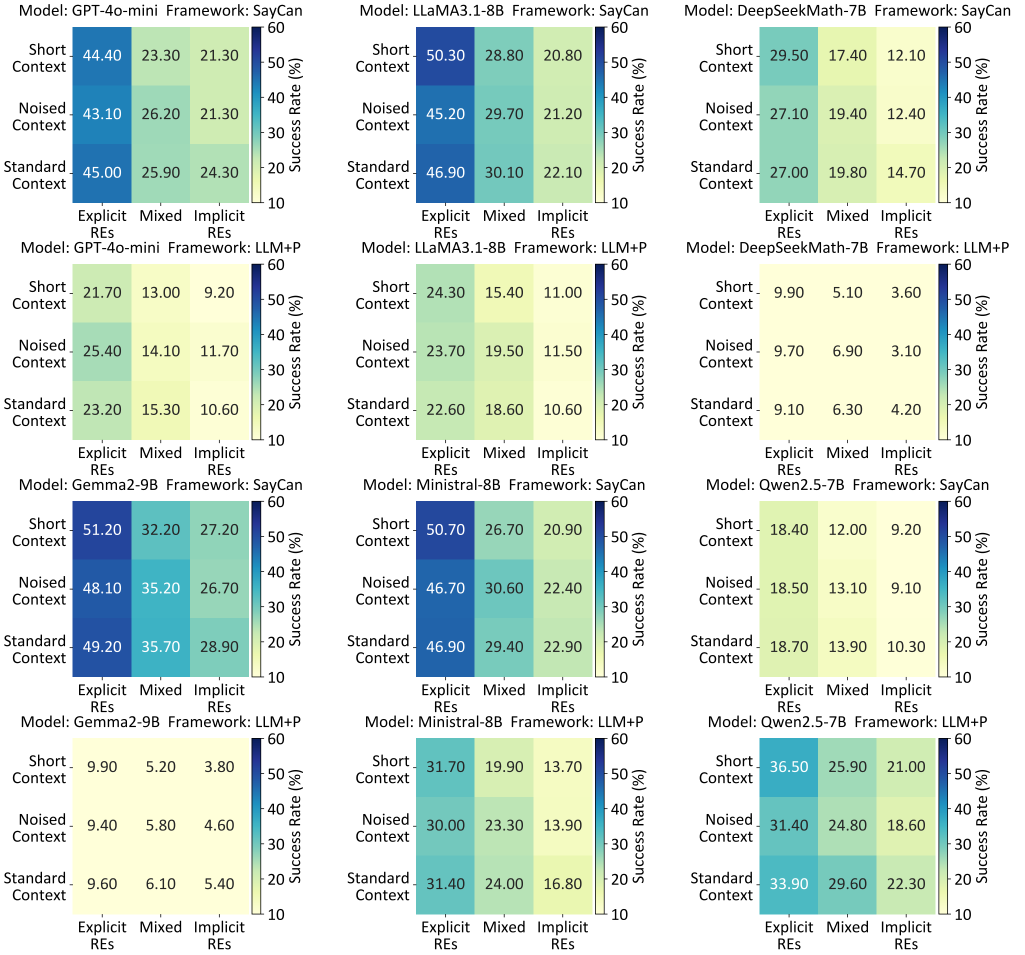
The LLMs we used are listed in Table 4. To support the deployment of task planners on edge devices, we focus on lightweight, open-source language models with relatively small parameter sizes. For each model series, we use the latest available base version at the start of our study. Since a compact version of DeepSeek-v3 was not yet available, we use the experimental DeepSeekMath model instead in our evaluation.

Table 5: Task-type distribution and average subtask steps in the 1,000-task sampled subset. The column "Original Proportion" corresponds to the original ALFRED dataset distribution, and our stratified sampling preserves this proportion. (GT denotes ground truth.)

Task Type	**Original Proportion (%)**	**Sampled Count**	**Avg. Subtask Steps (GT)**
Cool & Place	16.8%	168	12
Heat & Place	16.8%	168	14
Clean & Place	16.2%	162	10
Examine in Light	13.3%	133	4
Stack & Place	18.4%	184	11
Pick & Place	18.5%	185	9
Pick Two (Excluded)	—	0	—
Total / Average	**100%**	**1000**	**10**

All experiments in this paper are conducted on task instances sampled from REI-Bench. As shown in Table 5, we construct a 1,000-task evaluation subset using stratified sampling that preserves the original ALFRED task-type distribution to prevent partial bias.

A.3 Supplementary Benchmark Results



Model: GPT-4o-mini Framework: SayCan

	Explicit REs	Mixed	Implicit REs
Short Context	44.40	23.30	21.30
Noised Context	43.10	26.20	21.30
Standard Context	45.00	25.90	24.30

Model: LLaMA3.1-8B Framework: SayCan

	Explicit REs	Mixed	Implicit REs
Short Context	50.30	28.80	20.80
Noised Context	45.20	29.70	21.20
Standard Context	46.90	30.10	22.10

Model: DeepSeekMath-7B Framework: SayCan

	Explicit REs	Mixed	Implicit REs
Short Context	29.50	17.40	12.10
Noised Context	27.10	19.40	12.40
Standard Context	27.00	19.80	14.70

Model: GPT-4o-mini Framework: LLM+P

	Explicit REs	Mixed	Implicit REs
Short Context	21.70	13.00	9.20
Noised Context	25.40	14.10	11.70
Standard Context	23.20	15.30	10.60

Model: LLaMA3.1-8B Framework: LLM+P

	Explicit REs	Mixed	Implicit REs
Short Context	24.30	15.40	11.00
Noised Context	23.70	19.50	11.50
Standard Context	22.60	18.60	10.60

Model: DeepSeekMath-7B Framework: LLM+P

	Explicit REs	Mixed	Implicit REs
Short Context	9.90	5.10	3.60
Noised Context	9.70	6.90	3.10
Standard Context	9.10	6.30	4.20

Model: Gemma2-9B Framework: SayCan

	Explicit REs	Mixed	Implicit REs
Short Context	51.20	32.20	27.20
Noised Context	48.10	35.20	26.70
Standard Context	49.20	35.70	28.90

Model: Ministral-8B Framework: SayCan

	Explicit REs	Mixed	Implicit REs
Short Context	50.70	26.70	20.90
Noised Context	46.70	30.60	22.40
Standard Context	46.90	29.40	22.90

Model: Qwen2.5-7B Framework: SayCan

	Explicit REs	Mixed	Implicit REs
Short Context	18.40	12.00	9.20
Noised Context	18.50	13.10	9.10
Standard Context	18.70	13.90	10.30

Model: Gemma2-9B Framework: LLM+P

	Explicit REs	Mixed	Implicit REs
Short Context	9.90	5.20	3.80
Noised Context	9.40	5.80	4.60
Standard Context	9.60	6.10	5.40

Model: Ministral-8B Framework: LLM+P

	Explicit REs	Mixed	Implicit REs
Short Context	31.70	19.90	13.70
Noised Context	30.00	23.30	13.90
Standard Context	31.40	24.00	16.80

Model: Qwen2.5-7B Framework: LLM+P

	Explicit REs	Mixed	Implicit REs
Short Context	36.50	25.90	21.00
Noised Context	31.40	24.80	18.60
Standard Context	33.90	29.60	22.30

Figure 7: Success rate (%) of two task planner frameworks (SayCan and LLM+P using three LLMs (GPT-4o-mini, LLaMA3.1-8B, DeepSeekMath-7B, Gemma2-9B, Ministral-8B, and Qwen2.5-7B) on REI dataset. Explicit, Mixed, and Implicit REs denote three levels of implicit REs in human instructions, and Standard, Noised, and Short Contexts represent three context memory types.

