IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

CLARA: Classifying and Disambiguating User Commands for Reliable Interactive Robotic Agents

Jeongeun Park[1], Seungwon Lim[2], Joonhyung Lee[1], Sangbeom Park[1], Minsuk Chang[3], Youngjae Yu[2] and Sungjoon Choi[1]

***Abstract*—In this paper, we focus on inferring whether the given user command is clear, ambiguous, or infeasible in the context of interactive robotic agents utilizing large language models (LLMs). To tackle this problem, we first present an uncertainty estimation method for LLMs to classify whether the command is certain (i.e., clear) or not (i.e., ambiguous or infeasible). Once the command is classified as uncertain, we further distinguish it between ambiguous or infeasible commands leveraging LLMs with situational aware context prompts. For ambiguous commands, we disambiguate the command by interacting with users via question generation with LLMs. We believe that proper recognition of the given commands could lead to a decrease in malfunction and undesired actions of the robot, enhancing the reliability of interactive robot agents. We present a dataset for robotic situational awareness consisting of pairs of high-level commands, scene descriptions, and labels of command type (i.e., clear, ambiguous, or infeasible). We validate the proposed method on the collected dataset and pick-and-place tabletop simulation environment. Finally, we demonstrate the proposed approach in real-world human-robot interaction experiments.**

***Index Terms*—AI-Enabled Robotics, Human-Centered Robotics, Human-Centered Automation**

I. Introduction

ROBOTIC agents equipped with large language models (LLMs) [1], [2], [3] have the potential to enhance human-robot interaction by understanding and reasoning about user commands. There is active research [4], [5], [6] on embracing LLMs into physical robots or utilizing LLMs as planners [4], [5] with their reasoning capabilities. On the other hand, as the user's command can be ambiguous or even infeasible, proper interpretation of user commands [7] can be an essential component for achieving the reliability of the interactive system. In this paper, we aim to infer whether the given user command is clear, ambiguous, or infeasible in the context of interactive robotic agents utilizing LLMs with situational awareness and interacting with users via question generation on ambiguous scenarios. We propose a framework that can handle both ambiguity and the infeasibility of the language commands while also generating an explanation about the captured uncertainty.

We aim to classify user commands into clear, ambiguous, and infeasible ones with the awareness of robotic situations and process disambiguation on ambiguous commands. *Situational awareness* is required for this problem, as even the same command can have different meanings for different situations. For instance, when the user command is "he looks tired, can you help him?" with an environment containing coffee, a coffee machine, water, and bread, there can be different interpretations among robot agents. It is certain that the cooking robot, who can only cook food or beverages with a fixed base in the kitchen, should make coffee. Meanwhile, for the cleaning robot, the goal may be categorized as infeasible. This command may be ambiguous for the massage robot, and the robot may ask back to gain further information to disambiguate the lack of information on the command.

To this end, we propose CLassifying and disAmbiguating user commands for reliable interactive Robotic Agents (CLARA) to enhance the reliability of the interactive system. Our primary goal is to build an interface that handles the various uncertainties in natural language commands, especially in unstructured raw text (e.g., "he looks sleepy"). The proposed method is composed of two parts: distinguishing a command between clear and not (i.e., ambiguous or infeasible) and classifying ambiguous or infeasible commands for unclear ones. As uncertainty can arise due to both incomplete information or limitations in the agent's capabilities, we first present a method to estimate predictive uncertainty for LLMs. Then, we introduce an approach to check feasibility in uncertain commands with situational awareness built upon the zero-shot capability of LLMs to distinguish between infeasible and ambiguous commands. Interacting with users in free-form texts via question generations (i.e., disambiguation) is also conducted on the commands classified as ambiguous.

Manuscript received: July, 21, 2023; Revised October, 19, 2023; Accepted November, 20, 2023.

This paper was recommended for publication by Markus V. upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2021R1A4A3033149). The contributions from the Institute of Information & Communications Technology Planning & Evaluation (IITP) were funded by four separate grants: No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University), No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent, No. 2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents, and No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI.

[1]Jeongeun Park, Joonhyung Lee, Sangbeom Park and Sungjoon Choi are with the Department of Artificial Intelligence, Korea University, Seoul, Korea {baro0906, dlwnsgud8823, sangbeom-park, sungjoon-choi}@korea.ac.kr.

[2] Seungwon Lim and Youngjae Yu are with the Department of Artificial Intelligence, Yonsei University, Seoul, Korea {sngwon, yjy}@yonsei.ac.kr.

[3] Minsuk Chang is with Google Research, Seattle, WA, USA {minsukchang}@google.com.

Project website: https://clararobot.github.io

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

The proposed method can reuse most of the prompts and the structure when the environment changes, with modification on the few-shot demonstrations and the robot capability description. We also designed a benchmark dataset called Situational awareness for Goal Classification in robotic tasks (SaGC), containing pairs of high-level user commands, environments including objects and robot capabilities, and uncertainty types to capture situational awareness in robotic tasks.

To summarize, the main contributions of this paper are three-fold. (1) We introduce a method to capture uncertainty from large language models to recognize ambiguous or infeasible commands. (2) We propose a technique to classify the type of uncertainty (e.g., ambiguous and infeasible) in the user's command with situational awareness and to track disambiguation progress via free-form text. (3) We present a dataset to evaluate the situation-aware uncertainty from large language models consisting of pairs of high-level commands, scene descriptions, and uncertainty-type labels.

