2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Self-Recovery Prompting: Promptable General Purpose Service Robot System with Foundation Models and Self-Recovery

Mimo Shirasaka[1], Tatsuya Matsushima[1], Soshi Tsunashima[1], Yuya Ikeda[1], Aoi Horo[1], So Ikoma[1], Chikaha Tsuji[1], Hikaru Wada[1], Tsunekazu Omija[1], Dai Komukai[1], Yutaka Matsuo[1], Yusuke Iwasawa[1]

***Abstract*— A general-purpose service robot (GPSR), which can execute diverse tasks in various environments, requires a system with high generalizability and adaptability to tasks and environments. In this paper, we first developed a top-level GPSR system for worldwide competition (RoboCup@Home 2023) based on multiple foundation models. This system is both generalizable to variations and adaptive by prompting each model. Then, by analyzing the performance of the developed system, we found three types of failure in more realistic GPSR application settings: *insufficient information*, *incorrect plan generation*, and *plan execution failure*. We then propose the *self-recovery prompting pipeline*, which explores the necessary information and modifies its prompts to recover from failure. We experimentally confirm that the system with the self-recovery mechanism can accomplish tasks by resolving various failure cases.** https://sites.google.com/view/srgpsr

I. Introduction

A general-purpose service robot (GPSR) is a concept aiming to develop a robot system that accomplishes various types of human requests likely to happen in real-world environments [1]. As the system must handle diverse requests in varied environments, it needs to be generalized between them. Besides, the system must handle ambiguous commands in natural human interaction, such as speech, which may lack sufficient information for proper understanding without communication or leveraging common sense knowledge.

Recent progress in foundation models [2], a set of large pre-trained models with diverse datasets, has brought high generalization performances in perception and task planning from natural language to robotics. Furthermore, these models can be adapted to various tasks and environments with *prompting* [3], a technique to enhance the performance of the models by modifying the inputs without additional training. However, most of the robot learning studies utilize foundation models as modules, and there is a lack of discussions about the system design or integration and evaluation of complex environments such as household environments.

This paper first presents a robot system that won first place in RoboCup Japan Open (RCJ) 2023 and second place in RoboCup (RC) 2023 in the GPSR task. The GPSR task in RoboCup aims to benchmark the performance of the entire generalized robotic systems based on the concept of GPSR. To avoid confusion, we use GPSR to represent a task itself and GPSR to represent a concept throughout the paper. The competitions are held in a household environment, and robots are required to perform various tasks asked by a human operator. Figure 1 shows an example of requests

[1]The University of Tokyo. Email: m-shirasaka@g.ecc.u-tokyo.ac.jp

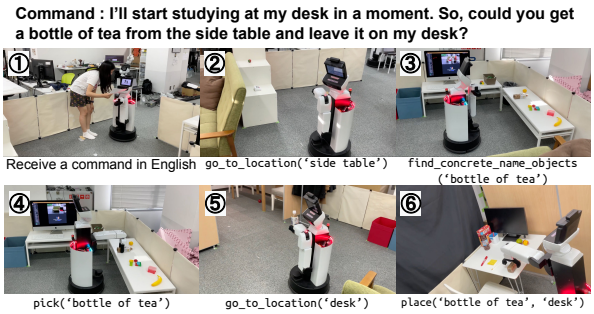


Fig. 1: GPSR task execution example by our system. Commands are converted into a sequence of skills and executed one by one.

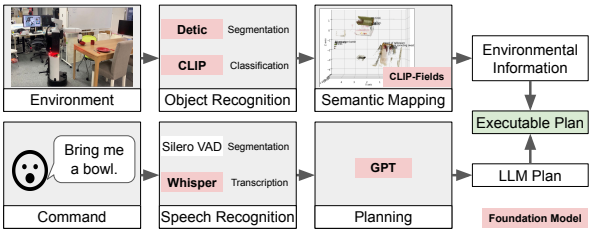


Fig. 2: Overview of our foundation-model-based system. The models collaborate to process the environment and a natural language command into an executable plan.

accompanied by the sequence of output of our system, which integrates multiple foundation models, including *GPT-4* [4] for planning, *Whisper* [5] for speech recognition, and *Detic* [6] and *CLIP* [7] for object recognition (Figure 2). In short, our system uses GPT-4 as the core of the system to generate the plan and the other three models to convert human requests and environmental information into text information or recognize part of the environment specified in the text. Notably, our system can be entirely *promptable*; we can easily tune the system only by specifying prompts (without model training). In section III, we describe more detailed integrations of each foundation model and provide evaluations at both the per-module and the whole system level.

