Proceedings of the 2024 IEEE
International Conference on Robotics and Biomimetics
December 10-14, 2024, Bangkok, Thailand

# Think Before Execute: Embodied Reasoning of Supportive Tools for Robot Service with Large Language Models

Jin Liu, Tianyu Fu, Fengyu Zhou, Chaoqun Wang

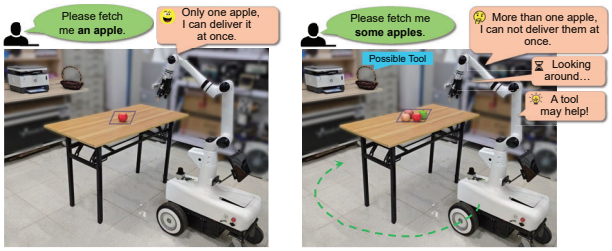
***Abstract*— The utilization of appropriate tools can significantly streamline task complexity and expand the boundaries of robot capabilities, indicating a high level of intelligence. Finding a useful tool is a non-trivial problem when the robot confronts complex tasks lacking detailed execution instructions. In this paper, we investigate the problem of robotic task execution in response to human verbal requests. Motivated by the development of Large language models (LLMs), we propose a robotic embodied reasoning framework conditioned on LLMs. It enables robots to rely on their physical manipulation system or exploit supportive tools in the scene to effectively complete their tasks. To this end, the environment knowledge about objects is modeled using a scene graph. More specifically, by utilizing the on-site knowledge in the scene graph and the general knowledge in LLMs, we develop an embodied evaluation module to conduct task executable analysis considering the robot's abilities and the associated objects in the environment. Then a series of execution sequences are generated to drive the robot to fruitfully finish the task. Extensive experiments conducted in various real-world scenes demonstrate the effectiveness and generalization of our proposed framework.**

## I. Introduction

"*Fetch me some apples from the table in the kitchen.*"–Requiring a service robot to execute such a straightforward task remains highly challenging, necessitating high-level task execution capabilities. To solve such embodied problems [1], a robot is required to possess embodied reasoning ability. It allows the robot to analyze the task and its surroundings, evaluate its own physical manipulation ability or explore potential supportive tools[2][3], and generate task execution sequences [4] in reaction to human requests.

Jin Liu, Tianyu Fu, Fengyu Zhou, and Chaoqun Wang are with the School of Control Science and Engineering, Shandong University, China. Email address: {202120638,ftyu}@mail.sdu.edu.cn,{chaoqunwang,zhoufengyu}@sdu.edu.cn.

This work was supported by Meituan Academy of Robotics Shenzhen and in part by Natural Science Foundation of Shandong Province under Grant NO.ZR2021QF122 .




(a) Task with self capability (b) Task with tool utilization

Fig. 1. An example of task execution. (a) Robots can rely on their own motion and control systems to pick up the apple from the table. (b) The robot cannot pick up all the apples on the table at once, it needs to use tools such as baskets to complete this task.

Recently, the rapid development of Large Language Models (LLMs) [5], [6] has brought about revolutionary advances in linguistic comprehension. The capability of LLMs to interpret user intention from natural queries and generate problem-solving plans based on human requests showcases their significant potential in robotics applications [7], [8]. This has sparked substantial interest in robotics to develop the embodied reasoning ability for diverse tasks, wherein LLMs serve as the cognitive core to direct robots' decision-making processes. Following this trend, recent studies first employ pre-trained vision-language models to extract visual object features, including shapes, positions, and then convert the features into textual descriptions. Subsequently, the descriptions with designed prompts are fed to the LLMs to obtain specific task plans [9], [10]. Despite the recent progress, the generation of task plans with LLMs typically lacks empathy for the robot encountered scenario, often resulting in disembodiment and context-free solutions. The embodied reasoning capability with LLM exploiting the robot's body and onsite environment supports is still less investigated.