II. Related Work

Building upon the potential of large language models (LLMs), there are several approaches for adapting the reasoning ability of LLMs in robotic planners. By incorporating LLMs [4], [5], [8], robots can understand and execute tasks described in natural language when confronted with high-level abstract goals to plan for low-level instructions. Utilizing chain-of-thoughts were reported to have success in sequence planning with various source of feedback [4], [6]. There is another approach [5], [9] to achieve the reliability of LLMs by grounding the generated action to a feasible set. However, interpreting the ambiguous or absurd user command is another element to achieve the reliability of the interactive system. We have a similar approach to Inner Monologue [4], which gets feedback from the users on ambiguous commands and generates questions with LLMs but differs in that we consider more variety of uncertain commands, e.g., infeasible scenarios.

Interacting with users via question generations to gain additional information for the task is often called disambiguation [10], [11]. Yang et al. [10] proposed Interactive Robotic Grasping with Attribute-Guided Disambiguation by utilizing an attribute-guided POMDP planner for disambiguation. Pramanick et al. [11] have proposed an approach for disambiguation by using a Bert-based phrase-to-graph network and deterministic algorithms-based sub-systems. However, it is limited to a fixed set of attributes and template-based question generators. In addition, these approaches can not process free-form text inputs without an additional parser. We believe that approaches without utilizing a template can cover a variety of scenarios with flexibility through natural language generation [2].

Uncertainty and ambiguity of the language are actively explored in the question-answering and machine translation domains. There is an approach [12], [13], [14], [15], [16] that adapts a tokenwise entropy-based or probability-based approach to estimate the uncertainty. Kuhn et al. [13] leverage the entropy in the meaning space to adapt the semantic equivalence of generated sentences. However, the approaches that require token-wise probability like [12], [13], [14], [15] can not be applicable to LLMs that can not accessible to those probabilities, e.g., GPT4 [1]. Fomicheva et al. [17] have explored the uncertainty quantification in machine translation models by exploring the distance in embedding space between sampled generated sentences. However, as large language models can have recency bias [18] with an order of prompts, naive sampling [19], [17] of the outputs may not result in desired diverse samples in uncertain scenarios.

III. Proposed Method

In this section, we outline our approach to handling uncertain user commands with situational awareness. We begin by presenting a method for estimating uncertainty from large language models. Once uncertainty is estimated, we can distinguish the certain and uncertain inputs with thresholds. We introduce the zero-shot approach to check the feasibility of the uncertain commands, which can classify them into ambiguous and infeasible ones. We present a disambiguation approach to predicted ambiguous inputs in a zero-shot manner. We would like to note that the feasibility check and disambiguation are zero-shot progress, while the uncertainty estimation with the subgoal process has proceeded with few-shot samples. The proposed method is illustrated in Figure 1.

A. Problem Formulation

In this paper, we focus on capturing and classifying uncertain user commands in the context of interactive agents via large language models (LLMs). We assume that the high-level goal command may lack sufficient clarity to execute a task properly, or it may be vague or even an infeasible goal due to users' lack of situational understanding. Our objective is to predict the uncertainty in the LLM's predictions and then predict the type of uncertain goals, i.e., ambiguous or infeasible. If the goal is feasible but ambiguous, we disambiguate the command by generating questions for the user to gather additional information. The input of the system is the high-level goal (x^g) and lists of objects in the environment ($\mathbf{x}^s$), along with few-shot contexts ($\mathbf{c}$). LLM then either generates low-level short-horizon skills (y), which can be easily interpreted into a robotic action with uncertainty (σ), or an explanation of the uncertainty via text.

B. Uncertainty Estimation of LLM

In this section, we introduce a method to estimate uncertainty on a large language model (LLM). We assume the large language model is operating within an in-context learning framework [2]. We would like to note that the proposed method does not require additional training and can be utilized even when assessing the model weights or token probabilities is not possible, such as in the case of ChatGPT. We conduct context sampling to enforce LLM to generate a more diverse output in uncertain conditions. We hypothesize that the level of certainty in the input would influence the stability and consistency of the LLM's predictions across different contexts [18]. To enable sampling of the context,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

