

Robot Manipulation Planning with Large Pre-Trained Models of Language, Vision, and Action

Yongsung Son
Digital Convergence Research Laboratory
Electronics and Telecommunications
Research Institute
Daejeon, Korea
ysson@etri.re.kr

Hyonyong Han
Digital Convergence Research Laboratory
Electronics and Telecommunications
Research Institute
Daejeon, Korea
hyoyoung.han@etri.re.kr

Junhee Park
Digital Convergence Research Laboratory
Electronics and Telecommunications
Research Institute
Daejeon, Korea
juni@etri.re.kr

Abstract— This paper explores the integration of large pre-trained models of language, vision, and action to enhance robot manipulation planning. By leveraging advanced language models such as GPT-4 and Gemini, task planning can be articulated in natural language, allowing for intuitive and precise task specifications. The paper describes task and motion planning (TAMP), which is crucial for robot operation, optimizing precise execution by considering the work environment and linking high-level decision-making with detailed motion control.

Keywords—robotic control, motion planning, pre-trained model, prompt engineering, LLM

I. INTRODUCTION

Recently, with the development of ultra-large generative AI models, the development of robotic artificial intelligence technology that enables autonomous performance of complex tasks in robot operation in a manufacturing environment is a global technology trend. To realize this, technology research is underway to recognize the work environment, understand tasks, and create operation procedures [1]. Prompt engineering methods to generate robot motion using Large Language Models (LLMs) was studied [2,3,4,5]. In order to plan and control more efficient robot movements based on this task generation method, various frameworks and optimization techniques are being studied [6,7,8,9]. Research was also conducted to provide the safety and predictability necessary for robots and people to collaborate through a robot task creation platform [10], and benchmarks to evaluate the performance of the robot task creation system were also reported [11]. In this paper, we are designing task generation for robot arm manipulation for assembling products.

II. PROMPT ENGINEERING FOR TASK PLANNING

In our approach, we use an LLM (GPT4) trained on a text corpus, enabling it to provide accurate function predictions based on specific instructions. We integrate perceived target position information and motion functions from the LLM into a code template, allowing the robot to execute tasks effectively. Task execution is managed hierarchically: long-horizon tasks (over 10 motion functions) are considered first-layer tasks and are broken down into short-horizon tasks by the LLM. Short-horizon tasks (fewer than 10 motion functions) are treated as second-layer tasks, with the LLM directly returning the necessary functions to accomplish them. Fig.1 outlines a robotic assembly system that uses a camera to capture images, a Vision-Language Model (VLM) to analyze and identify objects, a task generator to create a task script, a task verifier to ensure feasibility, and a robot to execute the assembly tasks based on the verified script, integrating vision-based recognition with language model-driven planning and execution.

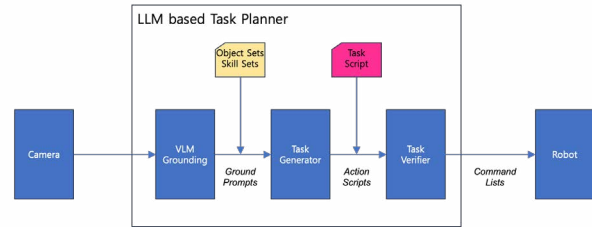


Fig.1 LLM-based task planning with Robot Manipulation

III. SYSTEM ARCHITECTURE

A. Robotic Arm Assembling Product System

The core components of this robotic assembly system include a robotic arm equipped with a gripper, a vision system utilizing a camera for work environment recognition, and a control system for task execution. The operating procedure involves the camera capturing images of the workspace to identify the parts to be assembled. The vision system processes these images to determine the positions and orientations of the parts. The control system then directs the robotic arm to pick up the parts in the correct sequence and assemble them according to the predefined work order, ensuring precision and efficiency in the assembly process.

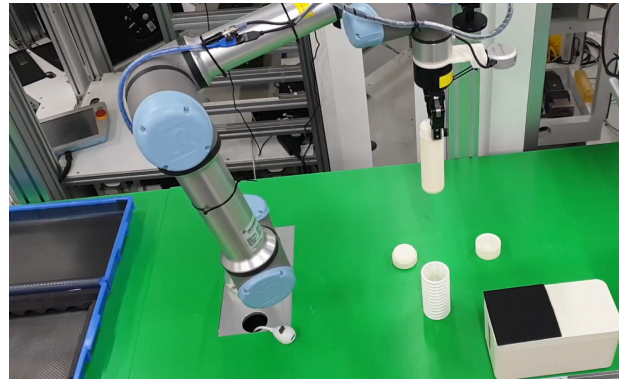


Fig.2 Robotic Arm Assembling Product System

B. LLM based Task Planner

In Fig.3, the LLM-based Task Planner operates through a series of systematic steps to achieve robotic task generation and execution. The process begins with image capture and recognition, where a camera captures images of the work environment, focusing on a table where all objects are placed. Using YOLO and GPT-4, the system performs image recognition to determine the current state from the visual

information. This visual information is then converted into a text script, with the positions of the objects and the hand identified through joint values. Due to the probability of recognition error in the image recognition method, a Ground Prompt is generated by reorganizing the observation results into the image recognition results of GPT-4 and the list of objects recognized by YOLO. This Ground Prompt ensures a more accurate and reliable understanding of the work environment, combining the strengths of both GPT-4 and YOLO to mitigate errors and improve task planning and execution.

