

Large Language Models for Orchestrating Bimanual Robots

Kun Chu*, Xufeng Zhao, Cornelius Weber, Mengdi Li, Wenhao Lu, and Stefan Wermter

Abstract—Although there has been rapid progress in endowing robots with the ability to solve complex manipulation tasks, generating control policies for bimanual robots to solve tasks involving two hands is still challenging because of the difficulties in effective temporal and spatial coordination. With emergent abilities in terms of step-by-step reasoning and in-context learning, Large Language Models (LLMs) have demonstrated promising potential in a variety of robotic tasks. However, the nature of language communication via a single sequence of discrete symbols makes LLM-based coordination in continuous space a particular challenge for bimanual tasks. To tackle this challenge, we present LAngeage-model-based Bimanual ORchestration (LABOR), an agent utilizing an LLM to analyze task configurations and devise coordination control policies for addressing long-horizon bimanual tasks. We evaluate our method through simulated experiments involving two classes of long-horizon tasks using the NICOL humanoid robot. Our results demonstrate that our method outperforms the baseline in terms of success rate. Additionally, we thoroughly analyze failure cases, offering insights into LLM-based approaches in bimanual robotic control and revealing future research trends. The project website can be found at <http://labor-agent.github.io>.

I. INTRODUCTION

While humans routinely operate with both hands in their daily lives, achieving effective bimanual coordination remains a formidable challenge in robotics as the dual-arm setting faces complexity in terms of temporal and spatial coordination. Bimanual manipulation offers a significant difficulty for the existing planning-based and learning-based methods. Previous planning-based methods [1], [2], [3] largely focus on motion planning for controlling two arms in manipulating large objects under certain constraints, limiting their application in dynamic or complex task spaces. On the other hand, learning-based methods, such as Reinforcement Learning (RL) and Imitation Learning (IL), enable a robot to learn control policies from human-designed rewards [4], [5] or human-teleoperated demonstrations [6], [7]. However, it is known that RL algorithms are notorious as hard to train, especially in dual-arm settings that have high degrees of freedom (DoFs), and collecting human demonstrations for bimanual robots is labor-intensive and time-consuming. At the same time, the high complexity associated with the variety of bimanual patterns suggests that low-level control

This research was funded by the German Research Foundation (DFG) in the project Crossmodal Learning (TRR-169) and by the China Scholarship Council (CSC).

The authors are with the Knowledge Technology Group, Department of Informatics, University of Hamburg, 22527 Hamburg, Germany. E-mail: {kun.chu, xufeng.zhao, cornelius.weber, wenhao.lu, stefan.wermter}@uni-hamburg.de, mengdi.li@studium.uni-hamburg.de.

*Corresponding author.

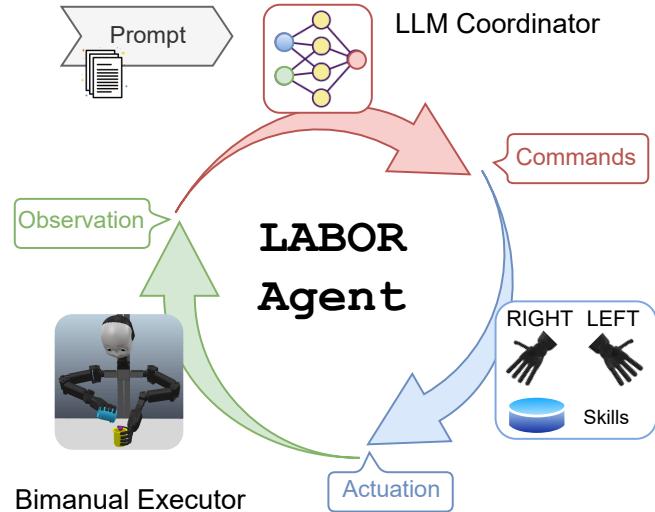


Fig. 1. Illustration of the LABOR agent. During the execution of the task, with the guiding prompt, the LLM coordinator first decomposes the task and then generates the step action plan, including the control command for the left and right hand. The bimanual robot executor then performs actions on the environment according to the commands, and the results provide feedback to the LLM for the next action, and so on until the end of the task.

policies and high-level planning must be considered for an integrated control system design [8].