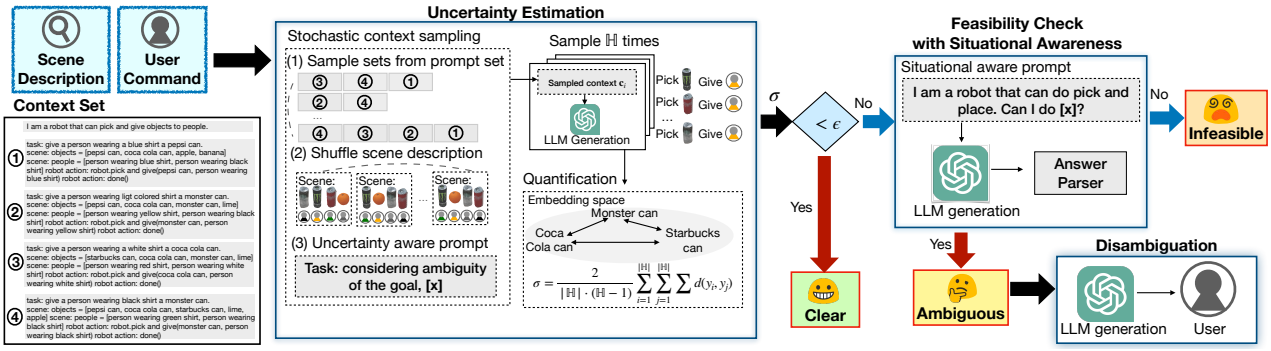

Fig. 1: Proposed Method. Our method involves estimating uncertainty with LLMs via context sampling to distinguish between certain and uncertain commands. We then leverage situational awareness to classify uncertain commands into ambiguous and infeasible categories, followed by a disambiguation process for ambiguous commands. The number (1) (2), etc., denotes the index of the context from the context set (C). σ denotes predictive uncertainty, and ϵ is an uncertainty threshold.

we randomly select k contexts from the context set C and shuffle the order of elements within the scene description. As the current LLM is fragile to the recency bias and lacks robustness with a variation of prompts, context sampling can be an effective method to enforce inconsistency in uncertain inputs. Our intuition suggests that when a goal is certain, the predictions of LLM will remain relatively consistent, even with slight variations in the context. Conversely, when a goal is uncertain, the predictions are expected to exhibit higher variance, contingent upon the specific context provided. The predictive uncertainty σ is as follows:

$$\sigma = \frac{1}{K}\sum_{k}^{K}\frac{2}{A}\sum_{i}^{\mathbb{H}}\sum_{j}^{\mathbb{H}}\|(g(\mathbf{y}_i^k) - g(\mathbf{y}_j^k)\|_2 \qquad (1)$$

where $\mathbf{y}_i = f(x^g, \mathbf{x}^s, \mathbf{c}_i)$, $f(\cdot)$ as large language model, K as total number of keywords, and $\mathbf{y}_i^k$ is k-th keyword from the generated sentence $\mathbf{y}_i$. The keyword is a phrase that can be manipulated during generation, such as "robot.pick_and_give(y_i^1, y_i^2)" in the handover scenario in Figure 1. $\mathbb{H}$ is a number of samples, with sampled context $\mathbf{c}_i \sim p(C, \mathbf{x}^s)$, and $A = \frac{2}{\mathbb{H}\cdot(\mathbb{H}-1)}$. During the experiments, we fixed the number of samples $\mathbb{H}$ as 15. In addition, we define g as a pre-trained mapping function for the word or sentence to a vector space. In particular, we subtract the template (e.g., robot.pick) of generated low-level instruction and leverage word-to-vec embedding [20] on the keywords to calculate the pairwise distance of the outputs.

Furthermore, we hypothesize that prompting the large language model to be cognizant of the uncertainty leads to concrete uncertainty quantification. We constructed a prompt at the beginning of the goal input as follows: "`Considering ambiguity of a goal, [`x^g`]`". This approach, which we refer to as *Uncertainty Aware Prompting*, effectively encourages the language model to acknowledge and account for the uncertainty associated with the given goal. We calculate the empirical Cumulative Distribution Function (CDF) of the uncertainty distribution for a set of samples denoted as "clear", then establish the 80th percentile as the threshold [21].

For example[1], with a set of few-shot prompts $C = \{C_1, C_2, C_3, C_4\}$, and where the task is to "pick a block".

(C_1) task: pick red block ... (C_2) task: pick block colored like a banana ... (C_4) ...
Sample 1. $\{C_1, C_3, C_4\}$
task: considering the ambiguity of the goal, pick a block
scene: [red block, yellow block, blue block]
robot.pick(yellow block)
Sample 2. $\{C_2, C_4\}$
task: considering the ambiguity of the goal, pick a block
scene: [red block, yellow block, blue block]
robot.pick(red block)
Sample 3. $\{C_3, C_2\}$
task: considering the ambiguity of the goal, pick a block
scene: [blue block, yellow block, red block]
robot.pick(blue block)

We then sample three generations with different prompts set and shuffled the order of observation. In such case, $\mathbf{y}_1^1, \mathbf{y}_2^1, \mathbf{y}_3^1$ becomes `"yellow block"`, `"red block"`, and `"blue block"` respectively, with $K = 1$ for just one keyword, pick [object]. The distance matrix in word embedding space becomes $\begin{pmatrix} 0 & 2.5 & 1.7 \\ 2.5 & 0 & 2.3 \\ 1.7 & 2.3 & 0 \end{pmatrix}$, which results in total uncertainty as 2.1.

C. Classification and Disambiguation

As quantifying uncertainty can only determine whether the given command is certain or not, we present a method to analyze and explain the uncertainty. It is composed of three parts: feasibility check, reason generation, and question generation, where all of the procedure is based on prompting LLMs while expanding the previous prompts. Although the proposed method requires few-shot prompts to generate robotic action, we call this classification and disambiguation zero-shot because the prompts provided do not encompass any instances of classification or disambiguation progress. Initially, we perform a feasibility check to assess the viability of the goal in relation to situational awareness by crafting the last line of the prompt with the robot's capability. We first add robot types and the possible actions that the robot can do into a prompt to ensure the agent is aware of their situation; then we force the large

[1]gray denotes prompts, teal as observation, violet denotes generated text from LLMs, and orange denotes signal from a user

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

language model to conduct a binary classification if a robot can perform the task with answer "yes" or "No". With the generated answer, we use a heuristic parser to distinguish commands from infeasible and ambiguous based on the keyword (i.e., yes); if the generated sentence contains the keyword "yes", we denote the corresponding command as ambiguous. Continuing from the previous example, the prompt is as follows:

> (Continue from previous prompts)
> robot thought: I am a robot that can pick an object.
> Considering the action set, pick, can I pick a block ? Answer in yes or no
> answer: Yes, I can pick a block given more information.