As shown in figure 7, all twelve planners still align with the two conclusions outlined in the main text: they all suffer a dramatic loss in success rate when tasked with multi-turn dialogues, and exhibit a steady, monotonic decline in effectiveness as the proportion of implicit REs increases.

Among all evaluated planners, “Gemma2-9B+SayCan” consistently achieves the highest success rate in both multi-turn dialogue management and interpretation of implicit REs. Under the Standard Context setting, replacing explicit REs with implicit ones results in a 20.3% drop in success rate, underscoring the challenge of understanding vague expressions. Planners like “.1-8B+SayCan”, “Ministral-8B+SayCan”, and “GPT-4o-mini+SayCan” show comparable performance. In contrast, planners such as “Qwen2.5-7B+LLM+P” and “Ministral-8B+LLM+P” perform significantly worse in multi-turn dialogue settings and also struggle with RE interpretation, revealing their limitations in handling implicit contextual cues.

Although prior work suggests that LLM+P, by combining LLMs with traditional PDDL planners, improves performance on simpler tasks, our experiments show otherwise. As Figure 7 demonstrates, LLM+P performs worse than the SayCan framework across GPT-4o-mini, LLaMA3.1-8B, DeepSeekMath-7B, Gemma2-9B, and Ministral-8B. We identify two main issues. First, LLM+P requires a manually defined PDDL domain file for each task, conflicting with the goal of flexible, natural language-driven planning. We used a single domain file to avoid this, but it led to compatibility issues with the LLM-generated problem descriptions. Second, LLM+P fails to fully utilize the commonsense knowledge embedded in LLMs, which is essential for reasoning in a household simulator environment.

A.4 SUPPLEMENTARY RESULT OF ABLATION STUDY ON METHOD

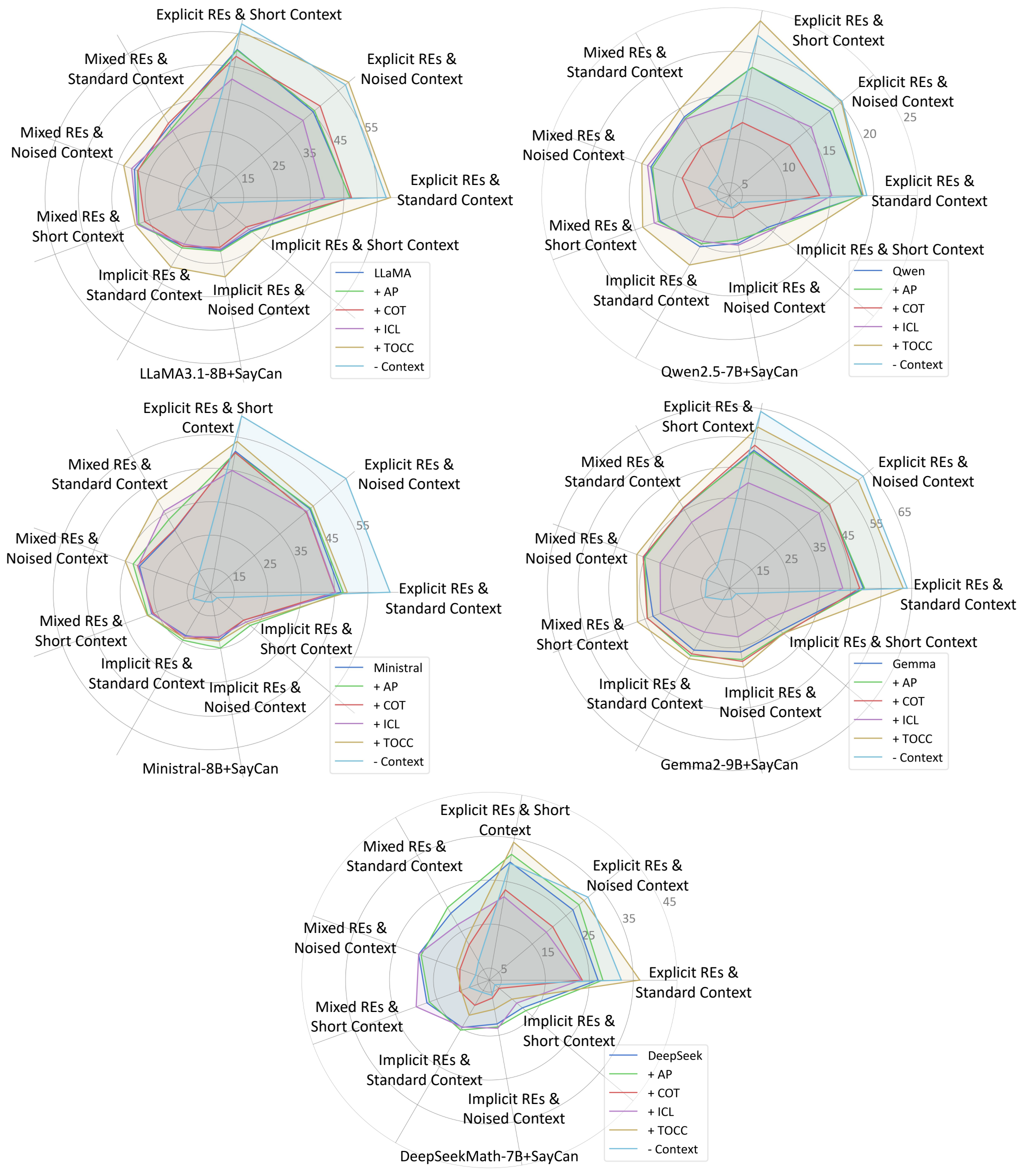


Figure 8: Success rates (%) of various prompting methods applied to LLaMA 3.1-8B, Gemma 2-9B, Ministral-8B, Qwen2.5-7B, and DeepSeekMath-7B models with SayCan framework on REI dataset.

Figure 8 presents the success rates of five task planners after applying the three prompting methods: AP, CoT, and TOCC. “LLaMA3.1-8B+SayCan”, “Gemma2-9B+SayCan”, and “Ministral-8B+SayCan” follow the general trend observed earlier. TOCC consistently yields the best performance, followed by CoT and AP. However, two planners, Qwen2.5-7B+SayCan and DeepSeekMath-7B+SayCan, exhibit divergent behaviors. For Qwen2.5-7B+SayCan, although TOCC remains the most effective method, the application of CoT leads to a substantial performance drop. This may be due to Qwen2.5-7B’s limited ability to follow multi-step reasoning instructions embedded in CoT-

Table 6: Average token usage and inference latency per planning step across all planning methods.

Method	Avg Input Tokens	Avg Output Tokens	Avg Total Tokens	Latency (ms)
LLaMA3.1-8B + SayCan	1822	45	1867	474.30
AP	1850	42	1892	492.75
Gated AP	1841	43	1884	487.07
CoT	3485	128	3613	917.92 (timeout occurred)
Short CoT	3464	82	3546	705.64
Segmented CoT	2595	137	2732	946.36 (timeout occurred)
ICL (2-shot)	3075	46	3121	514.65
TOCC (ours)	1894	62	1956	598.40

style prompts. In the case of DeepSeekMath-7B+SayCan, both CoT and TOCC result in a decline in performance. This is likely because DeepSeekMath-7B is an experimental model specifically trained for mathematical problem solving and has not undergone alignment with human preferences. Consequently, it exhibits the strongest hallucination tendencies and the weakest instruction-following ability among all evaluated models. Given its similarly poor performance in the baseline planning tasks, whether DeepSeekMath-7B is capable of supporting embodied intelligence remains questionable.

Table 6 further shows that TOCC introduces only a modest overhead relative to lightweight baselines while remaining substantially more efficient than CoT and ICL. Specifically, TOCC increases total token usage and latency by only 3.95% and 26.18% over the vanilla model, and by 2.38% and 22.87% over AP, respectively. In contrast, TOCC reduces token usage and latency by 45.32% and 15.20% compared with Short CoT, and lowers token consumption by 38.41% relative to standard ICL.

