Multi-modal LLM-enabled Long-horizon Skill Learning for Robotic Manipulation

Runjia Tan
School of Mechanical and
Aerospace Engineering
Nanyang Technological University Singapore
runjia.tan@ntu.edu.sg

Shanhe Lou
School of Mechanical and
Aerospace Engineering
Nanyang Technological University Singapore
shanhe.lou@ntu.edu.sg

Yanxin Zhou
School of Mechanical and
Aerospace Engineering
Nanyang Technological University Singapore
yanxin.zhou@ntu.edu.sg

Chen Lv*
School of Mechanical and
Aerospace Engineering
Nanyang Technological University Singapore
lyuchen@ntu.edu.sg

***Abstract*—The advent of Large Language Models (LLMs) has empowered robots to execute tasks based on human instructions. Nonetheless, the challenge still persists in endowing robots with the capability to learn from the progress during interacting with human, which blocks the application of robots on human-like assistant. In response, this paper proposes a LLMs-based framework aimed at facilitating robots in acquiring new skills through interaction history. The proposed framework comprises three integral components: 1) a subsystem named the Fast Learner, consisting of three functionally distinct GPT-based modules. These modules are adept at decomposing long-horizon instructions into a sequence of manageable tasks while synthesizing skill schemas from historical interactions. 2) a hierarchical skill library categorizing tasks based on complexity, alongside a collection of meta-tasks that encompass all other tasks. 3) a scene understanding module which identifies regions of interest and generates relationship graphs based on visual input combined with textual prompts. Comprehensive evaluation of the system is conducted in both simulated environments and with real robots, demonstrating its exceptional proficiency in long-horizon task decomposition and effectiveness in acquiring and mastering new skills.**

I. Introduction

The remarkable comprehension and reasoning performance of Large Language Models (LLMs), such as GPT4 [1] and Llama2 [2] in conversational contexts, have catalyzed significant scholarly attention towards equipping robots with similar cognitive capabilities. Such advancements enable human operators to directly instruct robots to accomplish tasks, thereby alleviating the need for meticulous programming [3] [4]. Nonetheless, the categories of task encountered in daily life are vast and varied, surpassing the coverage of LLMs' pre-existing datasets. Consequently, there arises a urgent need for methodologies leveraging LLMs' reasoning capabilities and commonsense knowledge storage to facilitate the acquisition of new skills by robots, thus addressing this pervasive challenge.

In this paper, we introduce a novel LLMs-based approach aimed at enabling robots to continually acquire new manipulation skills from human-robot interaction history. The proposed system comprises a central module, the Fast Learner, complemented by two submodules, namely the Skill Library and the Scene Understanding module, as illustrated in Fig. 1. Inspired by hierarchical policy learning [5] [6], we hypothesis that complex tasks can be deconstructed into constituent subtasks. Concretely, we design Fast Learner to invoke GPT API to discern human intention and decompose long-horizon task to a sequence of pre-learned tasks. Accordingly, we define tasks resistant to further decomposition as meta-tasks, which can be defined and trained before leaning other skills. It signifies that any novel tasks can be articulated as sequences of a set of meta-tasks. To elucidate the interplay among tasks, we have devised a tree-like skill library to accommodate these skills, whereby tasks are systematically arranged in hierarchical levels. This organizational paradigm lays the groundwork for fostering the autonomous evolution of robotic capabilities over time. Besides, we add a scene understanding module based on GPT4-V to empower our system capability to generalize learned skills to new environment.

In comparison to the existing studies, this work aims to make four distinctive contributions:

- A LLMs-based system designed to facilitate robot to systematically acquire new manipulation skills from human-robot interaction history.
- Establishment of a hierarchical skill library, which organizes tasks according to their complexity and stores relationship among stored skills.
- A scene understanding module to transfer text-vision prompts to relationship graph and highlight objects of interest.
- Experimentation coducted on the simulation platform exceeded R3M [7] on all five tasks and deployed the learned skills directly to the physical robot seamlessly.