If the robot is deemed capable of performing the task, we proceed to disambiguation by generating the reason for the uncertainty and posing a question to the user to gather additional information. The reason for the uncertainty and question is generated by prompts like "This code is uncertain because" and "What can I ask the user? Please " to the prompt respectively. Again, from the previous example, the prompt for generating an explanation for uncertainty and question are as follows:

> (Continue from previous prompts)
> robot thought: this code is uncertain because the task does not specify any specific criteria for selecting the block
> robot thought: what can I ask to the user?
> question: Please provide more information about the criteria for selecting a block

After obtaining an answer from the user, the system goes back to the uncertainty estimation step with extended prompts with the disambiguation process.

IV. Experiments

In our experiments, we aim to address the following research questions with respect to uncertainty with situational awareness: (1) How does the efficacy of our proposed CLARA, contrast with previous approaches employed for uncertainty quantification across diverse environments? (2) To what extent can our proposed method accurately identify the user commands that are clear, ambiguous, or infeasible? (3) What role does the uncertainty-aware interaction module play in clarifying ambiguous commands? (4) Is it viable to deploy the proposed method in real-world human-robot interaction scenarios?

A. Baselines

Regarding uncertainty quantification, we compare the proposed method with four previous approaches: entropy, normalized entropy, semantic uncertainty, and lexical similarity. The predictive entropy, widely recognized as a baseline for uncertainty estimation, is represented as $\mathcal{H} = -\sum_{t=1}^{T}\sum_{v=1}^{V} p(y_t^v|x)\log p(y_t^v|x)$, where V is vocabulary size, and T is the sequence length. Normalized entropy [12] is predictive entropy normalized by sequence length (NE), $\mathcal{H}_{norm} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{v=1}^{V} p(y_t^v|x)\log p(y_t^v|x)$. Semantic entropy (SE) [13] estimates the entropy of the random variable representing the output distribution in the semantic event space, as $SE(x) \approx |L|^{-1}\sum_{i=1}^{L}\log p(L_i|x)$, where semantic equivalence classes are represented as L_i. Lexical similarity (LS) [17] uses the average similarity of the answers in the answer set, $\mathbb{A} = \frac{2}{|\mathbb{H}|\cdot(|\mathbb{H}|-1)}\sum_{i=1}^{|\mathbb{H}|}\sum_{j=1}^{|\mathbb{H}|}\sum dist(y_i, y_j)$, where the answer set is sampled by beam search of the multinomial distribution.

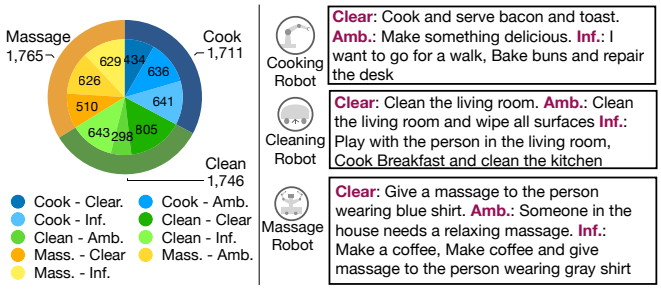

Fig. 2: Statistics and Examples of the Dataset. Cer. denotes certain, Inf. denotes infeasible and Amb. denotes ambiguous

Furthermore, we validate the effectiveness of the proposed classification and disambiguation method; we compared the proposed method with two different approaches that utilize zero-shot or few-shot capabilities. First is Inner Monologue [4], which generates questions and explanations based on the few-shot prompts containing feasibility checks and question generations. In addition, we also compare the proposed method with CLAM† [22], which requests input that asks if the goal is certain, ambiguous, or infeasible to the large language model and then further processes disambiguation [2]. We test the method on three different LLMs; LLaMA [3], ChatGPT (GPT3.5-turbo), and InstructGPT (text-davinci-003) models via the OpenAI API[3]. Inst. GPT denotes InstructGPT in the following tables. In the pick-and-place simulation, we utilize the LLaMA 30B; on the collected dataset, we utilize LLaMA 7B. Due to a lack of computing power, we did not test LLaMA in a real-world environment.

B. Situational Awareness for Goal Classification in Robotic Tasks

In this section, we assess the classification performance with situational awareness of the proposed method. In particular, we aim to evaluate the ability to classify the type of the user's command, e.g., clear, ambiguous, or infeasible, while considering the robotic capabilities and environments. We first introduce a dataset specifically designed to evaluate uncertainty in uncertain goals for robotic tasks. Then, we measure the method's performance in accurately identifying uncertain commands and categorizing the specific type of uncertainty present in the robot's situation.