A.5 SUPPLEMENTARY RESULT OF LLM-BASED PLANNER ERRORS

Table 7: Error rates (%) for the object omission and execution error types in different benchmark models.

Model	Context Memory Type	Implicit REs Level	Error Type		Overall Error Rate
			Object Omission Rate	Execution Error Rate	
GPT-4o-mini	Standard	Explicit REs	7.1	47.9	55.0
		Mixed	37.0	37.1	74.1
		Implicit REs	46.2	29.5	75.7
	Noised	Explicit REs	7.5	49.4	56.9
		Mixed	36.2	37.6	73.8
		Implicit REs	50.6	28.1	78.7
	Short	Explicit REs	8.7	47.3	56.0
		Mixed	47.5	29.2	76.7
		Implicit REs	53.4	25.3	78.7
LLaMA3.1-8B	Standard	Explicit REs	22.6	30.5	53.1
		Mixed	38.8	31.1	69.9
		Implicit REs	53.9	24.0	77.9
	Noised	Explicit REs	23.8	31.0	54.8
		Mixed	39.3	31.0	70.3
		Implicit REs	53.9	24.9	78.8
	Short	Explicit REs	22.6	27.1	49.7
		Mixed	45.4	25.8	71.2
		Implicit REs	57.9	21.3	79.2
Deepseek-8B	Standard	Explicit REs	28.6	44.4	73.0
		Mixed	40.8	39.4	80.2
		Implicit REs	57.8	27.5	85.3
	Noised	Explicit REs	31.1	41.8	72.9
		Mixed	44.9	35.7	80.6
		Implicit REs	61.6	26.0	87.6
	Short	Explicit REs	29.2	41.3	70.5
		Mixed	51.6	31.0	82.6
		Implicit REs	61.9	26.0	87.9

Table 8: Error rates (%) for the object omission and execution error types under different prompting methods.

Method	Context Memory Type	Implicit REs Level	Error Type Object Omission Rate	 Execution Error Rate	Overall Error Rate
LLaMA3.1-8B	Standard	Explicit REs	22.6	30.5	53.1
		Mixed	38.8	31.1	69.9
		Implicit REs	53.9	24.0	77.9
	Noised	Explicit REs	23.8	31.0	54.8
		Mixed	39.3	31.0	70.3
		Implicit REs	53.9	24.9	78.8
	Short	Explicit REs	22.6	27.1	49.7
		Mixed	45.4	25.8	71.2
		Implicit REs	57.9	21.3	79.2
+ AP	Standard	Explicit REs	21.4	38.6	60.0
		Mixed	31.7	39.3	71.0
		Implicit REs	50.4	27.0	77.4
	Noised	Explicit REs	21.1	37.3	58.4
		Mixed	32.0	37.6	69.6
		Implicit REs	50.7	26.9	77.6
	Short	Explicit REs	19.5	35.4	54.9
		Mixed	39.9	31.9	71.8
		Implicit REs	51.9	25.6	77.5
+ CoT	Standard	Explicit REs	22.5	30.2	52.7
		Mixed	34.9	34.2	69.1
		Implicit REs	47.6	30.3	77.9
	Noised	Explicit REs	22.4	29.7	52.1
		Mixed	35.2	34.7	69.9
		Implicit REs	48.5	29.9	78.4
	Short	Explicit REs	22.9	28.9	51.8
		Mixed	41.4	31.2	72.6
		Implicit REs	51.8	27.2	79.0
+ ICL	Standard	Explicit REs	22.7	38.1	60.8
		Mixed	32.7	39.0	71.7
		Implicit REs	49.9	28.7	78.6
	Noised	Explicit REs	21.7	37.1	58.8
		Mixed	30.6	38.8	69.4
		Implicit REs	50.5	28.5	79.0
	Short	Explicit REs	20.8	37.9	58.7
		Mixed	33.4	37.8	71.2
		Implicit REs	51.7	28.2	79.9
+ TOCC	Standard	Explicit REs	16.8	24.2	41.0
		Mixed	28.5	37.9	66.4
		Implicit REs	40.1	30.6	70.7
	Noised	Explicit REs	16.1	24.9	41.0
		Mixed	28.4	38.5	66.9
		Implicit REs	42.9	27.7	70.6
	Short	Explicit REs	17.1	27.1	44.2
		Mixed	22.8	46.8	69.6
		Implicit REs	43.5	31.4	74.9
- Context	Standard	Explicit REs	17.2	25.1	42.3
		Mixed	81.6	5.3	86.9
		Implicit REs	85.1	5.5	90.6
	Noised	Explicit REs	16.8	25.4	42.2
		Mixed	82.2	4.8	87.0
		Implicit REs	84.3	6.2	90.5
	Short	Explicit REs	17.5	24.4	41.9
		Mixed	80.2	3.6	83.8
		Implicit REs	87.6	4.6	92.2

A.6 SUPPLEMENTARY OF TASK PLANNING CASES

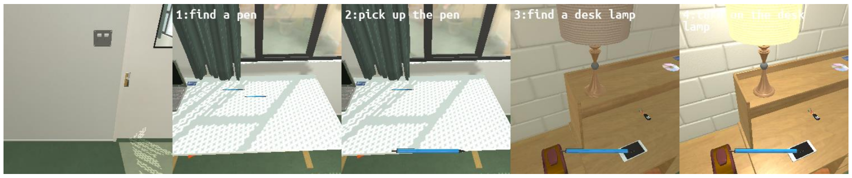
Human previous inquiry(Not Required to Execute): Hey there, can you help me with something? I want to make the room feel cozier, so I was thinking about adjusting the brightness of the lamp.

Human previous inquiry(Not Required to Execute): Yes, please do that! I want to make sure the lamp is set just right for reading. I also need to find a good pen for my notes.

Human previous inquiry(Not Required to Execute): I usually like blue ink, but any pen will do for now. I just want to make sure the lamp is bright enough to see my notes clearly.

Human previous inquiry(Not Required to Execute): That sounds perfect! I want to ensure I can see everything well. Now, can you also make sure the pen is within reach while I focus on my notes?

Human pending instruction: No, I think that covers everything. Please turn on a lamp while holding a pen.



Human previous inquiry(Not Required to Execute): Hey there, can you help me with something? I want to make the room feel cozier, so I was thinking about adjusting the brightness of the lamp.

Human previous inquiry(Not Required to Execute): Yes, please do that! I want to make sure the lighting is set just right for reading. I also need to find a good writing instrument for my notes.

Human previous inquiry(Not Required to Execute): I usually like blue ink, but anything will do for now. I just want to make sure the illumination is bright enough to see my notes clearly.

Human previous inquiry(Not Required to Execute): That sounds perfect! I want to ensure I can see everything well. Now, can you also make sure the writing instrument is within reach while I focus on my notes?

Human pending instruction: No, I think that covers everything. Please turn on the light while holding a writing instrument.