This study is supported under the RIE2020 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

II. Related Works

A. Task decomposition to meta group

The concept of decomposing complex tasks into manageable components, often termed as meta-tasks, has been a central theme in the field of robotics and artificial intelligence. This approach acknowledges the inherent complexity of real-world tasks and seeks to break them down into smaller, more comprehensible units. Sutton et al. [8] proposed the idea of hierarchical reinforcement learning, wherein tasks are organized hierarchically, with higher-level tasks composed of lower-level subtasks. Ikeuchi et al. [9] proposed Semantic Task Group (STG) in addition to Physical Task Group (PTG) and classified rigid object manipulations performed by humans. This represents when the set of human action is defined, the set of robot will be also settle down. Singh and Satinder [10] further explored the decomposition of tasks into meta-tasks in the context of robotic manipulation. They proposed a method for automatically generating meta-tasks based on task demonstrations, enabling robots to learn complex manipulation skills from human demonstrations. MetaWorld [11] proposes a set of 50 actions for evaluation and aims to acquire new skills much more quickly by leveraging prior experience to learn how to learn.

B. Incrementally Skill Learning

Robotic skill learning, the capability of robots to incrementally acquire and adapt knowledge incrementally over time, has garnered significant attention in recent years due to its crucial role in enabling robots to operate effectively in dynamic and changing environments. Timothée et al. [12] introduced the concept of continual learning in robotics, emphasizing the importance of enabling robots to learn from past experiences and adapt their behavior accordingly. Finn et al. [13] explored the application of meta-learning techniques to robotic continual learning. They proposed meta-learning algorithms that enable robots to rapidly acquire new skills from limited experience, thereby facilitating continual learning in resource-constrained environments. The incremental learning process in [14] can decrease the time paid on repeating work. However, it is only tested in the ideal virtual game environment. [3] proposed to use ChatGPT to decompose multi-step human instructions into robot actions, but it will cost work labour to guide robot hand in hand.

III. Method

We commence by presenting a comprehensive statement of the problem our system addresses in Section III-A. Subsequently, we elaborate on our approach to imbuing the robot with the ability to memorize the instructed skills in Section III-B. Finally, Section III-C delves into the framework of the scene understanding module, elucidating how it furnishes the robot with pertinent information necessary for the application of learned skills within the current scene context.

A. Problem Statement

The primary challenges our system aims to address are twofold: firstly, it can adeptly generate task sequences based on the current scene layout, human-provided prompts, and an expanding repository of learned skills, all while mitigating the risk of hallucination [15]. Secondly, the system must possess the capability to acquire new skills from interaction history and systematically organize them using a specialized data structure.

During the task sequence generation process, the system integrates inputs from multiple sources: human instructions denoted as L, visual observations captured by the camera denoted as o and skills s retrieved from the skill library S. Subsequently, the system outputs a sequence of tasks denoted as $\Gamma = \left\{s^0, s^1, ..., s^T\right\}$. Specifically, L encapsulates textual cues indicative of human intent, while s represents the skill's descriptions of functionality and associated arguments, drawn from the skill library denoted as $S = (s_0, s_1, ..., s_n)$.

Regarding skill acquisition, we define the interaction history as $H = \left\{L^0, s^0, L^1, s^1, ..., L^T, s^T\right\}$. The system then synthesizes this interaction history and generates a succinct summary, resulting in the output of a new skill description, denoted as s_{n+1}, which is seamlessly incorporated into the skill library S for subsequent iterations.

B. Learn skills from human instructions

The Fast Learner encompasses three primary modules along with an auxiliary module, positioned at the nucleus of the system architecture as illustrated in Fig. 1. Fundamental to the acquisition of new skills is the ability of the robot to execute corresponding tasks guided by human instructions, facilitated by the interplay of two pivotal modules: the code generator, also referred to as the 'Decider', and the code refiner, also named 'Critic'. The Decider module is tasked with decomposing long-term complex tasks into shorter, more manageable tasks, guided by the task set available in the skill library. Mathematically, this process can be represented as:

$$\Gamma = D(L|S) \tag{1}$$

the D denotes the Decider. It means when we give a snippet of prompt including skill descriptions, the LLMs-based Decider can output an abstract representation of a sequence of skills conditioned on input text. The code parser, the auxiliary module after Decider, facilitates the conversion of abstract task sequences into executable code, ensuring compatibility across various programming languages to accommodate algorithms developed in different linguistic paradigms. Complementing the Decider module is Critic, which assumes a dual role: firstly, to scrutinize the code generated by the Decider for correctness, and secondly, to modify the code based on human feedback.