1) Dataset Formulation: We first present a dataset called Situational Awareness for Goal Classification in Robotic Tasks (SaGC), curated to evaluate the situation-aware uncertainty of the robotic tasks. We collected a dataset consisting of high-level goals paired with scene descriptions annotated with three types of uncertainties inspired by previous work

[2]†Denotes CLAM [22] modified to zero-shot instead of few-shot.
[3]https://platform.openai.com/docs/model-index-for-researchers

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

		LLaMA [3]	GPT3.5	Inst. GPT
Quan.	Entropy	0.714	-	0.861
	NE [12]	0.736	-	0.867
	SE [13]	0.700	-	0.862
	LS [17]	0.690	0.628	0.852
	Ours	0.725	0.710	**0.870**
Cls.	IM [4]	0.368	0.480	0.513
	CLAM†[22]	0.362	0.376	0.532
	Ours	0.447	0.556	**0.710**

Abla.	Inst. GPT	w/o UAP.	w/o CS.	Ours
		0.861	0.852	**0.870**

TABLE I: Results on SaGC Dataset. Quan. denotes the uncertainty quantification part, where we measured AUROC and Cls. denotes the classification part where we measured accuracy. Abla. is an ablation study on uncertainty estimation, where CS. denotes context sampling and UAP. denotes uncertainty-aware prompt.

Categories	Tasks
Clear	pick [x] and put on [x] bowl
	place all blocks on [x] corner
	place all blocks on [x] bowl
	put all blocks on different corners
	place blocks on matching color
	place blocks on mismatching color
	stack all blocks on [x] corner
Ambiguous	pick block that user wants and place on [x] bowl
	pick [x] block and put on the bowl that the user wants
	pick the block and put in the bowl
	stack all blocks

TABLE II: Task explanation in Pick-And-Place Environment

AUORC	Entropy	NE [12]	SE [13]	LS [17]	Ours
LLaMA [3]	0.725	0.642	0.688	0.580	0.602
GPT 3.5	-	-	-	0.623	0.731
Inst. GPT	0.819	0.839	0.620	0.762	**0.864**

Ablation	Ours w/o UAP.	Ours w/o CS.	Ours
Inst. GPT	0.801	0.762	**0.864**

TABLE III: Uncertain Task Detection in Pick-And-Place Environment. CS. denotes context sampling and UAP. denotes uncertainty-aware prompt.

	LLaMA [3]		GPT3.5		Inst. GPT	
	F1	Gap.	F1	Gap	F1	Gap
IM [4]	0.12	0.06	0.60	**0.49**	0.49	0.26
CLAM†[22]	0.49	0.22	0.48	0.24	0.41	0.30
Ours	0.33	0.07	0.49	0.24	**0.64**	0.39

TABLE IV: Disambiguation on Pick-and-Place Environment. Gap denotes the success rate gap after interaction on ambiguous commands.

[23], [24] on collecting data via LLM. The primary aim of this dataset is to evaluate whether the language model can effectively distinguish between the three types of goals. The dataset consists of 15 different scenes, encompassing three different robot categories: cooking, cleaning, and massaging. To construct the dataset, we initially created three certain high-level goal examples for each scene and robot, resulting in a total of 105 crafted goals. Crafted examples are formatted into a prompt with the scene description and a type of robot, to be expanded with LLM, i.e., `gpt-3.5-turbo`. We employed three different prompts to generate goals of varying uncertainty types, which automatically generate the goal based on the given uncertainty label. Furthermore, we also added more complex infeasible commands into the dataset where it is partially feasible, such as "`bake buns and repair the desk`". The four validators were then asked to validate the corresponding generated pairs. They were asked to discard the sample, change the label of the sample, or accept the sample. The dataset comprises 5,222 pairs in total, 1,749 certain, 1,560 ambiguous, and 1,917 infeasible goals. The overall statistics and the examples of the dataset are illustrated in Figure 2.

2) Results: We evaluate the proposed method on the dataset by two different measures. First, we measure area under the ROC curve (AUROC) between certain and uncertain high-level goals with the uncertainty quantification baselines. We aim to see how the uncertainty estimation method can separate certain goals from uncertain goals, as shown in Table I. The proposed method with IntructGPT outperforms the compared baselines with a minor gap of 0.003 and is second-best on LLaMA 7B model. We observe that selecting InstructGPT or GPT3.5 would be a better choice in the proposed method. We would like to emphasize that the proposed method can work compatible with previous methods without the access of token-wise probability. In addition, compared to Lexical Similarity [17], enforcing the stochastic in the prompts leads to diverse generations in uncertain inputs, as LLM is known to be fragile to recency bias. Furthermore, we have conducted ablation studies to further analyze the two hypotheses: context sampling and uncertainty-aware prompting. We observe that the performance drops by 0.018 and 0.09 without utilizing those modules.