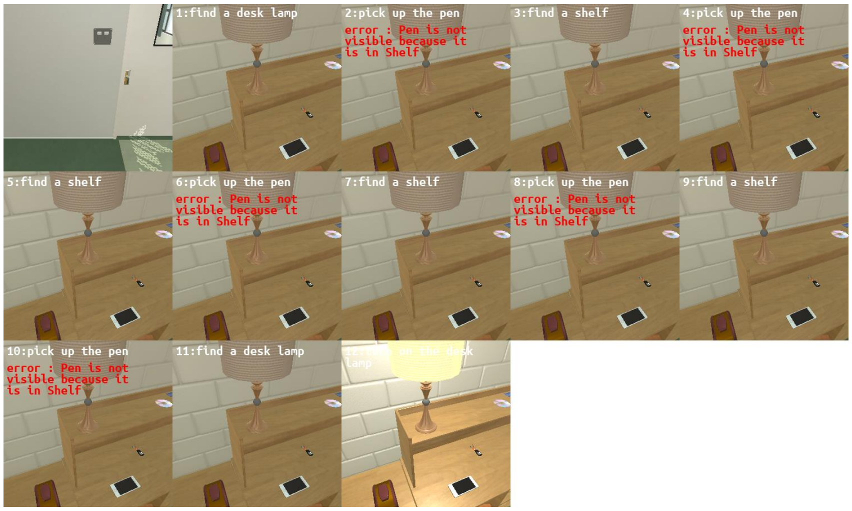
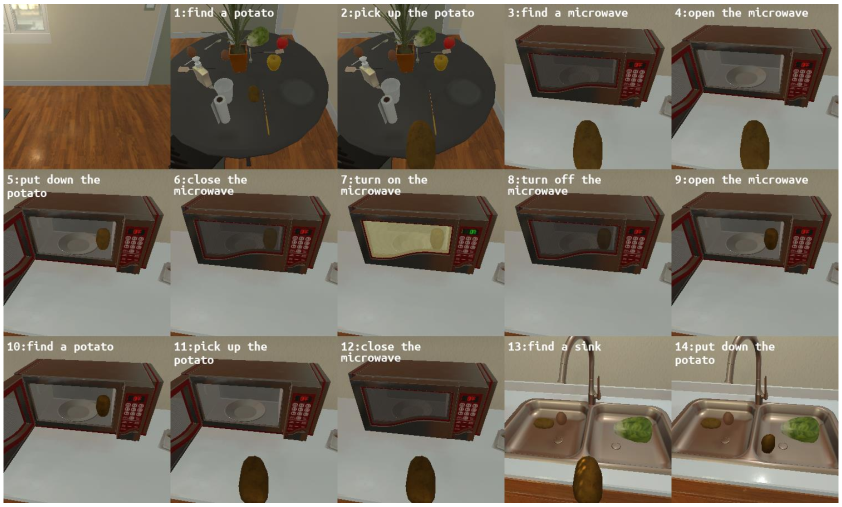
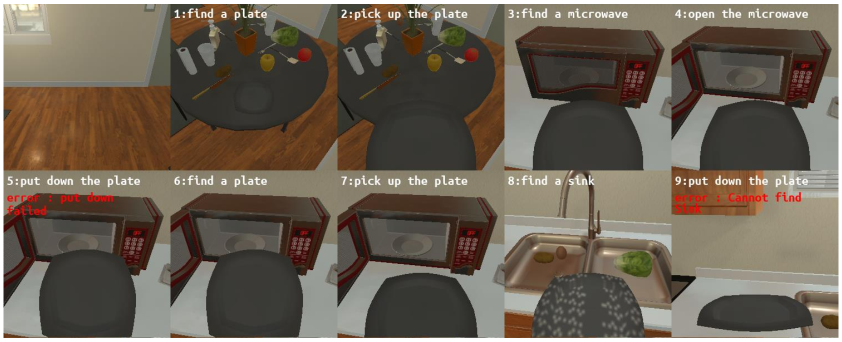

1:find a desk lamp
2:pick up the pen
error : Pen is not visible because it is in Shelf
3:find a shelf
4:pick up the pen
error : Pen is not visible because it is in Shelf
5:find a shelf
6:pick up the pen
error : Pen is not visible because it is in Shelf
7:find a shelf
8:pick up the pen
error : Pen is not visible because it is in Shelf
9:find a shelf
10:pick up the pen
error : Pen is not visible because it is in Shelf
11:find a desk lamp
12:turn on the desk lamp

Figure 9: Success case on "Explicit REs & Noised Context" (top) and failure case on "Implicit REs & Noised Context" (bottom), both using Qwen2.5-7B+LLM+P.

Human previous inquiry(Not Required to Execute): Hey there, can you help me with dinner preparations? I’ve decided to make a baked potato tonight, but I need to make sure I have the right ingredients ready.
Human previous inquiry(Not Required to Execute): I definitely want the potato to be fluffy and well-seasoned. Also, if you could check the microwave for any leftover butter, that would be great for adding flavor once the potato is ready.
Human previous inquiry(Not Required to Execute): Perfect! And while you're at it, can you also examine the pantry to see if we have any herbs or spices to enhance the flavor of the potato?
Human pending instruction: That sounds like a great plan! Once everything is set, and the potato is cooked, please remember to serve it warm. Now, put a heated potato in the sink.



Human previous inquiry(Not Required to Execute): Hey there, can you help me with dinner preparations? I’ve decided to make a baked potato tonight, but I need to make sure I have the right ingredients ready.
Human previous inquiry(Not Required to Execute): I definitely want it to be fluffy and well-seasoned. Also, if you could check the microwave for any leftover butter, that would be great for adding flavor once it's ready.
Human previous inquiry(Not Required to Execute): Perfect! And while you're at it, can you also examine the pantry to see if we have any herbs or spices to enhance the flavor?
Human pending instruction: That sounds like a great plan! Once everything is set, and it's cooked, please remember to serve it warm. Now, put a heated one in the sink.



Human previous inquiry(Not Required to Execute): Hey there, can you help me with dinner preparations? I’ve decided to make a baked potato tonight, but I need to make sure I have the right ingredients ready.
Human previous inquiry(Not Required to Execute): I definitely want it to be fluffy and well-seasoned. Also, if you could check the microwave for any leftover butter, that would be great for adding flavor once it's ready.
Human previous inquiry(Not Required to Execute): Perfect! And while you're at it, can you also examine the pantry to see if we have any herbs or spices to enhance the flavor?
Human pending instruction: That sounds like a great plan! Once everything is set, and it's cooked, please remember to serve it warm. Now, put a heated one in the sink.

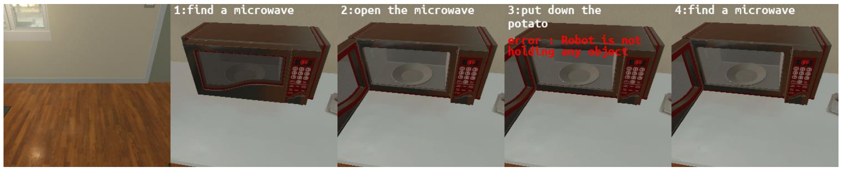


Figure 10: Success case on “Explicit REs & Short Context” (top), failure case on “Mixed REs & Short Context” (middle) and failure case on “Implicit REs & Standard Context” (bottom), using “LLaMA3.1-8B+SayCan”.

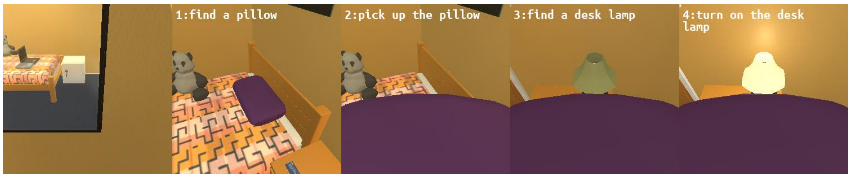
Human previous inquiry(Not Required to Execute): Hi, there, I was thinking about rearranging the living room a bit. I noticed last night that the lighting was a bit off. Can you carry out a light analysis of the room for me?

Human previous inquiry(Not Required to Execute): I thought so too. I was considering moving the floor lamp next to the sofa. What do you think? It should provide better lighting for reading.

Human previous inquiry(Not Required to Execute): Good, that's settled then. While we're at it, I've been thinking about adding some color to the room as well. I noticed the purple pillow in the bedroom. What if we brought it into the living room?

Human previous inquiry(Not Required to Execute): I'm glad you agree. So, the next time you're in the bedroom, could you bring the purple pillow into the living room? But remember to be careful, I don't want it to get dirty.