Subsequently, the history acquired from the Decider and the Critic modules is leveraged by another integral module, the function maker or 'Learner', to distill context information and synthesize novel skills, seamlessly integrating them into the skill library. This process can be encapsulated as:

$$s_{n+1} = C(H) \tag{2}$$

The skills within the system are structured in a JSON-meta format, a format conducive to the training and fine-tuning of GPT, enhancing the system's understanding of skills.

The skill library is conceived as a multi-root growing tree, organizing mastered skills by complexity level, as depicted in the top-right corner of the system overview in Fig. 1. Inspred by PTG task group [16], we posit that all long-term complex tasks can be decomposed into short-term simple tasks, with

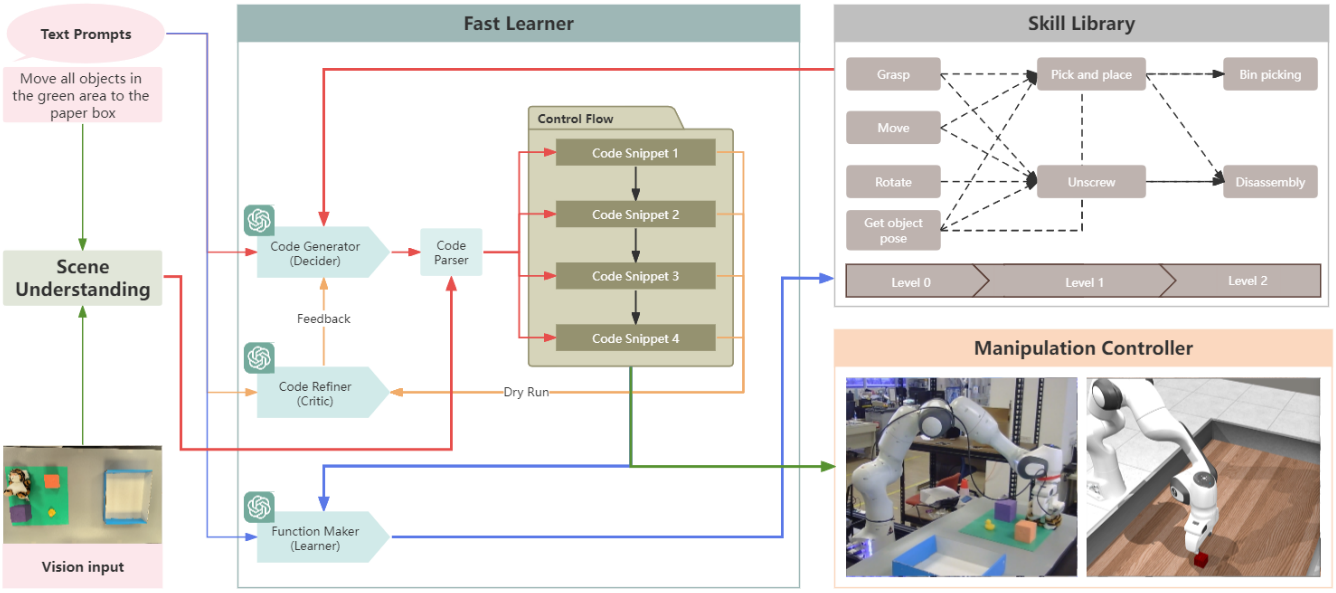
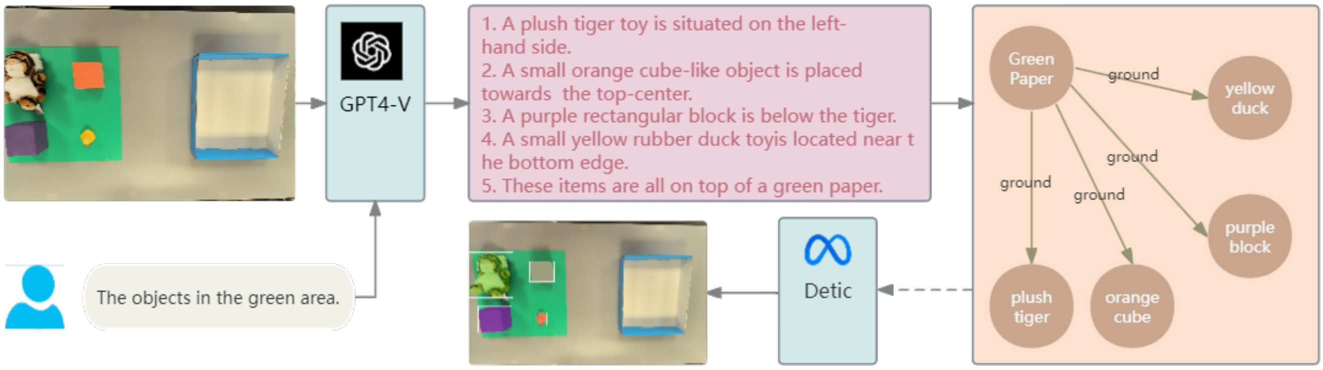