While the achievement of our system in the GPSR task supports the importance of foundation models to realize the concept of general-purpose service robots from the point of generalization and adaptation, there are still several issues regarding its performance; the system still cannot perfectly execute complex requests due to the accumulation of errors in each module, and the entire system becomes difficult to tune as the system grows larger (or by using more foundation models). More critically, the current GPSR task abstracts some desiderata of the GPSR concept due to the nature of robot competition. For example, most information on objects

(names, categories, and locations) is given before the task starts. Thus, there is no need to judge whether the information is sufficient or not. In section IV-A, we categorize three types of failure modes of the current robot system to achieve GPSR systems, namely 1) *insufficient information*, 2) *incorrect plan generation*, and 3) *plan execution failure*, and discuss the requirements of the robot system.

Based on the discussion, we introduce a *self-recovery mechanism* on top of the above-mentioned GPSR system to further enhance the system's versatility. Here, self-recovery means a system that retries to fulfill the original requests when encountering failures. While the notion of self-recovery is simple and has been implemented in various robot systems, we tailor it for promptable robot systems, which can improve performance only by adding or modifying their prompts. Specifically, we design a pipeline called *self-recovery prompting*, which refines prompts based on past experiences and active communication with the operator. For the experiments, we handcraft seven types of commands requiring retry associated with the aforementioned three failure modes of the original system and show that our system can recover from various types of failures.

II. Preliminaries & Related Works

A. General Purpose Service Robot (GPSR)

The concept of GPSR is introduced in Walker et al. [1] wherein robots are expected to perform diverse tasks given by humans in a natural manner (e.g., verbal communication). According to the concept of GPSR, RoboCup@Home league [8] tests the performance of GPSR as `GPSR` task [9]. The `GPSR` task is held in a real-world household environment, and the robots are expected to perform tasks given verbally by the operator (referee) as perfectly as they can within a time limit. The tasks are generated randomly with the command generator [10]. Since the rule is updated every year, we adapt the rule for RC 2023[1] in this paper.

B. Foundation Models for Robotics

Foundation models are a set of models trained on broad datasets at scale and adaptable to a wide range of downstream tasks [2], such as large language models (LLMs) [4, 11–13], vision-language models (VLMs) [6, 7, 14–16], and audio-language models (ALMs) [5, 17, 18]. A key characteristic of the foundation models is their generalization and adaptation ability, thanks to pre-training on massive and diverse datasets (often collected from the Internet). Especially, several foundation models, including Whisper [5], GPT [4, 11], CLIP [7], and Detic [6], are *promptable*; they can enhance the performance by adding text description to the input (called *prompting*) about the contexts such as detailed instruction [19] and environmental information [20].

In the robotics community, foundation models are utilized as modules for perception and planning. As for the perception, VLMs such as CLIP [7], Detic [6], and SAM [15] are utilized in object and environmental perception [19]. Similarly, ALMs such as Whisper [5] and AudioCLIP [18] are used for speech [21] and sound [22] recognition. In addition, several robot systems use LLMs as task planners. LLMs are expected to handle the ambiguity of natural language and convert them to machine-interpretable representations, reasoning missing information in commands. For example, SayCan [23] utilizes LLMs to generate plans given from natural language instructions such as *"I spilled my drink, can you help?."* As the variants, Code as Policies [24] generates Python codes (including calls of external perception modules) and executes them, and Obinata et al. [25] propose to generate state machine [26] using LLMs.

The closest setting and systems to ours is Obinata et al. [25], which proposes a `GPSR` task solution using foundation models in recognition and planning. While the usage of LLMs for planning and VLMs for object detection aligns with ours, we further utilize foundation-model-based modules for speech recognition and semantic mapping. Ours exceeded their performance in the `GPSR` task in RCJ 2023. Additionally, we address the typical failure cases and introduce a novel self-recovery mechanism into the foundation-model-based robot system (section IV).

C. Robot System with Self-Recovery

In the context of robotics, the importance of the notion of self-recovery has been emphasized and implemented in the motion planning of multi-legged robots [27] and the mechanical design of aerial robots [28]. This paper aims to realize self-recoverable task planning in GPSR systems under the framework of prompting with foundation models.