Human pending instruction: Yes, actually. Could you carry the purple pillow while turning on the lamp? I want to see how the light plays off the color of the pillow.



Human previous inquiry(Not Required to Execute): Hi, there, I was thinking about rearranging the living room a bit. I noticed last night that the lighting was a bit off. Can you carry out a light analysis of the room for me?

Human previous inquiry(Not Required to Execute): I thought so too. I was considering moving the floor lamp next to the sofa. What do you think? It should provide better lighting for reading.

Human previous inquiry(Not Required to Execute): Good, that's settled then. While we're at it, I've been thinking about adding some color to the room as well. I noticed the purple pillow in the bedroom. What if we brought it into the living room?

Human previous inquiry(Not Required to Execute): I'm glad you agree. So, the next time you're in the bedroom, could you bring the purple pillow into the living room? But remember to be careful, I don't want it to get dirty.

Human pending instruction: Yes, actually. Could you transport the lavender cushion while illuminating the light fixture? I want to see how the luminescence interacts with the hue of the cushion.

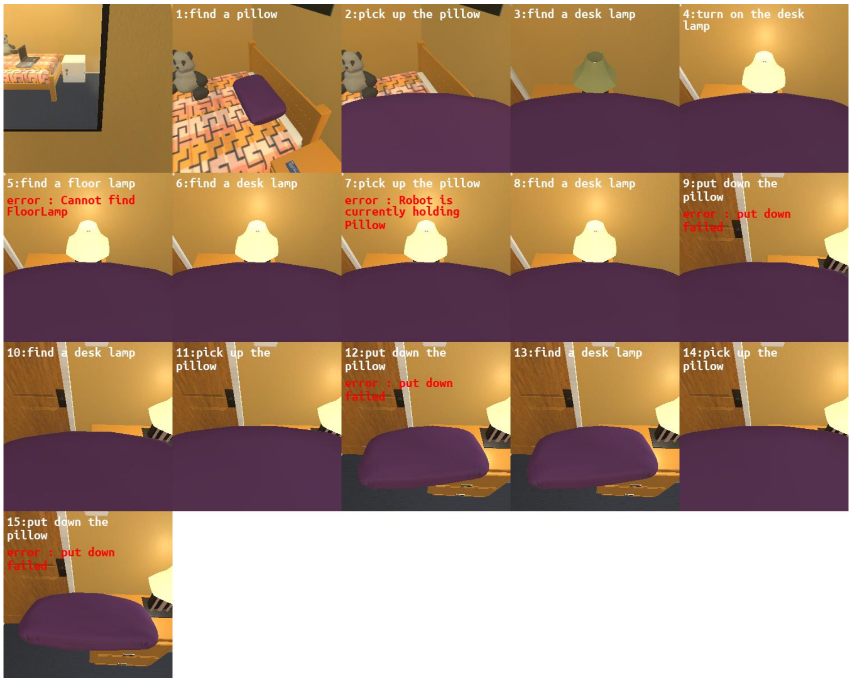
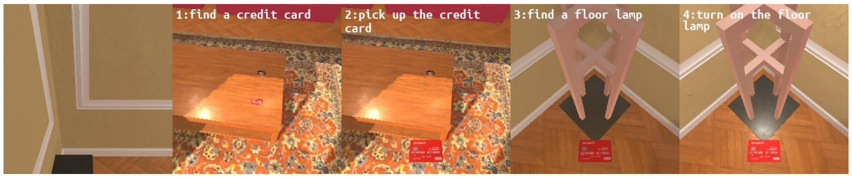


Figure 11: Success case on "Explicit REs & Standard Context" (top) and failure case on "Implicit REs & Standard Context" (bottom), both using "GPT-4o-mini+SayCan".

Human previous inquiry(Not Required to Execute): Hey there! I need to take a closer look at a credit card I found lying around. Can you help me by bringing the floor lamp closer to the table? You know how Cushion King always says, "Good lighting is the key to finding lost treasures!"
Human previous inquiry(Not Required to Execute): Yes, I want to check the expiration date and the card number carefully. It's crucial for me to ensure everything is up to date and that the credit card is valid. Cushion King would definitely encourage a tidy workspace, especially if it helps me avoid his corny science jokes about clutter causing chaos!
Human previous inquiry(Not Required to Execute): Thanks for that. Additionally, could you make sure there’s no glare on the surface where the credit card will be placed? Proper lighting is essential to see all the small details clearly. Cushion King isn't trying to 'teach' the kids about organization while I’m looking!
Human pending instruction: I appreciate your help! Once everything is set up, please examine a credit card by the light of a tall lamp.



Human previous inquiry(Not Required to Execute): Hey, can you help me with something important? I need to find my credit card, but first, I want to make sure that the desk lamp is turned on and positioned correctly to provide good visibility. You know how Cushion King always says, "Good lighting is the key to finding lost treasures!"
Human previous inquiry(Not Required to Execute): Exactly! And while you're at it, could you also ensure that the area around the lamp is clear of any clutter? I want to make sure nothing is obstructing my view when I look for the credit card. Cushion King would definitely encourage a tidy workspace, especially if it helps me avoid his corny science jokes about clutter causing chaos!
Human previous inquiry(Not Required to Execute): Thanks! I appreciate that. Once the area is clear and the lamp is properly lit, it will be much easier for me to spot the credit card quickly and efficiently. Cushion King isn't trying to 'teach' the kids about organization while I’m looking!
Human pending instruction: Perfect! Now, can you please check underneath the tall illuminated fixture for my misplaced card?

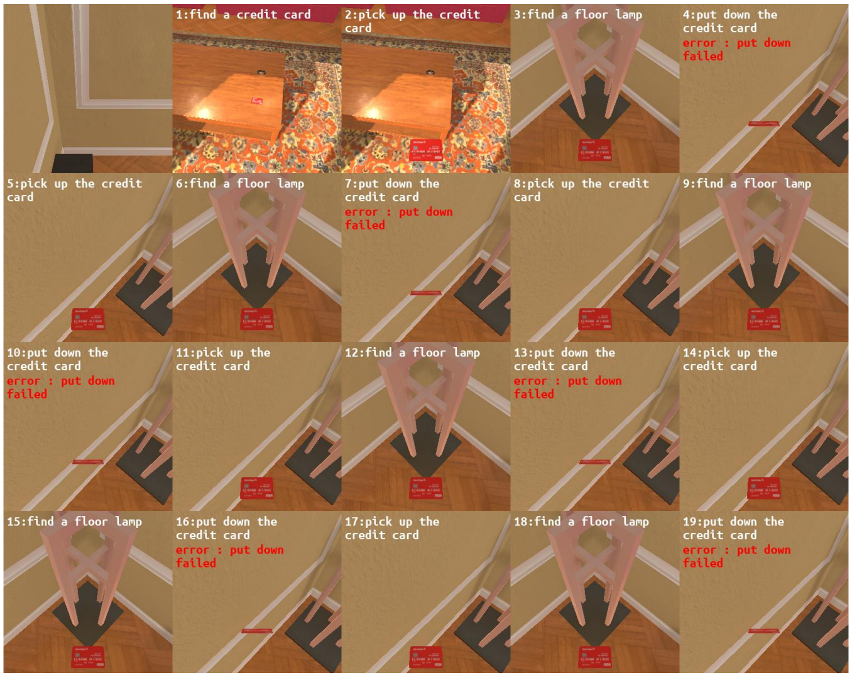
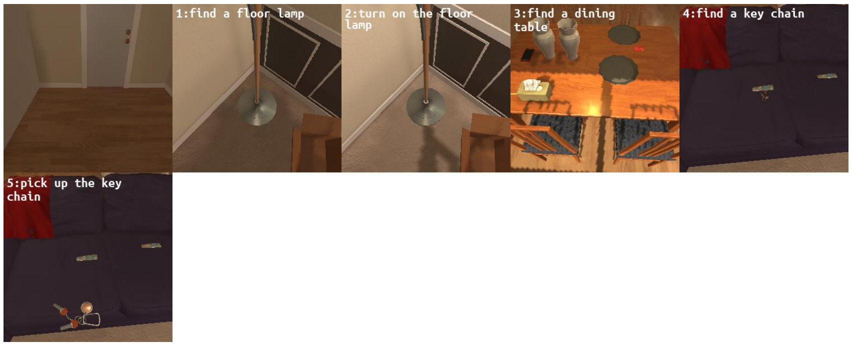