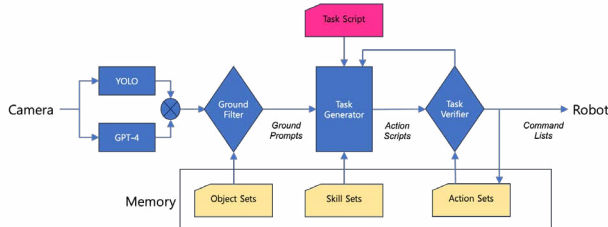


Fig.3 LLM based Task Planner

Ground Prompts are generated for all text scripts recognized by the vision recognition module. These prompts describe the current state and the goal task. The task generator then creates an action script by referring to the relevant skill set corresponding to the goal task, outlining the specific steps needed to accomplish the task.

Next, the task verifier checks the generated action script to ensure it is appropriate and feasible. This involves validating the sequence of actions and their suitability for the task. Once verified, the action script is translated into robot manipulation commands, which are sent to the robot to control its movements and perform the task.

Throughout this process, the system stores the object set for grounding, the robot's work skill set, and the action sets performed so far. This stored information is used to enhance the efficiency of future task generation, making the system more effective over time. This procedural flow ensures that the robotic system can autonomously recognize, plan, and execute tasks with high efficiency and accuracy.



Fig.4 Test Input Image for Visual Grounding

IV. SYSTEM EVALUATION

We built a prototype system to deploy task planning generation for assembling various objects using a robotic arm. Our experimental setup comprises a UR5 robot arm, a RealSense L515 camera mounted on the robot's wrist, and a Robotiq 2F-85 gripper. The assembly model consists of three distinct part types: the cap, shaft, and spring. We conducted tests on 50 different scenarios by altering the position and orientation of the objects. To evaluate our system, we prototyped our architecture and conducted extensive testing.

An example of a Ground Prompt is as follows. Fig. 4 explains the results of observation through VLM Grounding with a script. Objects recognized in an image contain relevant information. The Ground Prompt is completed by combining this script with information such as the location and order of objects recognized through YOLO. However, there is a problem with incorrect recognition information in the middle of the script, such as identifying a white portable hard drive and a black smartphone incorrectly. To address this, the system includes a procedure to verify the Ground Prompt using a ground filter to remove inaccuracies, relying on Object Sets stored in memory.

This picture shows a well-used office workspace. Here are the items visible in the image:

- A desk with a clutter of various office items.*
- A desktop computer monitor with sticky notes attached to the bottom bezel.*
- A traditional landline telephone to the left of the monitor.*
- A white mug, possibly for coffee or tea, near the telephone.*
- A wireless keyboard with a built-in trackpad on the right side and a wrist rest at the bottom.*
- A wireless mouse to the left of the keyboard on a mouse pad.*
- A black mouse pad with a wooden accent where the wrist rest is, and it also has a matching wooden stylus holder labeled "Logitech".*
- Various cables and chargers on the desk.*
- A pair of eyeglasses next to a black case to the right of the monitor.*
- A white portable hard drive.*
- A bottle of hand sanitizer behind the mug.*
- A black smartphone to the right side of the desk.*
- Assorted pens and markers in a clear holder behind the monitor.*
- Other miscellaneous items like a pack of tissues, a key ring, and more sticky notes.*

The Task Generator receives a Goal Task Script as input and generates Action Scripts from Ground Prompts. The following is an example of created Action Scripts. The unit actions of Action Scripts are created based on the Skill Set in memory. The Skill Set specifies robot operation procedures. The example below shows only a portion of the entire Action Script. The created Action Scripts are checked in the Task Verifier to see if they satisfy the Goal Task. If they are insufficient, the system goes through a repetitive loop to create additional Action Scripts.