Large Language Models (LLMs) have demonstrated remarkable knowledge and reasoning capabilities [9], [10], leading to a revolutionary change in numerous fields [11]. In robotics, an LLM is typically employed as a high-level planner to reason and invoke primitive skills for accomplishing complex tasks described in natural language [12], [13], [14]. However, existing methods primarily focus on uni-manual manipulations with complex chains of skills and the integration of LLMs in bimanual control remains unexplored. In this research, we explore further the feasibility of leveraging LLMs in robotic bimanual control, especially focusing on high-level temporal and spatial coordination between two hands. Our research question is: *How can LLMs be employed to achieve versatile bimanual manipulations in tasks with complex spatial relationships?*

To this aim, we introduce the **LABOR** (LAngeage-model-based Bimanual ORchestration) agent, which uses an LLM to effectively coordinate bimanual control to deal with complex manipulation tasks, as shown in Fig. 1. Specifically, we introduce spatio-temporal bimanual control patterns and design the guiding prompt for LLMs. The prompted LLM decomposes the task execution into corresponding uncoordi-

nated and coordinated stages to account for the spatial relationships and task requirements. Then it generates the step action commands for two hands respectively based on a pre-defined skill library. This proceeds iteratively until the long-horizon task is completed. We evaluate the LABOR agent on a simulation of the semi-humanoid bimanual NICOL, the Neuro-Inspired COLlaborator [15], on two classes of challenging long-horizon tasks. Compared to the Baseline agent, the higher success rate of the LABOR agent on these tasks indicates its efficacy in coordinating both hands to perform complex long-horizon tasks.

II. RELATED WORK

In this section, we describe the related work on planning-based and learning-based methods for bimanual manipulations, and LLM's use in robotics.

A. Bimanual Manipulation

1) Planning-based methods: Early works [16], [1], [8], [17] mainly focus on planning algorithms for dual-arm settings under the closed-chain constraint, which rely on static known dynamic models to generate kinematic configurations for two arms. With engineered primitives or pre-defined trajectories [2], [18], [19], [3], bimanual robots exhibit some flexibility in dealing with complex environments. However, such planning-based methods mainly rely on task-specific primitives, which are costly to design or collect in a more diverse task setting.

2) Learning-based methods: Some works have employed RL [20], [5], [4], [21] and IL [6], [22], [23], [7] framework in dual-arm settings. These approaches typically require massive samples and training time, due to the high dimensionality of the action space in a bimanual setting. Although some recent works proposed to design parameterized primitives to reduce the size of the action space for exploration, these movement primitives normally require costly hard-coded motion design or human-teleoperated demonstration collection, limiting their applications in complex environments.

The approaches listed above do little to consider the properties of the task at a higher level but rather adopt a single control policy for controlling the robot to accomplish the task. In this sense, the control policy in their tasks is normally fixed and monolithic which tackles only simple and short-duration tasks. In contrast, we explore the usage of LLMs to generate control policies from a higher perspective that considers the uncoordinated and coordinated processes for task accomplishment, thus adapting to complex long-horizon tasks.

B. LLM in Robotics

Learning robotic control policies to address various manipulation tasks often requires the collection of massive datasets [24], [25], [26], [27]. Recent works have leveraged common sense and established knowledge from LLMs to control robots in tackling complex tasks. Some studies have delved into distilling LLM expertise into localized models. For instance, researchers have utilized LLMs' programming

capabilities to generate executable code for fulfilling open instructions [28], [29], [30], or to provide direct guidance for RL policies [31], [32]. Other approaches involve decomposing complex tasks into manageable sub-tasks using LLMs, thus enabling robots to address them individually with pre-defined skills [13], [12], i.e., language-conditioned policies. We adopt the latter paradigm to construct a versatile robot equipped with atomic skills that can be efficiently reused across multiple tasks or potentially enhanced for new abilities [14]. Specifically, our robot is furnished with general skills for both arms, offering a vast number of potential combinations. However, this inherent complexity of combinatorial capabilities poses challenges to efficient reasoning and planning in bimanual manipulation, which will be explored in this study.

III. METHOD

This section introduces the LABOR agent from two perspectives: 1) Spatio-temporal Bimanual Control, where we present a series of control patterns that form the foundation for the agent's practical design; 2) LLM Orchestration, where we describe the design of an LLM-based pipeline for implementing various control patterns in bimanual control.

Bimanual manipulation control mainly involves the consideration of coordination and, when required, synchronization. In a macroscopic view of long-horizon bimanual manipulation tasks, a combination of coordination stages can be involved. For example, in the task of placing an apple in a bowl and holding the bowl to a serving position, both uncoordinated processes (grasping and releasing the apple) and symmetrically coordinated processes (holding and moving the bowl with two hands) are required. During task execution, the type of control patterns should first be decided based on the current spatial and temporal situations, and then specific control commands should be generated.