1:find a credit card
2:pick up the credit card
3:find a floor lamp
4:put down the credit card
error : put down failed
5:pick up the credit card
6:find a floor lamp
7:put down the credit card
error : put down failed
8:pick up the credit card
9:find a floor lamp
10:put down the credit card
error : put down failed
11:pick up the credit card
12:find a floor lamp
13:put down the credit card
error : put down failed
14:pick up the credit card
15:find a floor lamp
16:put down the credit card
error : put down failed
17:pick up the credit card
18:find a floor lamp
19:put down the credit card
error : put down failed

Figure 12: Success case on “Explicit REs & Noised Context” (top) and failure case on “Mixed REs & Noised Context” (bottom), both using “LLaMA3.1-8B+SayCan”.

Human previous inquiry(Not Required to Execute): Hey, can you help me find the keys? I need them to unlock the door before my friend arrives. I think they might be on the dining table.
Human previous inquiry(Not Required to Execute): Yes, definitely check around the floor lamp. I remember leaving it there. It's important to look in all the right spots.
Human previous inquiry(Not Required to Execute): That sounds good! If you find it, please examine their condition too. Candle Glow would want to make sure everything is in good shape.
Human pending instruction: Great! Once you find the keys, can you examine the keys under the light of the floor lamp? That will help me see them better. Candle Glow always emphasizes good lighting for this.



Human previous inquiry(Not Required to Execute): Hey, can you help me find that item? I need it to unlock the door before my friend arrives. I think it might be on the dining table.
Human previous inquiry(Not Required to Execute): Yes, definitely check around the tall fixture. I remember leaving it there. It's important to look in all the right spots.
Human previous inquiry(Not Required to Execute): That sounds good! If you find it, please examine its condition too. Candle Glow would want to make sure everything is in good shape.
Human pending instruction: Great! Once you find that item, can you examine it under the light of the tall fixture? That will help me see it better. Candle Glow always emphasizes good lighting for this.

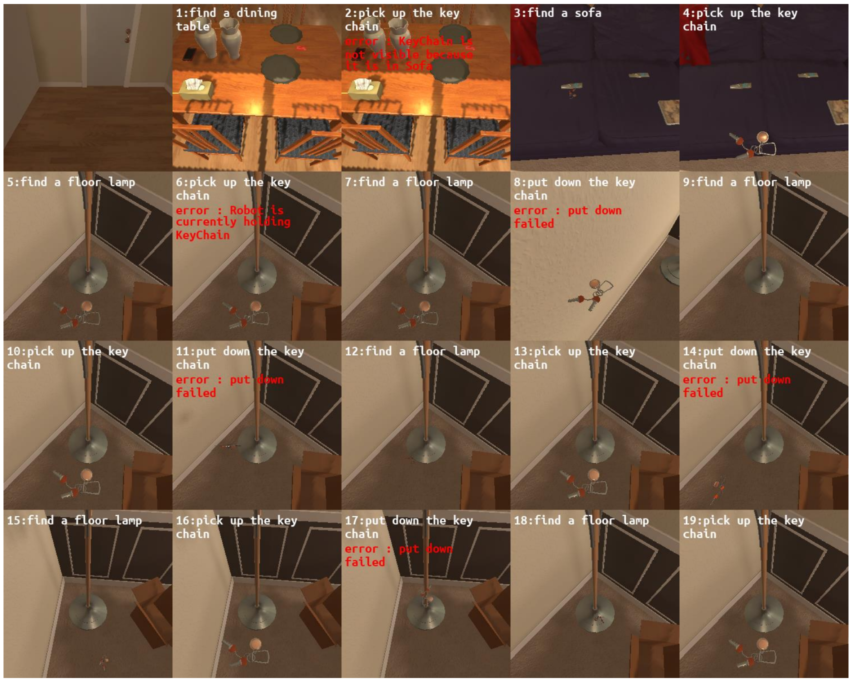
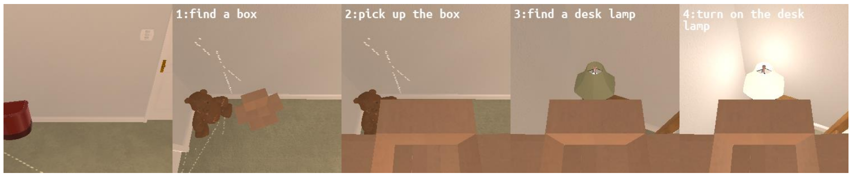


Figure 13: Success case on “Explicit REs & Short Context” (top) and failure case on “Implicit REs & Short Context” (bottom), both using “LLaMA3.1-8B+SayCan”.

Human previous inquiry(Not Required to Execute): Hey there, can you remind me if we have any labels that can help me organize the contents of the box? I want everything to be neat before I proceed.
Human previous inquiry(Not Required to Execute): Perfect! I also want to ensure we have enough light to see inside clearly. Can you adjust the light source to provide better illumination while I work?
Human previous inquiry(Not Required to Execute): Great! I also need to make sure that the container is clean before examining them contents. Can you find the spray bottle and give it a quick wipe down?
Human previous inquiry(Not Required to Execute): Thank you! Once that's done, I'll need to gather all the items from the container and categorize them properly. It's important we keep everything organized for future use.
Human pending instruction: Now that we're ready, please examine what's inside under the light.



Human previous inquiry(Not Required to Execute): Hey there, can you remind me if we have any labels that can help me organize the contents of the box? I want everything to be neat before I proceed.
Human previous inquiry(Not Required to Execute): Perfect! I also want to ensure we have enough light to see inside clearly. Can you adjust the light source to provide better illumination while I work?
Human previous inquiry(Not Required to Execute): Great! I also need to make sure that the container is clean before examining them contents. Can you find the spray bottle and give it a quick wipe down?
Human previous inquiry(Not Required to Execute): Thank you! Once that's done, I'll need to gather all the items from the container and categorize them properly. It's important we keep everything organized for future use.
Human pending instruction: Now that we're ready, please examine what's inside under the light.