In this paper, we present a robotic embodied reasoning framework conditioned on the LLMs for complex tasks in our daily lives, which mainly involves three modules: language comprehension module, embodied evaluation module, and tool utilization module. More specifically, the first module enables

2024 IEEE International Conference on Robotics and Biomimetics (ROBIO) | 979-8-3315-0964-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ROBIO64047.2024.10907522

the robot to utilize LLMs to comprehend human intentions and analyze the task. We develop an embodied evaluation module for the robot, considering the surrounding objects and the robot's own abilities. This module aims to streamline task complexity with the potential help of supportive tool utilization. As the examples shown in Figure 1, when the human requires the robot to fetch one apple, the robot relies on its physical execution system to accomplish it without any tool utilization. In contrast, if we require the robot to fetch more apples, the robot can still deliver them one by one, resulting in extremely low task execution efficiency. As a consequence, in the tool utilization module, the robot should exploit supportive tools (*e.g.* fruit basket) to strengthen its ability (*e.g.* deliver the apples at once) and streamline the task complexity. Finally, after the robot completes all the tasks, it updates the overall knowledge base to ensure the reliability and effectiveness of the next task execution. In addition, we further discuss the limitations and list potential technical solutions, hoping to bring some inspiration for future research.

Our main contribution lies in strengthening the robotic embodied reasoning capability conditioned LLMs. To better transfer the knowledge from the LLMs to the local environment, we establish a scene graph to describe the surroundings instead of organizing tedious prompts. Then we design an embodied evaluation module that evaluates a task considering the constraints of the robot body and the supports from the environments, which make the task solution originate from the deep analysis of the on-site scenario. We demonstrate that the proposed framework enables the robot to explore supportive tools or rely on its own physical manipulation system to streamline the task complexity and accomplish the task effectively. Extensive experiments in the real world demonstrate the effectiveness and generalization of our proposed framework.

## II. Problem Formulation

In this paper, we aim to endow robots, such as a mobile manipulator robot, with the capacity to autonomously depend on assessments of their own capabilities, incorporating available tools to proficiently address a variety of tasks. This task requires the robot to comprehend the natural language instructions $\mathcal{L}_\pi$ based on the large language models (LLMs), understand the complex environment $\mathcal{S}$, and execute tasks $\mathcal{T}_\pi$ by sequential skills $\mathcal{M}$ composed of basic actions $\mathcal{A}_\pi$, according to its embodied constraints $\mathcal{C}$, *i.e.* $\mathcal{M} = f_\theta(\mathcal{T}_\pi|\mathcal{L}_\pi, \mathcal{S}, \mathcal{C}, \mathcal{A}_\pi)$, where $f_\theta$ denotes planner $f$ with parameters $\theta$.

While LLMs can provide sufficient commonsense knowledge and reasonable planning procedures, they simply rely on a large amount of corpus in the early stages of model training to generate task sequences, making it difficult to meet the needs of robot task execution. Consequently, to make up for this deficiency, in this paper, we argue that LLMs need to consider the robot's own capabilities and leverage its reasoning ability to generate the planning and execution of action sequences to achieve true embodied intelligence.

## III. Methods

### A. Overview

An overview of our proposed framework is illustrated in Figure 2, consisting of three main components: 1) Language comprehension module, 2) Embodied constraint evaluation module, and 3) Tool utilization module. In specific, the robot first comprehends the given instructions via the language comprehension module and navigates to the target position conditioned on the pre-built scene graph. Subsequently, to make full potential utilization of its ability, the robot leverages LLMs to conduct task executable analysis while considering the embodied parameters and scene graph. If the robot can not complete the task within its own ability, it will actively explore available tools to solve the problems and update the knowledge graph after completing tasks, making it convenient for the robot's next application. The details of each module in our framework will be introduced in the following sections.

### B. Language Understanding

Existing LLMs exhibit powerful abilities in logical reasoning and language comprehension [11], [12], and have been widely utilized in robotic applications [6], [13]. In this module, we aim to enable the robot with the ability to *understand human intentions* and *analyze tasks* by making full utilization of LLMs.

Subsequently, conditioned on the knowledge graph, the module will retrieve relevant skills composed of basic actions for the navigation and manipulation tasks. In specific, we first pre-construct a hierarchical scene graph $\mathcal{G} = (V, E)$ for the whole environment, where $V \in \{V_r, V_k, V_o\}$, with $V_r$ signifying root house node, $V_k$ denoting the pre-defined key object node, $V_o$ representing the objects that are adhere to $V_k$. To acquire better demonstrations for