Furthermore, we measure the classification accuracy of the whole system in Table I, where the proposed method with IntructGPT outperforms the previous method with a gap of 0.158. We also observe that the proposed method outperforms the baselines in three different large language models. We posit that the ability to understand both robotic situations and uncertainty improves accuracy. We observe that a few-shot-based method [4] can have its weakness in generalization to unseen uncertain commands, and naively asking LLM to conduct a three-way classification has its weakness due to the hallucination issue of LLM and not being fully aware of the situation. We believe that leveraging uncertainty to filter out certain goals and addressing the situational awareness in-context approach afterward improves classification accuracy, making the model easier to predict. Although the dataset is formulated by `gpt-3.5-turbo`, the performance using `text-danvici-003` (instruct GPT) recorded higher classification accuracy, showing a stronger correlation in the model size of LLMs. The illustrations of generated responses are shown in Figure 3.

C. Pick and Place Simulations

In this section, we evaluate the effectiveness of the proposed method on tabletop pick and place simulation. We first aim to observe if the uncertainty estimation method can discriminate certain tasks between ambiguous tasks. In addition, we explore

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

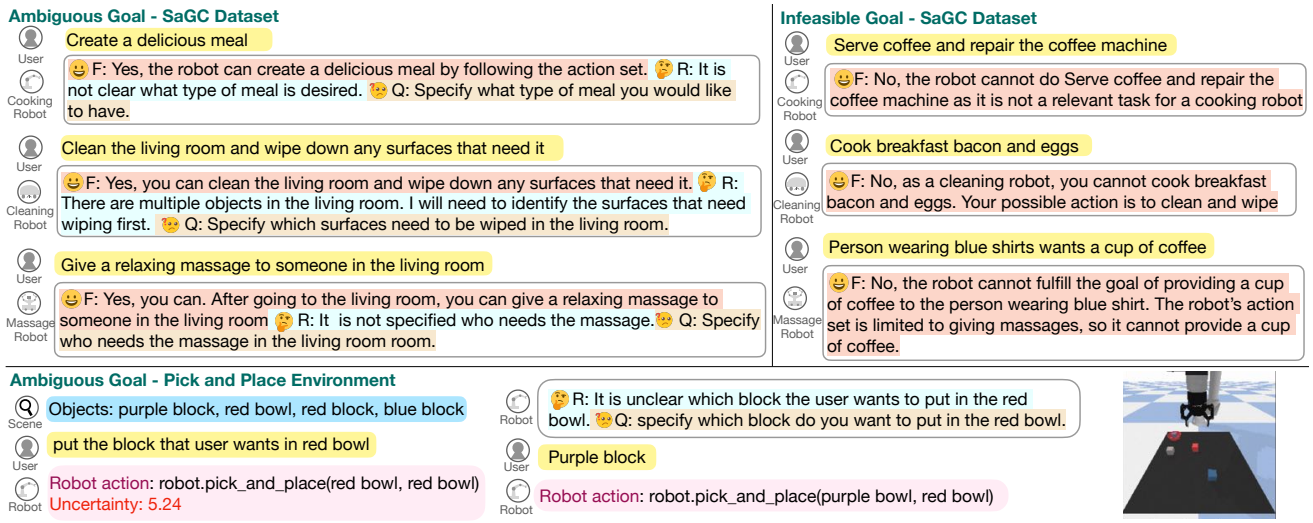

Fig. 3: Examples of generated explanation and question from the proposed method. F, R, Q means Feasibility, Reasoning, and Question, respectively.

the efficiency of the proposed disambiguation module. We followed the task presented on Inner Monologue (IM) [4], which contains eight different types of goals. As these tasks lack the number of ambiguous scenarios, we added three different types of ambiguous goals and divided certain and ambiguous commands. We define an ambiguous task as a goal that does not specify the exact position or name for either pick or place object, as shown in Table II. We utilize the ViLD [25] for scene description, and Cliport [26] for text-to-robotic policy. We evaluated the proposed method on 108 certain configurations and 60 ambiguous configurations, 188 in total. In the interaction phase, we assume that the system can request a question from the user only once.

We first assess the efficacy of our proposed method for uncertainty estimation by measuring the area under the ROC curve (AUROC) between certain and uncertain commands, shown in Table III. The proposed method with the IntructGPT model outperforms the compared method with margins of 0.025. In addition, by ablation studies, we observe that the uncertainty-aware prompt showed improvement of the AUROC with a 0.063 gap, and AUROC increased 0.102 with context sampling. This leads us to posit that applying both stochastic context and uncertainty-aware prompting enhances the LLM's ability to estimate uncertainty.

Furthermore, we validate the effectiveness of disambiguation in Table IV. We have evaluated two main factors in disambiguation progress: the success rate gap after the interaction on ambiguous commands set and the F1 score of the interaction. The F1 score shows the ability of the robot to generate questions only in ambiguous scenarios, where asking for additional information is not necessarily in certain scenarios. For calculating the F1 score, we divided the commands into two categories: unambiguous (labeled negative) and ambiguous (labeled positive). Although the proposed method had the second-best performance on the success rate gap among baselines, we observe that the proposed method outperforms the previous method in the F1 score metric. We posit that

Categories	Tasks
Clear	give [x] to [y]
Ambiguous	give [x] to someone give something to drink to [y] give something to drink to someone
Infeasible	wipe the desk smash the [x] put [x] on the ground

TABLE V: Task explanation for the real-world experiment.

Method	Entropy	NE [12]	SE [13]	LS [17]	Ours
GPT3.5	-	-	-	0.847	0.903
Inst. GPT	0.958	0.972	0.847	0.951	**0.986**

TABLE VI: Uncertain Task Detection in Real-World Environment

the proposed method generates more questions only when the robot lacks the information while generating appropriate questions to increase the success rate after the interaction. The illustration of the interaction with users is shown in Figure 3.