Some concurrent robot learning studies using foundation models provide solutions for managing failures in plan execution. For example, DoReMi [29] proposes to detect failures of skill execution via VLMs and replan if the skill fails. FindThis [30] proposes to resolve the ambiguity in object recognition through human-robot dialogue. KnowNo [31] presents a framework to ask humans for help in an interactive manner if the planning uncertainty is high. REFLECT [32] introduces a failure reasoning framework based on a robot's past experiences using LLM. In contrast, this paper presents the entire GPSR system in real-world household environments, which is promptable and has functions to address multiple types of failures autonomously.

III. Promptable System for `GPSR` Task

In this section, we first introduce our promptable GPSR system with foundation models, which won second place in the `GPSR` task and third prize in RoboCup@Home 2023.

For the realization of a GPSR system, multiple foundation models with high generalization and adaptability were leveraged for the system in this study. The following five models (four of which are foundation models, and one is a model that consists of an integration of foundation models) have the ability to enhance the system to be generalized and adaptive with prompting: Whisper [5] for speech recognition, GPT-4 [4] for task planning, Detic [6] for object detection and segmentation, CLIP [7] for object classification, and CLIP-Fields [35] for integration of environmental information. Figure 2 shows an example of how the foundation models can be used in our proposed system.

[1] https://github.com/RoboCupAtHome/RuleBook/tree/706764626baf073d56ab2e61c1a3c5d3c339cfb4

TABLE I: 21 Skill Functions.

Functions(Arguments)	Descriptions
go_to_location(location)	navigate the robot to {location}
ask_location(object)	ask location of {object} using Whisper, or if unsuccessful, infer with LLM
find_concrete_name_objects(object,room)	find {object} using Detic and CLIP in the {room}
find_category_name_objects(category,room)	find {category} objects using Detic and CLIP in the {room}
count_concrete_name_objects(objects)	count the number of {objects} using Detic and CLIP
count_category_name_objects(category)	count the number of {category} objects using Detic and CLIP
find_person(person)	find {person} using Keypoint R-CNN [33]
detect_person_pose(person)	detect {person} 's pose using Keypoint R-CNN
find_specific_pose_person(person,pose)	find {person} with {pose} using Keypoint R-CNN
count_specific_pose_person(person,pose)	count the number of {person} with {pose} using Keypoint R-CNN
count_person	count the number of person using Keypoint R-CNN
follow_person(person,location)	follow {person} to {location} using YOLOv8 [34]
guide(person,location)	guide {person} to {location}
pick(object,location)	pick {object} at {location}
hand_over(object,person)	hand over {object} to {person}
ask_person_to_hand_over(object,person,query)	ask{person} to hand over {object} by saying {query}
place(object,location)	place {object} on {location}
ask_question(person,question)	say {question} to {person} and get answer using Whisper and LLM
answer_question(opt: person)	answer to {person}'s question using Whisper and LLM
tell_information(information,person)	tell {information} to {person} using LLM
operate_door(location,operation)	{operation} (open/close) the door at {location}

For all the experiments, we used HSR (Human Support Robot) developed by Toyota Motor Corporation [36]. For computation, we used a laptop with RTX3080 mobile 16GB GPU. The experiments were conducted in a real-world simulated household environment with several rooms, such as a living room, a dining room, and a study room.

A. Overview

1) Speech Recognition: Speech recognition consists of two modules: a voice activity detection (VAD) module and a transcription module. Silero VAD [37] is used for VAD, and Whisper [5] is used for transcription. Since Whisper is promptable with natural language, transcription performance can be enhanced using prior knowledge about task settings, such as names of humans, objects, and locations.

2) Object Recognition: The object recognition module consists of a detection module and a classification module. Detic [6] and CLIP [7], both promptable foundation models, were used for detecting and classifying objects, respectively. We leverage the feature that these models accept open-vocabulary text inputs as prompts for object detection and classification, while conventional pre-trained models usually have fixed classes. For object detection, we employed two-stage object recognition to prevent oversights. We first segment images using Detic with prompt information of objects of interest (e.g., object name, category, description) and crop the target object area. Then, the cropped images are classified with CLIP based on similarities between the embeddings of text description of the target objects and the embeddings of the segmented images.

3) Planning: To convert a natural language command into an executable format, we leverage GPT-4 [4] in our system. We prepare 21 skill functions (Table I) that can accomplish given commands if appropriately combined. The desired output is an array where skill functions, including their arguments, are correctly arranged in the order they are executed by the robot in JSON format [38].