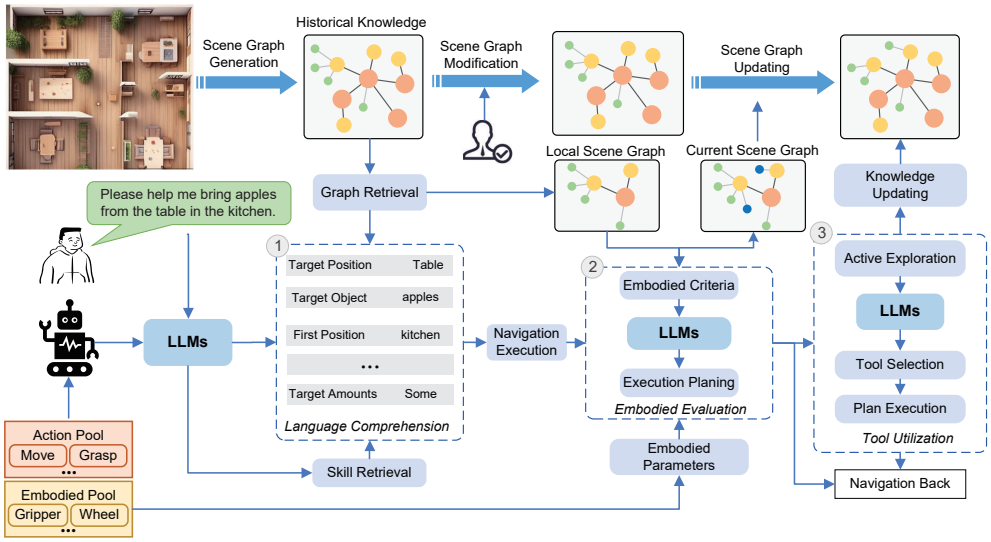



Fig. 2. Architecture of the overall framework. LLMs play an important role in three modules, namely language comprehension module, embodied constraint evaluation module, and tool utilization module. Conditioned on the retrieved scene graph knowledge and its own skills (including manipulation skills and movement skills), the robot first conducts the semantic analysis of human intention and task feasibility using LLMs. Once the robot reaches the target position, it assesses its capabilities and physical parameters to ascertain if it can accomplish the task autonomously without the need for tools. When tools are required, the robot utilizes a scene knowledge graph to search for appropriate tools and accomplish the given task. After the task is completed, the robot will proceed to update the current scene graph to facilitate the execution of subsequent tasks.

the environment, we require three human experts to validate and modify the generation results. Then, the robot will determine whether to execute path planning based on the graph analysis. When there is a conflict between the desired position in the planned path $P$ and the graph nodes (*i.e.* $\exists p \in P,\ p \notin V$), and the hierarchical architecture between the path sequence $\{p_1, p_2, \cdots, p_l\}$ and the scene graph $\mathcal{G}$ (*e.g.* $\exists p \in V_r, p \in V_o, p \notin V_k$ ), it needs to interact with people to determine the conflict location. Meanwhile, the module retrieves the potential skills, *e.g.* hold objects and social navigation [14], from the LLMs, which are composed of basic actions, *e.g.* grasping detection and moving forward. In instances where the robot engages in tasks beyond its skill set, it seeks interaction with humans to ascertain the subsequent steps in execution. Illustratively, robots possess the capability to move and navigate for obstacle avoidance and reach designated targets; however, they lack the capacity to execute actions like jumping over obstacles. By fully considering the above navigation and skill concerns, the robot will perform the next step of operation.

### C. Embodied Constraint Evaluation

When the robot arrives at the required location conditioned on the planned paths that are generated from the LLMs, it will fully consider its own limitations and capabilities, and execute the manipulation procedures. Due to its constrained visual perspective, the robot concentrates on the manipulated object, thereby establishing a dedicated operational workspace. Subsequently, it retrieves local scene graph $\mathcal{G}_L$ related to the desired objects belonging to the workspace. Notably, the spatial and semantic relationships among the objects are constantly changing, robots thus need to perceive the current scene graph $\mathcal{G}_C$ and analyze and merge the retrieved local scene graph $\mathcal{G}_L$ with the current perceived graph. Conditioned on the merged scene graph $\mathcal{G}_W$, the robot will leverage the LLM to retrieve skills and then combine their own embodied parameters, such as the opening and closing angle of their gripper, and the angle at which they can move and rotate, to accomplish the task. If the task can be completed by its own, *e.g.* when the human requires the robot to grasp an apple or when there is only one apple on the table, the robot can perform the apple grasping operation by retrieving grasping skills, *i.e.*Grasp(object)$\Rightarrow$ Grasp(apple). In this paper, we utilize a pre-trained grasping detection network [15] to generate optimal grasping candidates. Nevertheless, due to their own limitations, the robots complete the task in some scenes with low efficiency. *E.g.* when the human asks the robot to grasp a few apples, due to the fact that their skill pool only has the ability to grasp a single object, the robot can only deliver the apples one by one. In this case, one auxiliary tool, for example, a fruit basket, needs to be used to help