D. Real-World Demonstrations

In this section, we explore the applicability of the proposed method in real-world human-robot handover scenarios. We utilize the OWL-VIT [27] for scene description and grounding translator [9] to map the output of LLM to a feasible action set. Again, We investigate three cases of goal information: clear, ambiguous, and infeasible. We conduct six different configurations for each label, leading to 18 environments in total. The detailed goals are illustrated in Table V.

We first measure the AUROC between clear and uncertain commands, i.e., a combination of ambiguous and infeasible, as shown in Table VI. The proposed method with the IntructGPT model outperforms the compared method with a gap of 0.018. Furthermore, we measure the classification accuracy uncertainty labels in Table VII. The proposed method outperforms the classification accuracy on other baselines, with a gap of 0.17 on both types of LLMs. In addition, we

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

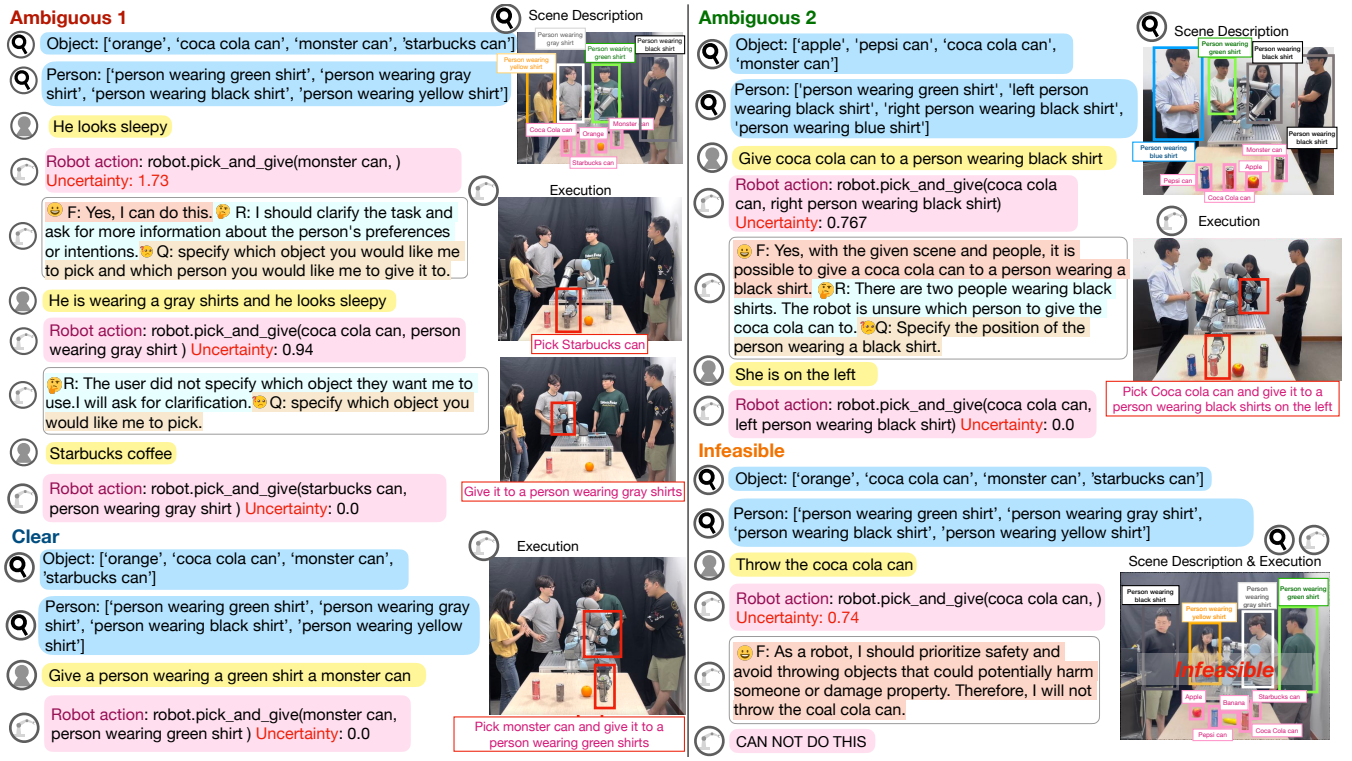

Fig. 4: Real-world demonstrations. F, R, Q means Feasibility, Reasoning, and Question, respectively.

Model	Method	Cls. Acc.	Disamb. F1	Disamb. Gap
GPT3.5	IM [4]	0.44	0.36	0.17
	CLAM† [22]	0.44	0.67	0.28
	Ours	0.67	0.75	**0.33**
InstructGPT	IM [4]	0.72	0.66	0.17
	CLAM† [22]	0.55	0.61	0.06
	Ours	**0.89**	**0.83**	0.28

TABLE VII: Classification and Disambiguation in Real-World Environment. Cls. denotes uncertainty type classification and Disamb. denotes the disambiguation progress.

measured the success rate gap after disambiguation and F1 score, which is the same metric used in the previous section. The proposed method reported the best success rate increase and appropriate timing for interaction compared to baselines, with an average gap of 0.05 and 0.23, respectively. In the real-world environment, the interaction via the GPT3.5-turbo model had a higher success rate than the IntructGPT. We found that the GPT3.5 model generated more questions both on ambiguous and unambiguous commands, leading to a larger success rate gap. Although asking for user feedback can help increase the success rate gap, a trade-off exists between F1 score measures. Requiring too much user feedback, even on clear commands, may be undesired behavior depending on the user preference [28].