The task planning is based on the Chain-of-Thought prompting [39, 40] and has a two-step structure. The first step is dividing the command into minimal steps and deciding the order to perform in natural language. For example, the command *"bring me an apple from the dining table"* is converted into an array of sentences such as *"Move to the dining table," "Find apple,"* and similarly. The array continues in the order of execution. In the second step, skill functions to be used with their arguments (e.g., locations, object names) are decided for each sentence leveraging function calling of GPT-4. By providing examples of the commands and their desired responses as prompts, it is possible to specify the output format and improve task planning accuracy.

4) Semantic Mapping: We integrate environmental information into a 3D semantic map using CLIP-Fields [35], which utilize three foundation models: Detic for object recognition, CLIP for image encoding, and Sentence BERT [41] for image label encoding. The robot can refer to the information in CLIP-Fields for task planning.

B. Experiments of Each Module

1) Speech Recognition: We compared the speech recognition performance with and without prompts. The prompt includes names of objects, humans, and locations (i.e., room and furniture) that may appear in commands. 12 commands were used for the experiments. The commands were generated by the command generator used in the Enhanced General Purpose Service Robot (EGPSR) task of RoboCup@Home 2023. 14 people participated in this study. The examinees were asked to read each command aloud once to reduce misread cases. Then, they were asked to read the same command twice, and the robot recognized their voices. The typical cases from the obtained results are indicated in Table II. As for location names, prompting in advance shows a reduced likelihood of variations in interpretation. This suggests that pre-defined location names as prompts are an effective technique for improving transcription performance.

2) Task Planning: The planning performance between using tuned prompts and minimal prompts was examined in comparison. To test the effect of providing a prompt on the LLM's reasoning ability of translation from given commands

into the sequence of execution steps, we compared the result of the first step of the task planning in section III-A.3.

The tuned prompt was adjusted so that most of the generated commands from the command generator [10] used in the EGPSR task are correctly converted into arrays of sentences. This prompt consisted of the settings of the environment, the situation the robot was in, and the iteration of example commands and their ideal responses. Since the alignment of the LLM output (i.e., an array of sentences) was impossible without a prompt, the minimal prompt (shown below) was designed with minimum sufficient content for eliciting the output format.

> *You are a helpful assistant for a robot. The robot is in a house. Your mission is to convert natural language command into a list of sentences. The robot will execute the sentences in order to complete the task.*

The commands used in this experiment were the same as in section III-B.1. The success or failure of planning for each output was judged by whether the command was completed when the robot performed each skill function flawlessly.

As a result, in many cases, plans generated with the minimal prompt were inappropriate, whereas those with the tuned prompt were executable. Table III displays some commands and their outputs for each prompt. The outputs of the minimal prompt lacked necessary preliminary action or contained sentences that could not be related to any skill function. Therefore, it can be said that providing instructions as a prompt is effective in eliciting LLM to generate executable plans.

3) Object Recognition: Object recognition performance was evaluated by comparing setting Detic for open-vocabulary mode with prompts and closed-vocabulary mode without prompts. Experiments were conducted using images with the same member of objects throughout the experiment.

CLIP was used consistently with prompts, and for both open-vocabulary Detic and CLIP, prompts were tuned using images of the same objects placed in different locations and orientations. For instance, the prompts for "white rope" and "jump rope" were set as follows.

> Prompts of a white rope and a jump rope for Detic
> *"rope": "a photo of a tangled white rope",*
> *"jump rope": "a photo of a green jump rope, a type of toy"*

> Prompts of a white rope and a jump rope for CLIP
> *"white rope": "a photo of a white rope",*
> *"jump rope": "a photo of a green jump rope"*

Validation tests were conducted using entirely new images.

Figure 3 shows that when Detic was used in open-vocabulary mode with the prompts shown above, it correctly detected the white rope, which was present in the closed-vocabulary case but remained undetected. During the segmentation with Detic, the white rope was misidentified as a green jump rope. Nevertheless, by incorporating prompts, even for objects with similar shapes, segmentation accuracy improved, and when applied to CLIP, correct recognition, as demonstrated in this case, could be expected. The result suggests the potential for improved recognition accuracy.

TABLE II: Comparison of transcription results (without and with prompts) for speech recognition with Whisper.