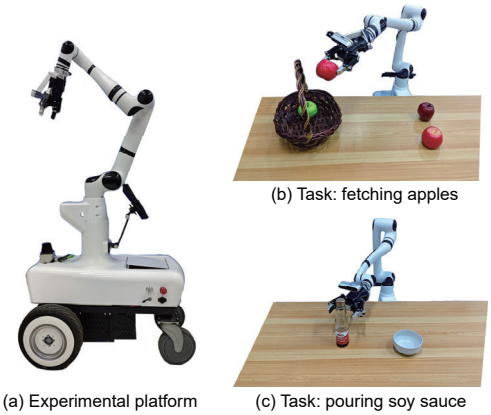



Fig. 3. (a) Real robot experimental environment. (b) and (c) show examples of tasks used in our experiments.

the robot to improve the efficiency.

### D. Robot Tool Utilization

When the robot can not complete the task by its own manipulation or complete the task with low efficiency, it should utilize extra tools to solve such an issue. In specific, when the human asks the robot to bring some apples from the table in the next kitchen as shown in Figure 2, based on the above embodied evaluation, it cannot pick up so many apples at once. At this point, it will use LLMs and the pre-built scene graph information to obtain a series of candidate tools to accomplish the task. The process can be formulated as:

$$\{o_1, o_2, \cdots, o_n\} \Leftarrow \texttt{LLMs}(\mathcal{T}_\pi^*), \quad o \in V \tag{1}$$

Conditioned on candidate tool lists and scene graphs, the robot explores the current environment to quickly find suitable tools. Nevertheless, some candidate tools may be far away from the current position of the robot, and it can be very time-consuming to reach that position. Therefore, we utilize the bottom-up strategy that the priority of tools in $V_o$ is higher than in $V_k$ to sort the candidate tools. Then, the robot will get the tool (*e.g.* fruit basket) to pick up the apples. Following the previous method in [16], we also utilize one rule-based heuristics to determine the effect point of the candidate tools. Next, the robot will require the LLMs to give the task plans and generate the corresponding skills for the task, which can be formulated as Loop{Grasp(one apple)$\rightarrow$ Hold(apple)$\rightarrow$ Place(apple) } $\Rightarrow$ Grasp(fruit_basket) $\Rightarrow$ Navigation(position). After completing the manipulation task, the robot re-perceives the scene graph information and updates the stored scene graph with all historical and current information graphs by employing rule merging. This process establishes a foundation for subsequent task executions.

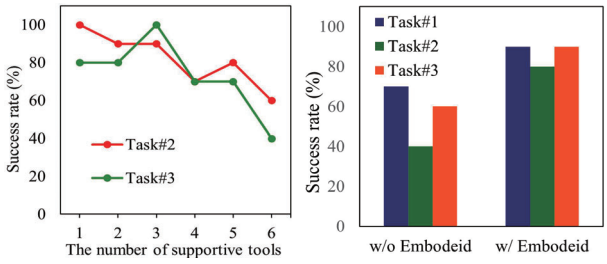



Fig. 4. (a) Results of embodied evaluation. (b) Results of embodied constraint judgement for LLMs.

## IV. Experiments

In this section, we empirically validate the framework proposed in this paper across three different difficulty tasks within a real-world environment.

### A. Experimental Settings

*a) Evaluation Metrics:* We assess the framework performance using three metrics: Success Rate (SR), indicating the proportion of successfully completed tasks out of the total attempted tasks; Time Consumption (TC), measured in seconds from the initiation of specific tasks to the completion of all tasks by the robot; and Execution Steps (ES), referring to the number of steps taken by the robot to accomplish tasks.

*b) Experimental Tasks:* We focus on one grasping task and pouring task, *i.e.* Grasping the apple (Task#1) represents a relatively simple task, whereas pouring the soy sauce bottle (Task#3, shown in Figure 3 (c)) poses a moderate level of difficulty. To further evaluate the effectiveness of our proposed framework, we introduced a more challenging task: grasping multiple apples (Task#2, shown in Figure 3(b)). In this task, we set four apples to be grasped at once. Figure 3 provides detailed descriptions of our experimental platform and task settings.