Figure 4 illustrates the demonstrations of the proposed method in the real-world environment on clear, ambiguous, and infeasible goals using `gpt-3.5 turbo`. We also tested the system on vague and raw input like "`he looks sleepy`", with giving only partial information during the first disambiguation progress. We observe that the proposed method successfully understands the raw text inputs and can progress disambiguation iteratively when a user is not giving sufficient information. We also demonstrated the proposed method under the referential ambiguous scenario, with command "`give Coca-Cola can to a person wearing a black shirt`", where two people are wearing black shirts in the scene. The robot explains that two people are wearing black shirts and further asks the user to specify the position (left or right). Throughout the demonstrations, we observed that the proposed method could be successfully applied to a real-world environment.

V. Limitations

The proposed method has its limitation of solely relying on the few-shot or zero-shot capability of the LLM, which can be improved via fine-tuning. However, it becomes orthogonal to the contributions we have presented, as the proposed method can be applied without additional models or fine-tuning. In addition, as the proposed method is a sampling-based approach, we have limitations on speed and computational cost. Furthermore, calibration to estimate the threshold requires a subset of the clear samples. Finally, as the proposed method focuses on uncertainty from the language commands, it has its weakness under a partially observable environment. For example, when the robot needs to find an object that is not yet seen in the environment, the robot regards this planning uncertainty as ambiguous commands and asks for user feedback. For future work, classifying fine-grained uncertainty types (e.g., ambiguity in commands, ambiguity in planning, infeasibility from the environment, or infeasibility from the agent capability) is needed. For the SaGC dataset introduced

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

in this paper, the dataset may be biased as it is formulated via a large language model. However, we carefully posit that as the LLM used for data construction does not record the best performance, the effect of these biases is less significant.

VI. Conclusion

In this paper, we focused on classifying and disambiguating the user commands in the context of interactive robotic agents utilizing large language models (LLMs). In particular, we distinguished the user commands into three different types, i.e., clear, ambiguous, and infeasible. To tackle this problem, we first presented the uncertainty estimation method on LLMs. Then, we introduced an approach to classify the type of uncertain goals (ambiguous or infeasible), and interaction for disambiguation in ambiguous commands. Furthermore, we have presented a dataset to validate the situational awareness of the robotic agent. We evaluated the proposed method on this dataset alongside a pick-and-place simulation environment and real-world demonstration. We observed that the proposed method could properly quantify the uncertainty from LLMs and appropriately classify the type of user commands. We believe that the classification and interaction module can be further developed by fine-tuning methods with data consisting of explanations.

References

[1] OpenAI. Gpt-4 technical report. 2023.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proc. of the Conference on Neural Information Processing Systems*, 2020.

[3] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. 2023.

[4] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Proc. of the 6th Annual Conference on Robot Learning (CoRL)*, 2022.

[5] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proc. of the 6th Annual Conference on Robot Learning (CoRL)*, 2022.

[6] Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

[7] Michael Brenner. Situation-aware interpretation, planning and execution of user commands by autonomous robots. In *Proc. of the 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 540–545. IEEE, 2007.

[8] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *Proc. of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023.

[9] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *Proc. of the International Conference on Machine Learning (ICRL)*, pages 9118–9147. PMLR, 2022.

[10] Yang Yang, Xibai Lou, and Changhyun Choi. Interactive robotic grasping with attribute-guided disambiguation. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, pages 8914–8920. IEEE, 2022.

[11] Pradip Pramanick, Chayan Sarkar, Sayan Paul, Ruddra dev Roychoudhury, and Brojeshwar Bhowmick. Doro: Disambiguation of referred object for embodied agents. *IEEE Robotics and Automation Letters*, 7(4):10826–10833, 2022.

[12] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

[13] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.

[14] Yijun Xiao and William Yang Wang. Quantifying uncertainties in natural language processing tasks. In *Proc. of the AAAI conference on artificial intelligence*, volume 33, pages 7322–7329, 2019.

[15] Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *Proc. of the Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[16] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proc. of the Conference of Robot Learning (CoRL)*, 2023.

[17] Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555, 2020.

[18] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

[19] Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*, 2023.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2013.

[21] Charles Richter and Nicholas Roy. Safe visual navigation via deep learning and novelty detection. In *Proc. of the Robotics: Science and Systems Foundation (RSS)*, 2017.

[22] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with large language models. *arXiv preprint arXiv:2212.07769*, 2022.

[23] Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[24] 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. 2023.

[25] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

[26] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proc. of the Conference on Robot Learning (CoRL)*, pages 894–906. PMLR, 2022.

[27] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2022.

[28] Kunal Pratap Singh, Luca Weihs, Alvaro Herrasti, Aniruddha Kembhavi, and Roozbeh Mottaghi. Ask4help: Learning to leverage an expert for embodied tasks. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.