Command: *Go after the person at the bed please.*	
w/o Prompts	w/ Prompts
person of the band, please person at the bat place	person at the bed please person at the bed please
Command: *Offer something to drink to all the people dressed in white in the bedroom.*	
w/o Prompts	w/ Prompts
dressed in white in the bathroom	dressed in white in the bedroom

TABLE III: Comparison of generated plans (with minimal and tuned prompts) with GPT-4. "Success" indicates that a plan that would satisfy the command if each skill function was performed perfectly was generated.

Command: *Describe the objects on the kitchen table to me please*	
Minimal	Tuned
Try to find the object before going to the kitchen table	Success
Command: *Robot please retrieve the tropical juice from the side table, grasp the apple from the end table, and speak*	
Minimal	Tuned
Try to grasp the tropical juice before detecting Try to grasp the apple before releasing tropical juice Ambiguous sentence ("*Activate speech function.*")	Success

C. Results of RoboCup@Home GPSR task

We participated in RoboCup@Home Domestic Standard Platform League of RCJ 2022 and 2023 and RC 2023. The proposed system was evaluated in RCJ 2023 and RC 2023. In the competitions, the scores of the GPSR task were respectively given when speaking the transcribed command and completing the task. It should be noted that the case where the robot autonomously requested human help and continued the command execution was also regarded as a success, with a reduction of scores afterward. Notably, in our RCJ 2023 trial, all commands were completed within the time limit, scoring 170 points, the maximum score for the second-most challenging category (Category 2). Our team marked more than 180 % of the second-placed team.

IV. Self-Recovery Mechanism for Promptable Robot System

In the previous section, we proposed the entire system for GPSR task in RoboCup@Home, which can achieve top-level performance. However, owing to the nature of robot competition, some desiderata of GPSR are abstracted in GPSR task. For instance, most object information (names, categories, and locations) is provided beforehand, removing the necessity to assess whether the command contains sufficient information. Moreover, due to time constraints, skipping tasks can yield higher scores than recovering from failed commands. Therefore, achieving a higher score or winning in GPSR task does not equate to genuine GPSR system achievement.

In this section, we categorize challenges in attaining genuine GPSR. Ideally, GPSR can be achieved with complete environment information, accurate plan generation (skill sequences), and flawless skill execution. However, in general,

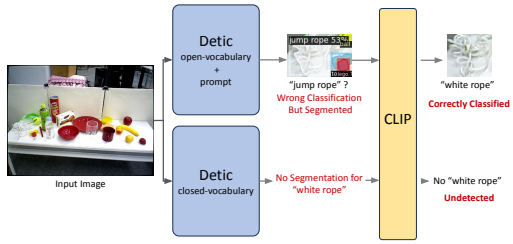


Fig. 3: Image recognition results with open and closed vocabulary modes of Detic. As discussed in section III-B.3, "white rope," undetected with closed-vocabulary mode, is successfully detected with open-vocabulary mode.

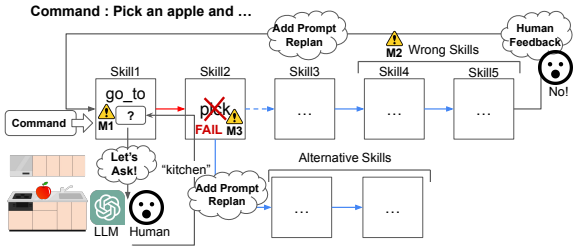


Fig. 4: Example of GPSR system's three failure modes and prompt-based self-recovery mechanisms. M1: Lack of location information triggers the system to ask a human or LLM and incorporates obtained information. M2: Incorrect performance triggers replanning. M3: Execution failure activates the per-skill recovery function.

these assumptions are often violated, posing challenges to GPSR realization. Here we analyze common GPSR issues, organizing failure modes into three categories: *insufficient information*, *incorrect plan generation*, and *plan execution failure*. Then, we propose to add a *self-recovery mechanism* into the system and evaluate the performance under the settings of the aforementioned three failure modes.

A. Three Failure Modes of GPSR Systems

(M1) Insufficient Information: In a domestic environment, robots have to perform in a dynamic environment; for example, the locations of objects and humans are ever-changing. Moreover, registering all the information about the environment (e.g., object or human names, categories, and locations) to the system beforehand is not feasible. Even if the system has enough reasoning or ability to recognize human intent, lacking information about the environment prohibits the system from generating the correct plan at once.

The information necessary to plan can be lacking in many ways, such as *"I lost my watch. Could you find it for me?"* (a situation where humans do not know the location of the objects), or *"Could you bring me a cup?"* (a situation where humans know where it is but do not clarify in the command).