### B. Experimental Results

**Embodied ability.** Tool utilization is one of the most important manifestations of embodied intelligence. We explored three settings: without using tools (WUT), random using tools (RUT), and using specific tools (UST). Then we compared them with our proposed method under three tasks. Notably, we set ChatGPT [17] as the LLM. To ensure the validity of the results, we conducted ten repeated experiments on these tasks and reported the results in Table I. As can be observed from the Table I, our proposed framework obtains 100%, 90%, and 90% success rates under Task#1, Task#2, and Task#3, respectively. The experimental results indicate that while the robot

TABLE I

RESULTS OF TOOL UTILIZATION.

| Type | Task | SR | TC | ES |
|---|---|---|---|---|
| WUT | Task#1 | 100% | 26s | 4 |
| | Task#2 | 70% | 329s | 16 |
| | Task#3 | 0% | - | - |
| RUT | Task#1 | 90% | 38s | 5 |
| | Task#2 | 70% | 283s | 24 |
| | Task#3 | 60% | 155s | 15 |
| UST | Task#1 | 80% | 47s | 8 |
| | Task#2 | 80% | 252s | 17 |
| | Task#3 | 70% | 113s | 12 |
| Ours | Task#1 | 100% | 25s | 4 |
| | Task#2 | 90% | 232s | 15 |
| | Task#3 | 90% | 90s | 10 |

operates without tools, it achieves a relatively high success rate in simple tasks. Nevertheless, the success rate notably declines for complex tasks that entail lengthy execution steps. The random utilization of tools in task execution introduces additional steps in simpler tasks, such as Task#1, which diminishes accuracy and prolongs the process. Nevertheless, the embodied constraint judgment module in our framework can assist in determining whether to utilize the tool reasonably. These results demonstrate that our framework has the highest accuracy and the shortest time consumption and execution steps, indicating the effectiveness and efficiency of tool selection. We verify the effect of the number of supportive tools in the scene provided by the LLMs. The results are illustrated in Figure 4 (a). We can find that as the candidate number of tools increases, the success rate decreases. We speculate that it may be due to an increase in the number of similar tools available for selection, resulting in incorrect judgments in the LLMs. Figure 4 (b) shows the comparison results of using embodied constraint judgment module or not. As can be seen from the figure, all three tasks achieved performance gains of 20%, 40%, and 30% compared with the baseline method (*i.e.* without embodied constraint evaluation module), respectively, demonstrating the effectiveness of embodied constraint evaluation module.

**Scene graph updating.** Due to the possibility of robots utilizing tools available in the environment to execute tasks, object positions in the environment may shift after task execution. In light of this scenario, we chose to update the scene knowledge graph following each task execution. While a single task does not prompt changes in the scene graph, our experiments primarily focus on analyzing the data from subsequent task executions, particularly after successfully utilizing tools. The success rate, and

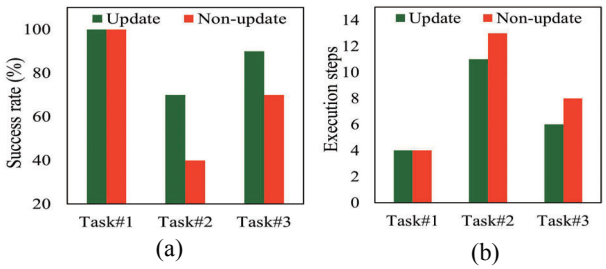



Fig. 5. (a) Results of success rate for scene graph updating. (b) Results of execution steps for scene graph updating.

execution step results are illustrated in Fig. 5 (a), and 5 (b), respectively.

The results indicate that, while updating the scene graph has minimal impact on simple tasks like Task#1, it significantly benefits the execution of complex tasks (Task#2, Task#3). For these complex tasks, updating the scene graph improves task success rates by 30% and 20% respectively. Moreover, the time consumption is reduced for successful tasks by 62 seconds and 22 seconds respectively. These findings underscore the significant benefits of scene graph updating for the execution of complex tasks. They facilitate robots in promptly identifying useful tools, thereby enhancing task success rates and reducing task completion time.

### C. Real-World Cases

To further reveal the effectiveness of our framework, we present the task execution process for the three tasks conditioned on our mobile manipulation robotic platform (shown in Figure 3(a)) in Figure 6. The robot first comprehends the human instruction and reaches the target manipulation space. As can be seen from Figure 6(a), the robot can rely on its own grasping skill to grasp one apple. However, if the task surpasses the robot's inherent abilities, such as those depicted in Figures 6(b) and (c), it must explore the utilization of tools for resolution. These results further demonstrate that our proposed framework can empower robots to make embodied evaluations to think of tasks before executing them.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a robotic embodied reasoning framework conditioned on the LLMs for complex tasks in our daily lives. The framework aims to empower robots to accurately assess their capabilities, allowing them to exploit supportive tools or rely on their inherent abilities to complete tasks. To reveal the effectiveness of our framework, we conducted extensive experiments across three different difficulty tasks within real-world scenarios.