Testing Goal Task Script

- *I am a robot with two hand arms. I want to clean this desk. Would you show me task sequences to cleaning this desk?*

Certainly! Here is a task sequence for cleaning and organizing the desk, broken down into steps. Since you have two hand arms, you should be able to grasp and manipulate objects similarly to a human:

1. Remove Clutter:

- ① *Pick up any trash, such as empty snack packets or used tissues, and dispose of them in a waste bin.*
- ② *Gather any dishes, like the mug, and set them aside to be washed or placed in a dishwasher.*

2. Organize Papers:

- ① *Collect all loose papers and sticky notes.*
- ② *Sort through the papers, deciding what to keep and what to recycle.*
- ③ *File important papers in a desk drawer or a filing cabinet and properly dispose of the rest.*

3. Sort Miscellaneous Items:

- ① *Place personal items like the key ring into a desk drawer for safekeeping.*
- ② *Ensure that the glasses are placed in their case to prevent damage.*

V. LIMITATIONS AND FUTURE WORKS

The prototype implemented to design this system and verify its functionality revealed several limitations:

1. **Recognition Accuracy:** Despite using advanced models like GPT-4 and YOLO, the system still faces challenges with recognition accuracy. Misidentification of objects or incorrect interpretation of visual data can lead to errors in task execution.
2. **Real-Time Processing:** The prototype may struggle with real-time processing, especially when handling complex tasks or large amounts of visual data. This can result in delays or reduced responsiveness in dynamic environments.
3. **Hardware Dependence:** The performance of the system is highly dependent on the quality and specifications of the hardware used, including cameras and robotic arms. Limited hardware capabilities can restrict the overall effectiveness and precision of the system.
4. **Environmental Variability:** The prototype may not perform consistently across different work environments. Variations in lighting, object appearance, or workspace layout can affect the accuracy of image recognition and task execution.
5. **Integration Complexity:** Integrating various components, such as the vision system, task generator, and robot control, can be complex and prone to synchronization issues. Ensuring seamless communication and coordination among these components is a significant challenge.
6. **Scalability:** The current prototype might have limitations in scaling up for more complex or larger-scale assembly tasks. Adapting the system to handle a broader range of tasks or more intricate assembly processes requires further development and optimization.
7. **Learning and Adaptation:** While the system can generate tasks based on predefined skills and object sets, its ability to learn and adapt to new tasks or unexpected situations is

limited. Enhancing its learning capabilities and adaptability is crucial for more versatile applications.

8. **Safety and Reliability:** Ensuring the safety and reliability of the system, especially when interacting with humans or handling delicate tasks, remains a critical concern. The prototype needs robust safety mechanisms to prevent accidents and ensure dependable operation.

Furthermore, our future research will focus on addressing the challenges of assembling real objects and tasks in diverse environments. We aim to improve system performance by enhancing the inference and generation of task planning scripts and action scripts during real-world deployment of our approach.

VI. CONCLUSION

In this paper, we propose a system for LLM-based robotic manipulation tasks using large models to generate robot control sequences that achieve a goal task. This system highlights the potential of LLMs to simultaneously handle both planning and motion control by utilizing object sets, skill sets, and action sets stored in memory. In the future, we will evaluate all functionalities of LLMs and execute robotic manipulation in real-time.

ACKNOWLEDGMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [24ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways] and the Technology Innovation Program [24PR2110, Technology convergence of digital twin, artificial intelligence and robotics for AI-Robot based autonomous manufacturing in unstructured environments] funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea)“.

REFERENCES

- [1] Dhruv Shah, et al., “LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action,” arXiv preprint, arXiv:1909.12271, 2019.
- [2] Takahide Yoshida, et al., “From Text to Motion: Grounding GPT-4 in a Humanoid Robot “ALTER3”, ” arXiv preprint, arXiv:2312.06571v1, Dec, 2023.
- [3] Anonymous authors., “LLM+A: Grounding Large Language Models in Physical World with Affordance Prompting,” under review of ICLR 2024.
- [4] Yan Ding, et al, “Task and Motion Planning with Large Language Models for Object Rearrangement,” IROS 2023
- [5] Guangran Chen, et al., “Empowering Large Language Models on Robotic Manipulation with Affordance Prompting,” arXiv preprint, arXiv:2404.11027v1, April, 2024.
- [6] Lirui Wang, et al., “Gensim: Generating Robotic Simulation Task via Large Language Models,” ICLR 2024
- [7] Ishika Singh, et al., “ProgPrompt: Generating Situated Robot Task Plans using Large Language Models,” ICRA 2023.
- [8] Murtaza Dalal, et al., “Plan-Seq-Learn : Language Model Guided RL for Solving Long Horizon Robotics Tasks,” ICLR 2024
- [9] Antonio Sabbatella, et al., “Prompt Optimization in Large Language Models,” Mathematics, 2024.
- [10] Haokun Liu, et al., “LLM-Based Human-Robot Collaboration Framework for Manipulation Tasks,” arXiv preprint, arXiv:2207.04429v2, Aug, 2023.
- [11] Jae-Woo Choi, et al, “LoTa-Bench: Benchmarking Language-oriented Task Planners for Embodied Agents,” ICLR 2024