(M2) Incorrect Plan Generation: Even when the system has information sufficient to accomplish the task (i.e., no insufficient information problem), the current system in section III cannot perfectly accomplish the task. For example, the robot can catch noises along with spoken commands and mistranscribe them, which leads to the generation of a wrong plan. Moreover, wrong plans can be generated due to a lack of reasoning performance or common sense of the planner. Suppose a simple case where the planner just extracts verbs in the order of appearance in the commands and makes them into executable skill sequences, and the command is "Could you fetch me an apple?" the plan may start with "Bring the apple to the operator," which is a mistake. Instead, the ideal plan is to "go to the location where the apple is," "find it," "grab it," and then "bring it back to the operator."

TABLE IV: Commands tested in section IV-C. Blue text indicates the information to navigate is sufficient, and red text indicates the information to navigate is insufficient. Our self-recovery prompting pipeline successfully recovered from all failure cases.

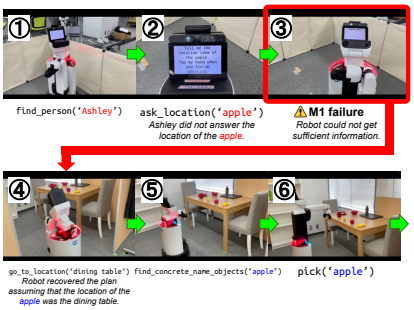
Command	Failure Modes M1	M2	M3
1. Could you bring me an apple from the side table?	✓		✓
2. Hi HSR, I am starting to feel hungry so could you grab an apple from dining table and put it on my desk? I will be there in a moment.	✓		✓
3. I lost my mug so could you find it for me?	✓		
4. Thank you, HSR. I am getting tired. Could you prepare a fruit for me on the side table? I will have some rest at the sofa in a moment.	✓		✓
5. Could you help me find the apple that I bought the other day? Ashley might know where it is, so maybe you can ask her if she knows where it is. When you find it, please bring it to me.	✓		
6. Could you bring me the apple from the stair-like shelf?		✓	
7. Could you look for Ashley in the dining room and ask her if she wants dinner at home tonight?			✓

(M3) Plan Execution Failure: Even if the plan is generated correctly with sufficient environmental information, the robot system may fail to execute skills, a common issue in real-world systems. This occurs due to imperfections in skill execution. For instance, robots may miscompute manipulation poses and fail to grasp objects. The inability to find an object or false positives is also considered an execution failure. It is crucial to note that execution failure can result from hardware errors or environmental factors (e.g., non-existent objects in the environment).

B. Self-Recovery Prompting Pipeline

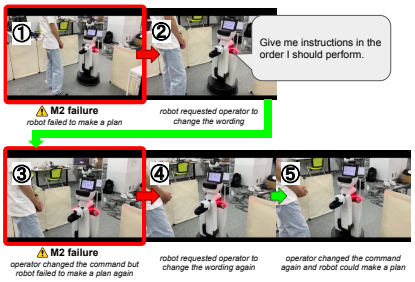
To deal with the three failure modes for realizing GPSR systems, we introduce a self-recovery mechanism from the failure modes with prompting, called *self-recovery prompting pipeline*, as illustrated in Figure 4. In concrete, we developed a self-recoverable GPSR system as an entire system by replanning and human-robot interaction based on the foundation-model-based system described in section III.

1) Recovery for Insufficient Information (M1): In the case of insufficient information (M1), the missing information necessary for planning is supplemented with common sense that the planning module has (e.g., food is likely to be in the kitchen or dining room) and additional information obtained by talking with humans (e.g., asking where the apple is). In concrete, two recovery functions are implemented in our GPSR system. If the location name (e.g., dining table) is missing from the command (or dialogue with humans), the system infers a potential location leveraging LLM and plans to visit there. If a location name referenced by the operator or LLM output is undefined in the system, the robot asks the operator to rephrase the location name and extracts the location using LLM from their response.



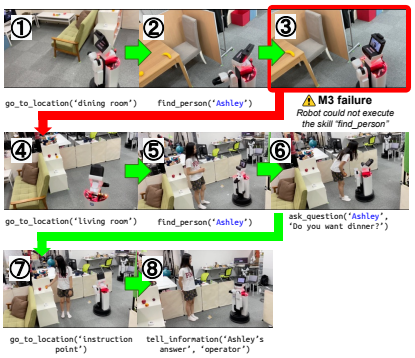
command 5

Could you help me find the apple that I bought the other day? Ashley might know where it is, so maybe you can ask her if she knows where it is. When you find it, please bring it to me.



command 6

Could you bring me the apple from the stair-like shelf?



command 7

Could you look for Ashley in the dining room and ask her if she wants dinner at home tonight?