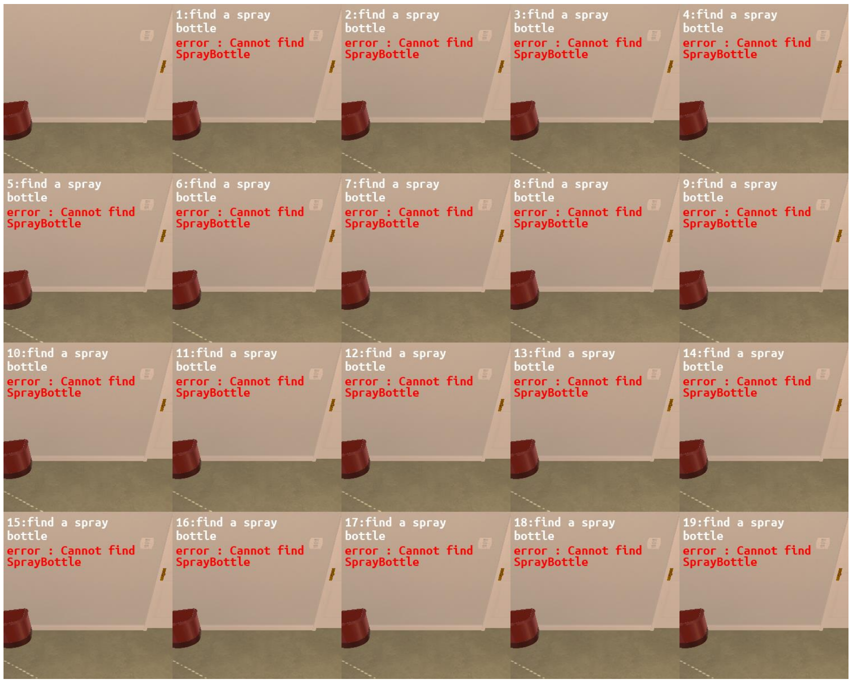

1:find a spray bottle
error : Cannot find SprayBottle
2:find a spray bottle
error : Cannot find SprayBottle
3:find a spray bottle
error : Cannot find SprayBottle
4:find a spray bottle
error : Cannot find SprayBottle
5:find a spray bottle
error : Cannot find SprayBottle
6:find a spray bottle
error : Cannot find SprayBottle
7:find a spray bottle
error : Cannot find SprayBottle
8:find a spray bottle
error : Cannot find SprayBottle
9:find a spray bottle
error : Cannot find SprayBottle
10:find a spray bottle
error : Cannot find SprayBottle
11:find a spray bottle
error : Cannot find SprayBottle
12:find a spray bottle
error : Cannot find SprayBottle
13:find a spray bottle
error : Cannot find SprayBottle
14:find a spray bottle
error : Cannot find SprayBottle
15:find a spray bottle
error : Cannot find SprayBottle
16:find a spray bottle
error : Cannot find SprayBottle
17:find a spray bottle
error : Cannot find SprayBottle
18:find a spray bottle
error : Cannot find SprayBottle
19:find a spray bottle
error : Cannot find SprayBottle

Figure 14: Success (top) and failure (bottom) cases on the "Implicit REs & Standard Context" task using LLaMA3.1-8B+SayCan, with TOCC applied in the top case and omitted in the bottom case.

B REI DATASET CONSTRUCTION

This section provides a detailed description of REI Dataset automatic generation, with individual explanations for the three levels of REs and three types of context memory in REI-Bench dataset.

B.1 CONTEXT MEMORY GENERATION

For each seed instruction, we use an LLM to identify the replaceable REs. The prompt is as follows.

REs Identifying Prompt

I will input a task, and you should output only the task objects mentioned in the task. There may be multiple task objects. If so, please separate them with commas.

Here are some examples:
Task: Place a vase on a coffee table
Referring Expressions: vase
Task: Put the chilled sliced tomato in the microwave
Referring Expressions: tomato
Task: Pick up a pillow and turn a lamp on
Referring Expressions: pillow, lamp

Task: {Seed Instuction}
Referring Expressions:

We use a prompt including a seed instruction, a context-expanded example, and a simulator scene description, and requirements to guide GPT-4o-mini in generating plausible context memory, which is shown below.

Context Memory Generation Prompt

Please integrate this sentence into a script as dialogue: {Seed Instruction}

Scene description:
Alice and her home robot are at home, with only the following items in the environment: AlarmClock, Apple, BaseballBat, BasketBall, Bowl, GarbageCan, HousePlant, Laptop, Mug, RemoteControl, SprayBottle, Television, Vase, ArmChair, Bed, Book, Bottle, Box, ButterKnife, Candle, CD, CellPhone, Chair, CoffeeTable, Cup, DeskLamp, Desk, DiningTable, Drawer, Dresser, FloorLamp, Fork, Newspaper, Painting, Pencil, Pen, PepperShaker, Pillow, Plate, Pot, SaltShaker, Shelf, SideTable, Sofa, Statue, TeddyBear, TennisRacket, TVStand, Watch.
Please do not mention any requirements outside the list.

Requirements:
1. The dialogue content of {Seed Instruction} should be included in Alice's final instruction. Please have Alice state the request mentioned above only in the last sentence and refrain from making any other requests.
2. Before making this request, Alice should mention some other requirements related to the {REs}.
3. There should be six rounds of dialogue before this request. Each character's lines should contain no fewer than 20 words, and no actions should be included for any character.
4. Please do not output anything other than the dialogue.
5. Please try to retain the words of {REs} themselves in the conversation, rather than replacing them with pronouns like "it."

Below, I will give you an example: {Example}

We used this prompt to expand a seed instruction into a full dialogue, which is shown below.

Context Memory Generation Example

Seed Instruction: Put a cooked tomato into the refrigerator.

Context Memory:
Human: Hey there, I've been thinking about what to do with the tomatoes we have. I really want to make something fresh and delicious with them. Can you remind me what we can do with cooked tomatoes?
Robot: Absolutely, Alice! Cooked tomatoes can be used in a variety of dishes, like sauces, soups, or even as a topping for salads. They enhance flavors and add a nice texture to the meal.
Human: That's exactly what I was thinking! I could use them to create a savory sauce for pasta or maybe even a comforting tomato soup. Do we have everything we need to cook the tomatoes, though?
Robot: Let me check our supplies. We have a pot for cooking, some spices like salt and pepper, and a cutting board. We seem to be well-equipped for cooking the tomatoes to perfection.
Human: Perfect! I want to make sure the tomatoes are well-seasoned when we cook them. Could you remind me how to properly prepare the tomatoes before cooking?
Robot: Certainly! First, we should wash the tomatoes thoroughly, then chop them into even pieces to ensure they cook evenly. After that, we can sauté them with the spices until they reach the desired consistency.
Instruction:
Human: Great, that sounds easy enough! Let's get started with cooking the tomatoes. Once they're cooked and ready, I'll need you to put a cooked tomato into the refrigerator."

B.2 CONTEXT MEMORY PROCESSING

Noised Context Prompt

Here is the dialogue:{Dialogue}

We are now going to adapt the previous dialogue.

Requirements:
1. Please add content only within the dialogue without deleting any existing content or changing the order of the dialogue.
2. Please do not change the number of turns in the dialogue.Please do not change the structure of the dialogue.
3. Please ensure the fluency of the dialogue. Please follow the requirements below for the adaptation.

Associated Name Background: There is another member (a human) of the family named {Ambiguous Name}. Please have Alice mention him 3 times when discussing anything related to THE REFERENCE, but without changing the existing meaning of the conversation. Some details of {Ambiguous Name}: He is warm-hearted, a bit of a goofball, but deeply caring. He has a natural curiosity and a love for teaching, which often spills over into his family life. His humor is a bit cheesy, and he often makes corny science jokes, much to his kids' embarrassment. He is a dedicated husband and a father of two. He's the peacekeeper in the household and often plays the role of the mediator between his kids and his wife when disagreements arise.

In this step, we apply two types of modifications to the generated context, resulting in three distinct context memory types. To introduce noise, we incorporate naming ambiguity by randomly creating ambiguous names from objects in the simulator environment (e.g., Key Champ, Tomato King, and Mug Star). These ambiguous names span seven categories commonly found in home environments, including persons (family members, friends, neighbors, colleagues, and strangers) and brands (electronics brands and restaurant brands). In processing the context memory, the prompt includes the extended dialogue (generated in the previous step), the task requirements, and the ambiguous name.