Fig. 1. Overall Framework. Given text-vision prompts, the Fast Leaner can generate codes to control manipulator with the assistance of the Scene Understanding module and Skill Library. Additionally, Fast Learner can acquire new skills from its context and history and store them in the Skill Library.



1. A plush tiger toy is situated on the left-hand side.
2. A small orange cube-like object is placed towards the top-center.
3. A purple rectangular block is below the tiger.
4. A small yellow rubber duck toyis located near t he bottom edge.
5. These items are all on top of a green paper.

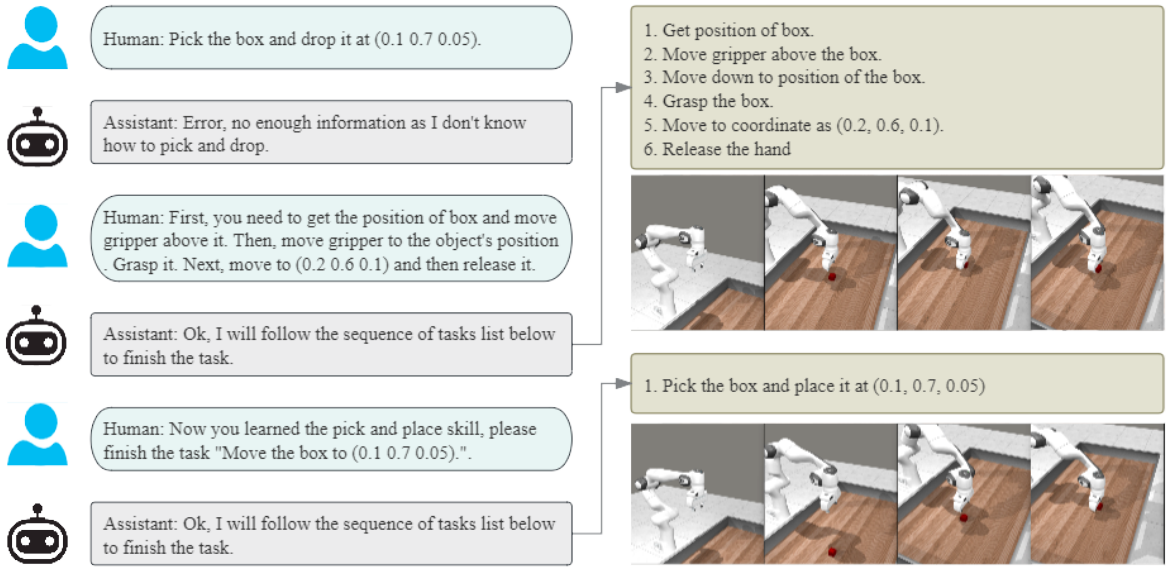
Fig. 2. Scene understanding module. The integration of prompts with scene images serves as input for the GPT4-V model, facilitating the generation of descriptive layouts. These layouts subsequently undergo refinement to enhance the graph representation depicting the relationships among objects within the scene. Concurrently, the categories pertaining to objects of interest are fed into the Detic module, facilitating precise localization of these objects.

tasks resistant to further decomposition designated as meta-tasks or meta-skills, preordained and cataloged at level 0 within our skill library. Therefore, we can note the index of skill as τ_{ij} and the i is the level number while j represents jth skill in ith level. Although we store our skill in added time, the actual skill can be got by using $s_{\tau_{ij}}$. When initialized, only nodes representing meta tasks exist, which are marked as level 0. When new skill is added into the library, we will get the its index $n+1$ and relationship R_{n+1}, which stores its children nodes. The level of the new node will be calcuated as $max(R_{n+1})+1$. For example, one new skill comprises of a level 0 skill and a level 1 skill and its skill is $max(0,1)+1=2$. In our system, the meta skills can be changed easily so that it can keep the best performance while equiping the SOTA algorithm such as screwing skill from reinforcement learning trained in nut-and-bolt assemblies scene [17].