Fig. 5: Example of three failure modes and our system's execution with self-recovery prompting. Commands 5, 6, and 7 in Table IV correspond to the left, middle, and right, respectively. Red-box-highlighted areas indicate failure patterns at each command. Green arrows indicate a normal plan transition and red arrows indicate that a recovery plan has been triggered.

2) Recovery for Incorrect Plan Generation (M2): In the case of incorrect plan generation (M2), we develop solutions regarding command recognition and plan generation. Promptable speech recognition module (e.g., Whisper) can be enhanced by updating the prompts as described in section III. For plan generation, the prompts for the LLM-based planner are updated, reflecting human feedback given to the system after finishing the original plan to confirm task completion. If the task is not deemed complete, a new plan is regenerated with updated prompts.

3) Recovery for Plan Execution Failure (M3): In the case of plan execution failure (M3), recovery can be per-skill or per-plan. Per-skill recovery involves two functions: one retries skill execution within the plan (e.g., navigation), and the other generates alternative skill sequences using the following prompt template.

> *The robot is supposed to* {task_content} . *The robot tried to* {failed_action} {robot_at}, *but failed. What should the robot do next?*

Per-plan recovery occurs when the task is deemed a failure based on human feedback after executing the entire plan, akin to addressing the 2nd failure mode (M2). In this case, the prompts of the LLM-based planner are updated with the feedback, and the entire plan is regenerated and executed. For instance, if an incorrect object is recognized during object recognition, the system prompts the operator for more information, particularly the object's name and color. Prompts for the object recognition module are then updated, and the task plan is regenerated.

C. Experiments

1) Experiment Setup: Experiments were conducted to examine the system's ability to recover from each of the failure modes using our proposed approach. The system is tested in a domestic environment similar to that of section III. The difference from the setting in the previous section is that object and human names, and their locations were not provided beforehand (a map with location names was provided). Following the experimental purposes, commands for the tests were manually created instead of generated using the command generator. All commands were expected to be too challenging with the original system in section III. Table IV lists the prepared commands, with a checkmark (✓) denoting the presence of corresponding failure mode characteristics.

2) Results: Our self-recovery prompting mechanism successfully resolved failures for all tested commands. Three of the seven results, exemplifying recovery functions in accordance with M1, M2, and M3, are detailed below and illustrated in Figure 5.

For the 5th command, the robot asked Ashely for the apple's location but received no response, potentially halting the system due to lack of information. However, the developed system overcame this potential failure point by seeking general knowledge of LLM for the location of *"apple"*.

For the 6th command, a phrase (*"apple from the stair-like shelf"*) was difficult to transcribe, making plan generation difficult. Our system addressed this by requesting the operator rephrase the command.

For the 7th command, an execution failure at the finding person phase was a possible failing point. The system recovered from it by re-planning.

V. Discussion and Conclusion

In this paper, we first developed promptable GPSR systems utilizing multiple foundation models, which can achieve top-level performance in the worldwide competition (RoboCup@Home 2023). By analyzing the performance of the developed system, we organized three failure modes in more realistic GPSR applications: *insufficient information*, *incorrect plan generation*, and *plan execution failure*. We then proposed the self-recovery prompting pipeline, which leverages the prompting of the system to overcome each failure mode, and evaluated the entire system using seven handcrafted commands. To enhance further studies in GPSR systems with self-recovery, benchmarks equipped with adaptive human-robot interaction in more natural domestic environments will be essential to standardize the performance, which may also be realized with LLMs and VLMs.

Acknowledgement

This research was supported in part by KIOXIA Corporation.

References

[1] N. Walker, Y. Jiang, M. Cakmak, and P. Stone, "Desiderata for planning systems in general-purpose service robots," *arXiv preprint arXiv:1907.02300*, 2019.

[2] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the Opportunities and Risks of Foundation Models," *arXiv preprint*, 2021. [Online]. Available: https://crfm.stanford.edu/assets/report.pdf

[3] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[4] OpenAI, "GPT-4 Technical Report," *arXiv e-prints*, p. arXiv:2303.08774, Mar. 2023.