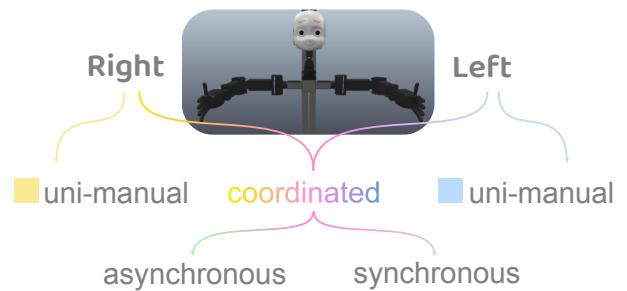


Fig. 2. Spatio-temporal control adopted by the LABOR agent.

A. Spatio-temporal Bimanual Control

Inspired by earlier works [6], [33], [34] on dual-arm manipulations, as shown in Fig. 2, we introduce the following spatio-temporal control patterns for two hands of a bimanual robot to manipulate on a worktable as follows,

- Uncoordinated Control: two hands act and manipulate independently in their own working spaces (i.e., *left*

area and *right area* respectively), referred to as uni-manual manipulation for each hand. For safety concerns and better efficiency, the *left area* is not accessible by the right hand, and vice versa.

- Coordinated Control: two hands act and manipulate independently in terms of either spatial or temporal relation, e.g., co-working on the *overlap area*. The collaboration between two hands can be either asynchronous or synchronous — An asynchronous type of control involves one hand constructing pre-conditions for the other, while a synchronous control indicates a, usually precise, mutual dependency between them.

This allows efficient control of both hands: in *left area* and *right area* the two hands can be controlled for different manipulations in parallel, while in *overlapping area* the temporal order, i.e. synchronous or asynchronous, is made explicit depending on the task requirements and spatial relationships between objects.

B. LLM Orchestration

LLMs exhibit remarkable abilities in terms of step-by-step reasoning and instruction-following responses in robotics, which makes it possible to generate policies under temporal and spatial constraints in bimanual manipulations. As depicted in Fig. 3, the designed prompt contains background information regarding the bimanual robot, like coordinates and workspaces of two hands, following some rules specifying the bimanual control principles in different stages. At the same time, the skill library is available to the LLM in the form of tool APIs. Given this, the prompts explicitly require the LLM to decompose the task into uncoordinated and coordinated stages and generate specific action plans based on the provided skill library. After executing an action assigned by the command, the LLM obtains observations from the environment. Iterating in this way, the LLM progressively generates a solution to the task while self-correcting mistaken commands, thus generating a chain of skills for different stages throughout the lifespan of the task. An example of LABOR agent reasoning is shown in Fig. 4, where the LLM coordinator generates commands for two hands respectively based on the task requirements and the spatial relationships of the objects, and then the bimanual robot executor performs the corresponding actions in the task environment. In summary, the LABOR agent uses the LLM to explicitly coordinate two hands based on the decomposed stages while continuously generating skill-based commands, to accomplish complex long-horizon bimanual tasks effectively.

IV. EXPERIMENTS

To determine the extent to which an LLM can manage to control the bimanual robot in everyday tasks, we deploy the LABOR agent on the semi-humanoid NICOL¹, the Neuro-Inspired COLlaborator [15], on two classes of bimanual tasks

¹The NICOL robot website can be found at <https://www.inf.uni-hamburg.de/en/inst/ab/wtm/research/neurorobotics/nicol.html>

You are a humanoid robot on a worktable to solve the task given by a human user in a simulated environment.

Background:

Coordinates are provided in terms of [x, y, z] in meters in the 3D world. The x-coordinate is from back to front, the y-coordinate is from right to left, and the z-coordinate is from bottom to up.

There are three areas on the worktable:

- Left Area (y coordinate $y > 0.2$): only the left hand can reach;
- Right Area (y coordinate $y < -0.2$): only the right hand can reach;
- Overlap Area ($-0.2 < y < 0.2$): both hands can reach.
- The robot's hands are initialized vertically and open at the beginning.

Environment Setting:

There are two cups placed on the table:

- One blue cup is filled with water, with the coordinate as $\{[x_1, y_1, z_1]\}$
- One yellow cup is empty, with the coordinate as $\{[x_2, y_2, z_2]\}$
- Marked points names are listed as ...

Task:

Pour the water from the blue cup into the yellow cup, put the blue cup back, and serve the water to the human user at the serve point. Do not release the cup at the serving point.

Guidelines:

If task-related objects are at different areas (uncoordinated stages, two hands required)

- 1) *uncoord_both*: two hands both ...

If task-related objects are in the same area (uncoordinated stages, one single hand required)

- 2) *unimanual_left*: only the left hand acts...
- 3) *unimanual_right*: only the right hand acts...