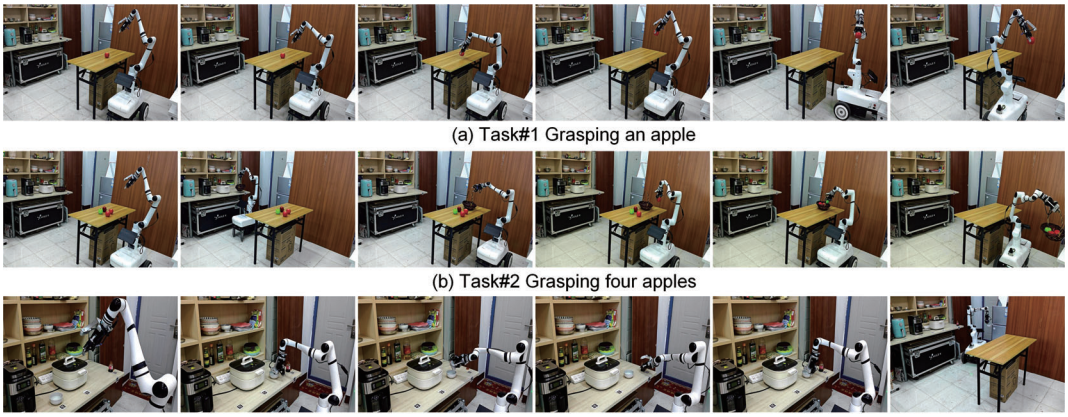



Fig. 6. Qualitative results of the execution process for the three tasks. Task#1 can be effectively accomplished without tools, while Task#3 can not be completed without tools (*i.e.* small bowl). Though Task#2 can be done by delivering apples one by one without tools, the efficiency is relatively low. Nevertheless, our framework enables the robot to explore supportive tools (*i.e.* fruit basket) to accomplish the task and improve the task execution efficiency.

Importantly, the proposed framework's performance meets task requirements even after replacing various LLMs, confirming its robustness and applicability.

In future research, we will incorporate larger models to streamline data and time requirements for robot planning and training processes.

## REFERENCES

[1] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[2] M. Xu, P. Huang, W. Yu, S. Liu, X. Zhang, Y. Niu, T. Zhang, F. Xia, J. Tan, and D. Zhao, "Creative robot tool use with large language models," *arXiv preprint arXiv:2310.13065*, 2023.

[3] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held, "Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds," in *Conference on Robot Learning*. PMLR, 2023, pp. 1038–1049.

[4] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[5] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[6] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley *et al.*, "Robots that ask for help: Uncertainty alignment for large language model planners," in *Conference on Robot Learning*. PMLR, 2023, pp. 661–682.

[7] J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, H. Mao, Z. Li, X. Zeng, R. Zhao *et al.*, "Tptu: Task planning and tool usage of large language model-based ai agents," in *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[8] B. Li, P. Wu, P. Abbeel, and J. Malik, "Interactive task planning with language models," in *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[9] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. Chai, "Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent," *arXiv preprint arXiv:2309.12311*, 2023.

[10] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *arXiv preprint arXiv:2303.06247*, 2023.

[11] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, pp. 1–40, 2023.

[12] H. Huang, O. Zheng, D. Wang, J. Yin, Z. Wang, S. Ding, H. Yin, C. Xu, R. Yang, Q. Zheng *et al.*, "Chatgpt for shaping the future of dentistry: the potential of multi-modal large language model," *International Journal of Oral Science*, vol. 15, p. 29, 2023.

[13] D. Shah, B. Osiński, S. Levine *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*. PMLR, 2023, pp. 492–504.

[14] R. Chandra, R. Maligi, A. Anantula, and J. Biswas, "Socialmapf: Optimal and efficient multi-agent path finding with strategic agents for social navigation," *IEEE Robotics and Automation Letters*, 2023.

[15] J. Liu, J. Xie, S. Huang, C. Wang, and F. Zhou, "Continual learning for robotic grasping detection with knowledge transferring," *IEEE Transactions on Industrial Electronics*, 2023.

[16] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," *IEEE Robotics and Automation Letters*, 2023.

[17] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, no. 5, pp. 1122–1136, 2023.