[5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[6] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *European Conference on Computer Vision*. Springer, 2022, pp. 350–368.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[8] T. Wisspeintner, T. Van Der Zant, L. Iocchi, and S. Schiffer, "RoboCup@ Home: Scientific competition and benchmarking for domestic service robots," *Interaction Studies*, vol. 10, no. 3, pp. 392–426, 2009.

[9] L. Iocchi, D. Holz, J. Ruiz-del Solar, K. Sugiura, and T. Van Der Zant, "RoboCup@ Home: Analysis and results of evolving competitions for domestic and service robots," *Artificial Intelligence*, vol. 229, pp. 258–281, 2015.

[10] RoboCup@Home, "RoboCup@Home Command Generator," https://github.com/kyordhel/GPSRCmdGen.git, 2015.

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[12] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and Efficient Foundation Language Models," 2023.

[13] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "PaLM: Scaling Language Modeling with Pathways," 2022.

[14] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning*. PMLR, 2022, pp. 12 888–12 900.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[17] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[18] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending clip to image, text and audio," in *ICASSP 2022-2022 IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 976–980.

[19] T. Matsushima, Y. Noguchi, J. Arima, T. Aoki, Y. Okita, Y. Ikeda, K. Ishimoto, S. Taniguchi, Y. Yamashita, S. Seto *et al.*, "World robot challenge 2020–partner robot: a data-driven approach for room tidying with mobile manipulator," *Advanced Robotics*, vol. 36, no. 17-18, pp. 850–869, 2022.

[20] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "ChatGPT for Robotics: Design Principles and Model Abilities," Microsoft, Tech. Rep. MSR-TR-2023-8, February 2023. [Online]. Available: https://www.microsoft.com/en-us/research/publication/chatgpt-for-robotics-design-principles-and-model-abilities/

[21] S. Liu, A. Hasan, K. Hong, R. Wang, P. Chang, Z. Mizrachi, J. Lin, D. L. McPherson, W. A. Rogers, and K. Driggs-Campbell, "DRAGON: A Dialogue-Based Robot for Assistive Navigation with Visual Language Grounding," *arXiv preprint arXiv:2307.06924*, 2023.

[22] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," *arXiv preprint arXiv:2303.07522*, 2023.

[23] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[24] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[25] Y. Obinata, N. Kanazawa, K. Kawaharazuka, I. Yanokura, S. Kim, K. Okada, and M. Inaba, "Foundation Model based Open Vocabulary Task Planning and Executive System for General Purpose Service Robots," *arXiv preprint arXiv:2308.03357*, 2023.

[26] J. Bohren and S. Cousins, "The SMACH High-Level Executive [ROS News]," *IEEE Robotics & Automation Magazine*, vol. 17, no. 4, pp. 18–20, 2010.

[27] S. Peng, X. Ding, F. Yang, and K. Xu, "Motion planning and implementation for the self-recovery of an overturned multi-legged robot," *Robotica*, vol. 35, no. 5, pp. 1107–1120, 2017.

[28] A. Briod, A. Klaptocz, J.-C. Zufferey, and D. Floreano, "The AirBurr: A flying robot that can exploit collisions," in *2012 ICME International Conference on Complex Medical Engineering (CME)*. IEEE, 2012, pp. 569–574.

[29] Y. Guo, Y.-J. Wang, L. Zha, Z. Jiang, and J. Chen, "DoReMi: Grounding Language Model by Detecting and Recovering from Plan-Execution Misalignment," *arXiv preprint arXiv:2307.00329*, 2023.

[30] A. Majumdar, F. Xia, brian ichter, D. Batra, and L. Guibas, "FindThis: Language-Driven Object Disambiguation in Indoor Environments," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=nNsZxc2cmO

[31] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, Z. Xu, D. Sadigh, A. Zeng, and A. Majumdar, "Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=4ZK8ODNyFXx

[32] Z. Liu, A. Bahety, and S. Song, "REFLECT: Summarizing robot experiences for failure explanation and correction," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=8yTS_nAILxt

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[34] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[35] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," *arXiv preprint arXiv:2210.05663*, 2022.

[36] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of human support robot as the research platform of a domestic mobile manipulator," *ROBOMECH journal*, vol. 6, no. 1, pp. 1–15, 2019.

[37] S. Team, "Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier," https://github.com/snakers4/silero-vad, 2021.

[38] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of JSON schema," in *Proceedings of the 25th international conference on World Wide Web*, 2016, pp. 263–273.

[39] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[40] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[41] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 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)*, 2019, pp. 3982–3992.