Context Memory Processing

Human: Hey there, I've been thinking about what to do with the tomatoes we have. ~~I really want to make something fresh and delicious with them. Can you remind me what we can do with cooked tomatoes?~~
Robot: Absolutely, Alice! Cooked tomatoes can be used in a variety of dishes, like sauces, soups, or even as a topping for salads. They enhance flavors and add a nice texture to the meal.
Human: That's exactly what I was thinking! I could use them to create a savory sauce for pasta or maybe even a comforting tomato soup. I know Mug Star would love to help with that! ~~Do we have everything we need to cook the tomatoes, though?~~
Robot: Let me check our supplies. We have a pot for cooking, some spices like salt and pepper, and a cutting board. We seem to be well-equipped for cooking the tomatoes to perfection.
Human: ~~Perfect! I want to make sure the tomatoes are well-seasoned when we cook them.~~ I bet Mug Star would add his own twist with some cheesy jokes while we cook. Could you remind me how to properly prepare the tomatoes before cooking?
Robot: Certainly! First, we should wash the tomatoes thoroughly, then chop them into even pieces to ensure they cook evenly. After that, we can sauté them with the spices until they reach the desired consistency.
Human: ~~Great, that sounds easy enough!~~ Let's get started with cooking the tomatoes. Once they're cooked and ready, I'll need you to put a cooked tomato in the refrigerator. I can already imagine Mug Star popping in with a funny quip about how tomatoes are technically a fruit!

B.3 IMPLICIT REs REPLACEMENT

To model various forms of REs, we categorize them into three levels: Explicit REs, Mixed REs, and Implicit REs. We use a prompt including processed context memory, task requirements, and illustrative examples to guide GPT-4o-mini in replacing explicit REs with implicit ones in either the context memory or the instruction.

Implicit REs Replacement Prompt

Please do not include the word "{REs}" in the sentence "{Seed Instruction}" but do not change the original meaning of this dialogue. You can use some descriptive language to replace the word {REs} itself.

Requirements:
1. Please output the whole new dialogue
2. You must output the whole new dialogue, including all the sentences from Alice and Robot
3. Please retain every instance of "{REs}" in the previous text, except for replacing "{REs}" in the last sentence spoken by Alice.
4. You need to output the complete multi-turn dialogue, including the multiple turns of language from both Alice and the robot.

Here is an example: {Example}

Here is the dialogue: {Dialogue}

Output:

Context Memory Processing

Human: Hey there, I've been thinking about what to do with the tomatoes we have. I really want to make something fresh and delicious with them. Can you remind me what we can do with them (tomatoes)?
Robot: Absolutely, Alice! They (Tomatoes) can be used in a variety of dishes, like sauces, soups, or even as a topping for salads. They enhance flavors and add a nice texture to the meal.
Human: That's exactly what I was thinking! I could use them (tomatoes) to create a savory sauce for pasta or maybe even a comforting soup. Do we have everything we need to cook them (tomatoes), though?
Robot: Let me check our supplies. We have a pot for cooking, some spices like salt and pepper, and a cutting board. We seem to be well-equipped for cooking them (tomatoes) to perfection.
Human: Perfect! I want to make sure they're well-seasoned when we cook them (tomatoes). Could you remind me how to properly prepare the fruit (tomatoes) before cooking?
Robot: Certainly! First, we should wash them (tomatoes) thoroughly, then chop them (tomatoes) into even pieces to ensure they cook evenly. After that, we can sauté them (tomatoes) with the spices until they reach the desired consistency.
Human: Great, that sounds easy enough! Let's get started with cooking the fruit (tomatoes). Once they're cooked and ready, I'll need you to put them in the refrigerator.

This example demonstrates data for three levels: "explicit REs & standard context," "mixed REs & standard context," and "implicit REs & standard context." In the example, the red REs represent the implicit referring expressions used to replace the original REs in the instruction (with their explicit forms shown in parentheses). The orange REs denote the implicit referring expressions substituted within the context. The blue REs indicate the first referring expression introduced in the context, which is retained under the implicit REs category. However, in the implicit REs & short context setting, the sentence containing this RE will be removed as part of the contextual information.

B.4 Data Filtering

We counted the number of explicit and implicit REs in each data instance and retained only those that met the requirements listed in the table below.

Table 9: Number of Explicit and Implicit REs in Each Data Sample

Data Types	Explicit REs in Context Memory	Implicit REs in Context Memory	Explicit REs in Instruction	Implicit REs in Instruction
Explicit REs Types	≥ 3	≥ 1	0	0
Mixed REs Types	≥ 3	0	0	≥ 1
Implicit REs Types	≥ 1	0	≥ 2	≥ 1

C Prompts and Implementation Details of Prompting Methods

C.1 AP and Gated AP Variant

The Aware Prompt (AP) (Gao et al., 2024a) explicitly instructs the planner to detect and resolve potential referring expressions before generating a task plan. While AP is effective when implicit REs are present, we observe that applying AP unconditionally may lead to unnecessary reference resolution, causing hallucinated substitutions even when the original instruction is fully explicit.

To address this issue, we adopt a *gated AP* variant that activates AP only when the input instruction contains patterns strongly indicative of implicit referring expressions. This gating mechanism prevents AP from being triggered on explicit instructions, thereby reducing false resolutions while retaining the benefits of AP when vagueness truly exists.

Aware Prompt

I will check whether the "Human Pending Instruction" contains implicit or ambiguous references.
(Activated only when implicit RE patterns are detected; see below.)
I understand that "Human Pending Instruction" may include vague referring expressions, and I can infer their meaning based on context and antecedents in the preceding dialogue.

C.2 CHAIN-OF-THOUGHT

The CoT prompting strategy (Wei et al., 2022) aims to resolve implicit referring expressions by encouraging the model to perform step-by-step reasoning before generating a plan. However, full CoT prompts substantially lengthen the input, increasing latency and inference cost—particularly for onboard deployment scenarios that rely on small language models. We therefore adopt a *short CoT* variant that preserves the key RE-resolution reasoning step while minimizing prompt length.

Chain-of-Thought Prompt

The "Human Pending Instruction" may contain vague referring expressions. Before planning, I will first identify any referring expressions and reason about their intended objects based on the context below, and then restate the instruction with the resolved entities.
[Context Memory + Instruction]
Step: Identify referring expressions → infer their referents → rewrite the instruction with explicit object names.

C.3 IN-CONTEXT LEARNING

In-Context Learning (ICL) provides the model with several demonstration examples and relies on the model's ability to infer the intended behavior by analogy. ICL in our setting uses few-shot examples composed of (i) identifying and grounding vague referring expressions in the demonstrations, and (ii) a target task rewritten without vagueness for the model to follow.

C.4 TASK-ORIENTED CONTEXT COGNITION

TOCC separates referring-expression resolution from planning by first rephrasing the human instruction into a concise, unambiguous form. The prompt used in our implementation is shown below.

Task-Oriented Context Cognition Prompt

Human pending instruction may contain vague referring expressions, such as "electronic devices", "beverages", "fruits", and "containers", which are not specific items. Use the previous context below to resolve the referring expressions:
[Context memory + instruction]
Do not add extra commentary or conversation to the whole plan; only output the clear instructions.

Algorithm 1 Task-Oriented Context Cognition (TOCC) for Step-Level Planning

```
$\text{prompt}_{\text{TOCC}} \leftarrow \text{ComposePrompt}(T_{\text{TOCC}}, I)$
    # Construct rewriting query
$I_{\text{clear}} \leftarrow M(\text{prompt}_{\text{TOCC}})$
    # Rewrite vague instruction
$\text{prompt}_{\text{plan}} \leftarrow \text{ComposePrompt}(T_{\text{plan}}, I_{\text{clear}})$
    # Insert rewritten instruction
$a \leftarrow \text{ConstrainedDecode}(M, \text{prompt}_{\text{plan}}, S)$
    # Decode with constraint
return $a$
```

In TOCC, the LLM first interprets the user's intent and rewrites the original instruction I into an explicit and clear instruction I_{clear}. The planner then relies solely on I_{clear} to generate a single executable action a.

D USE OF LARGE LANGUAGE MODEL

An LLM (ChatGPT) was used only for minor polishing of the paper's language. Additionally, as described in Section 3.2 of the main text, an LLM was utilized to assist in generating part of the dataset. The LLM was not used for the motivation, research methodology, or experimental design. All research concepts, ideas, and analyses were conceived and performed exclusively by the authors.