Upon receiving instructions from a human operator, the robot initiates a process of parsing and decomposing the instruction into a task sequence. Subsequently, it searches for related skills from the highest level, marking tasks feasible with skills available at the current level. This incremental search process confers two-fold advantages: conservation of computational resources, notably GPT4 tokens, leading to cost-effectiveness, and mitigation of hallucination risks arising from an overload of input descriptions. This iterative process continues until tasks at level 0, i.e., meta-tasks, are exhaustively explored. If the task is deemed achievable, the robot generates a corresponding task sequence incorporating learned skills; otherwise, it solicits further information or clarifications from the human operator.

C. *Understand relationship of objects in the scene*

Although LLMs have a strong ability to handle with text-sequence task, it cannot make great decision while facing lack of information from environment. Therefore, it need external



Human: Pick the box and drop it at (0.1 0.7 0.05).

Assistant: Error, no enough information as I don't know how to pick and drop.

Human: First, you need to get the position of box and move gripper above it. Then, move gripper to the object's position . Grasp it. Next, move to (0.2 0.6 0.1) and then release it.

Assistant: Ok, I will follow the sequence of tasks list below to finish the task.

1. Get position of box.
2. Move gripper above the box.
3. Move down to position of the box.
4. Grasp the box.
5. Move to coordinate as (0.2, 0.6, 0.1).
6. Release the hand

Human: Now you learned the pick and place skill, please finish the task "Move the box to (0.1 0.7 0.05).".

Assistant: Ok, I will follow the sequence of tasks list below to finish the task.

1. Pick the box and place it at (0.1, 0.7, 0.05)

Fig. 3. Experiments on customized Metaworld environment. The first interaction loop shows *pick and place* task is not included in the meta skill set when initialized. Then human gives detail instructions to drive robot arm to finish *pick and place* task based on meta skills. After that, agent can understand *pick and place* order directly and generalize to other targets.

components those can perceive surroudnings to help detect targets or parse object relationship in the scene that we mainly focus on.

Firstly, the RGBD image, acquired through the depth camera, undergoes segmentation into distinct color and depth components. Subsequently, the color components are inputted into the GPT4-V interface along with corresponding prompts. The GPT4-V interface processes this input and produces semantic labels pertaining to the desired target objects, along with providing insights into the object relationships within the current workspace. These semantic labels are subsequently relayed to the an Open-vocabulary detector, Detic [18] module for further processing and segmentation. The resulting segmented mask is then overlaid onto the depth image, enabling the computation of precise positional information regarding the target object.

IV. Experiments

We tested our system to verify its capability to 1) decompose human instructions and parse them to sequence of tasks and summarize and learn new skill from interaction history, 2) depict objects relationship and mark objects of interest in current scene, 3) apply skills pre-learned in simulator to physical setup without modification. Besides, we make a performance comparison with R3M [7] on five different tasks and outperform them across the board.

A. Learn new skill from meta task

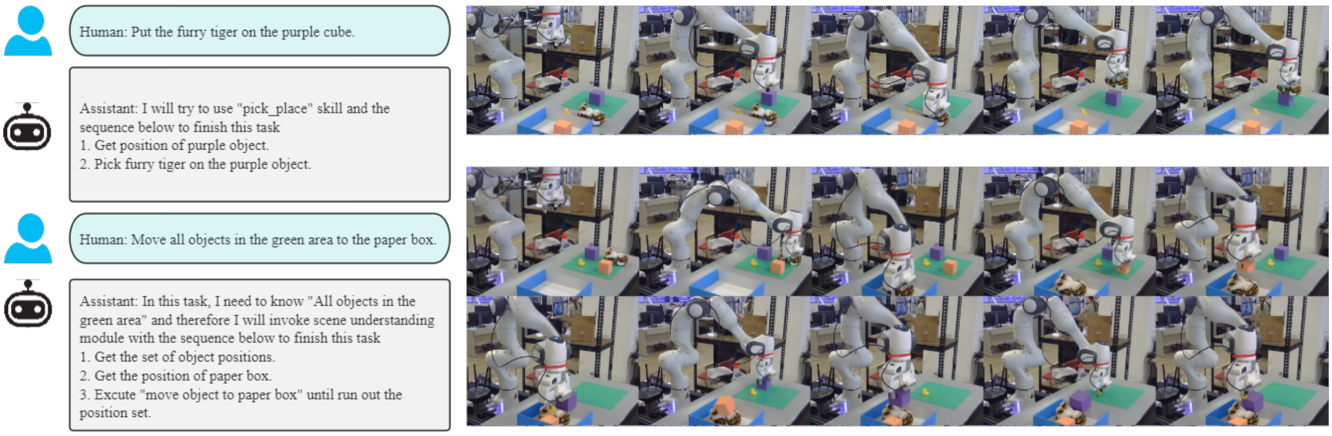
To illustrate the capability to decompose human instructions and parse them to sequence of tasks and summarize and learn new skill from interaction history, we presented our system with a task for moving a cube to target place. To evaluate the performance of system on this task, we code a new wrapper of MetaWorld [11], which is a Mujoco [19] based multi-task envrinvment, and change the predefined manipualtor from Sawyer to Franka Emika robot arm to keep consistency between simulator and physical setup. As for three components of Fast Learner, that is, Decider, Critic and Learner, GPT model with gpt-4-1106-preview version is used as backbone to ensure their best reasoning and comprehension abilities.

To elucidate the process by which our system acquires new skills, we arrange a simulation based experiment focused on completing a *pick and place* task, as depicted in Fig. 3. Initially, upon system initialization, the skill library contains solely meta-tasks. Therefore, when we order robot with ”grab a cube to <position>”, the Decider module indicates its inability to fulfill the task due to the absence of relevant skills. Subsequently, we provide detailed instructions to guide robot's task execution. The Decider parses the instructions into sequence of tasks and corresponding codes. However, the generated codes may not always run successfully due to erroneous instructions or the inherent limitations of AI robustness.

In such instances, the Critic module endeavors to rectify errors during a dry-run attempt. If unsuccessful, it will send feedback to operator and ask for help. Fig. 3 shows the failed case when input instruction mismatchs the skill library storage. After running tasks successfully, interaction process will be stored into history and can help agent understand more. The learner can be used with the history to summarize a new skill. Once a new skill is mastered, repeated teaching becomes unnecessary. For instance, we show that human can just order the robot with the text we failed before successfully and even combine the new skill with meta skill to finish one task.

B. Scene understanding with prompts

To ascertain the efficacy of our system, we devised an additional task building upon the preceding task of object



Human: Put the furry tiger on the purple cube.

Assistant: I will try to use "pick_place" skill and the sequence below to finish this task
1. Get position of purple object.
2. Pick furry tiger on the purple object.

Human: Move all objects in the green area to the paper box.

Assistant: In this task, I need to know "All objects in the green area" and therefore I will invoke scene understanding module with the sequence below to finish this task
1. Get the set of object positions.
2. Get the position of paper box.
3. Excute "move object to paper box" until run out the position set.

Fig. 4. Experiments on physical workspace. The video sequence above shows that robot can transfer skills learner in simulator to physical workspace directly and recognize the spatial relation such as *on*. The clips below presents agent can mark objects of interest conditioned on human text prompts and execute task on them.

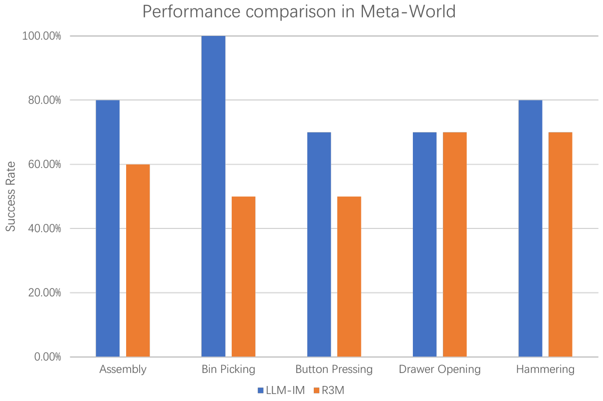


Fig. 5. Success rate comparison between R3M and our system.

manipulation, specifically involving the grabing of objects to designated locations. To address the challenge of handling multiple targets concurrently, we incorporated a prompt-based scene understanding module, the details of which has been illustrated in Section III-C. Within the scene understanding module, we present a preliminary result depicting an overview of our workspace. As illustrated in Fig. 2, our system effectively employs the GPT4-V interface to identify objects of interest implied by the prompts, while the open-vocabulary detector employs these identified categories to facilitate segmentation of the scene.

C. Real Robot test and performance comparison

Finally, to assess the transferability of learned skills from the virtual environment to the physical setup without requiring modifications, we tasked the robot with executing previously acquired skills, as depicted in Fig. 4. In this setup, we employed the Realsense D435 perception sensor to capture RGB and depth information from the environment.

In the initial test scenario, we positioned a furry tiger and a purple cube as manipulable objects, alongside an orange box as an obstacle. The robot was instructed to ”place the furry tiger onto the purple cube,” thereby evaluating its capacity to seamlessly apply skills learned in the simulator to physical robot operations and its proficiency in understanding spatial relationships.Subsequently, another experiment was conducted, wherein four objects were arranged on a green canvas, with the agent instructed to ”move all objects within the green area into the paper box.” This test aimed to assess the robot’s comprehension of object interactions and its ability to execute a single skill multiple times within a single task.

To assess the generalization capabilities of our system across diverse tasks, we conducted a comparative analysis with R3M, focusing on five tasks from the Metaworld benchmark: assembly, bin picking, button pressing, drawer opening, and hammering. Each task was executed 10 times to ensure robust evaluation. As illustrated in Fig. 5, our system consistently outperforms R3M across all five tasks, thereby underscoring its efficacy and effectiveness in task execution and generalization.

V. Conclusion

In summary, our study presents a novel LLMs-based framework aimed at facilitating robots in acquiring new skills through interaction history. Through the integration of three integral components - the Fast Learner, the Skill Library, and the scene understanding module - the framework demonstrates remarkable proficiency in facilitating the acquisition and mastery of new skills and achieve higher success rate compared with R3M.

Looking ahead, future research should focus on expanding the evaluation of our system across diverse environments to enrich the meta-task set and enhance real-world applicability. Additionally, integrating human action recognition could address challenges associated with speech instructions, improving the system’s usability.

References

[1] OpenAI *et al.*, “Gpt-4 technical report,” 2023.
[2] H. Touvron *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” 2023.
[3] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, “Chatgpt empowered long-step robot control in various environments: A case application,” *IEEE Access*, vol. 11, p. 95060–95078, 2023. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2023.3310935

[4] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," 2023.

[5] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," 2019.

[6] 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," 2023.

[7] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," 2022.

[8] S. P. Singh, "Reinforcement learning with a hierarchy of abstract models," in *Proceedings of the National Conference on Artificial Intelligence*, no. 10. Citeseer, 1992, p. 202.

[9] K. Ikeuchi, N. Wake, R. Arakawa, K. Sasabuchi, and J. Takamatsu, "Semantic constraints to represent common sense required in household actions for multi-modal learning-from-observation robot," 2021.

[10] Z. Lin, Y. Chen, and Z. Liu, "Sketch rl: Interactive sketch generation for long-horizon tasks via vision-based skill predictor," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 867–874, 2023.

[11] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019. [Online]. Available: https://arxiv.org/abs/1910.10897

[12] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," 2019.

[13] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," 2017.

[14] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv: Arxiv-2305.16291*, 2023.

[15] C. Cui, Y. Zhou, X. Yang, S. Wu, L. Zhang, J. Zou, and H. Yao, "Holistic analysis of hallucination in gpt-4v(ision): Bias and interference challenges," 2023.

[16] I. Yanaokura, N. Wake, K. Sasabuchi, R. Arakawa, K. Okada, J. Takamatsu, M. Inaba, and K. Ikeuchi, "A multimodal learning-from-observation towards all-at-once robot teaching using task cohesion," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, 2022, pp. 367–374.

[17] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu, A. Handa, and D. Fox, "Factory: Fast contact for robotic assembly," 2022.

[18] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *ECCV*, 2022.

[19] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.