Cooperation between two hands in the overlap area (coordination stages, two hands required)

- 4) *coord_both*: two hands manipulate the object in the overlap area independently.
 - a. *sync*: both hands act symmetrically and simultaneously...
 - b. *async_left*: the right hand's action relies on the left hand's action first...
 - c. *async_right*: the left hand's action relies on the right hand's action first...

Decompose the task into an updated list of appropriate stages...

Fig. 3. Prompts for the LLM to orchestrate a bimanual robot to accomplish the task. The general prompting template, indicated in standard black text, remains consistent across all tasks, while task-specific details such as environment setting and task description are shown in gray.

in the CoppeliaSim² simulator. To validate the effectiveness of the LABOR agent in coordinating the bimanual control, we also introduce the vanilla LLM Agent as a baseline, in which our designed guiding prompts are not provided.

A. LABOR Agent Setup

We design a skill library³ for the NICOL robot, mainly including two kinds of hand-crafted skills, *Manipulation* skills denote the specific action designed for one hand to perform, while *Information* skills are used to obtain information about the robot and the environment. Specifically, the skills with parameters shown in parenthesis are listed below.

Manipulation skills:

- `move_to(side, obj_name)` moves the hand on the specified side to the position of the object named

²<https://www.coppeliarobotics.com>

³The skills are developed based on the `nicol_coppeliasim` Python library, https://github.com/knowledgetechnologyuhh/nicol_coppeliasim



Fig. 4. Example of a LABOR agent’s reasoning. With the guiding prompt, the LLM decomposes the task into multiple stages, i.e., uncoordinated and coordinated stages, and then generates action plans with skill primitives for both hands until the task is accomplished.

`obj_name`, along with a currently grasped object if any;

- `move_and_grasp(side, obj_name)` moves the hand on the specified side to the position of the object named `obj_name` and grasp it;
- `move_above(side, obj_name)` moves the hand on the specified side above the object named `obj_name`, with any possibly grasped objects;
- `push_to(side, source_obj, target_obj)` moves the hand on the specified side to push the object named `source_obj` from its current position to the position of the object named `target_obj`;
- `pour_out(side)` controls the hand on the specified side to turn the wrist to flip down. If there is an object grasped, then its content will be poured out;
- `release(side)` is used to open one hand on the specified side to release grasped objects;
- `reset(side)` is used to reset the hand on the specified side to its initial configurations;
- `wait(side)` is used to hold the hand on the specified side, including any possibly grasped objects, in its present state.

Information skills:

- `get_arm_state(side)` gets the status (i.e., hand positions, palm angles, and if the fingers are open or closed) of the hand on the specified side;
- `get_obj_position(obj_name)` gets the position of the specified `obj_name`;

These skills are designed to be generic and versatile, enabling the humanoid robot to perform complex, long-horizon manipulations involving interactions between objects and its hands. More importantly, the skills are tailored to the specific characteristics of humanoid robots, including hand movement, palm angles, and finger movements. This also points to a direction for skill design in humanoid robots: Firstly, overall hand movement toward the destination, and then, more fine-grained maneuvers provided by palm angles and finger movements. At the same time, to avoid some potential errors or collisions, some automated precondition detections are also considered in the design of the skill, for instance, when grasping a certain object, the fingers must be open, etc. In this sense, spatial dependencies or

constraints within humanoid robots are established through some automatic checks, while providing flexible choices to the LLM.

We use the OpenAI GPT-4o model as the LLM for inference in the LABOR agent. Since the accomplishment of bimanual tasks requires the LLM to generate an action plan for two hands several times, we take the LangChain⁴ Python library framework to enable the LLM to generate a chain of skills step by step. Based on the LangChain tool design structure, two types of tools APIs are designed: (1) `bimanual_control(left_command, left_para, right_command, right_para)`, in which the available choices are taken from the *Manipulation* skills with appropriate inputs; (2) `get_information` functions for the *Information* skills. It should be noted that the `wait` command is used to control the temporal order between two hands in asynchronous control, and when two hands are assigned the same action for the same object, then two hands will act synchronously for cooperation.

B. Task Environment Setup

NICOL has a head that can express stylized facial expressions, and two robotic *Open-Manipulator-Pro* arms with adult-sized five-fingered Seed Robotics RH8D hands (for details see [15]). With such equipment, NICOL has a large bimanual workspace to handle everyday objects and tasks. NICOL’s workspace with objects in the real and simulated world is shown in Fig 5. There are two classes of tasks designed in the experiments: ServeWater and ServeFruit, which all require the cooperation of two hands with a chain of skills to accomplish.

In the ServeWater task class, there are two cups on the work table. The yellow cup is empty and the blue cup is water-filled (we use a small ball as simulated water). The goal of the task is to use the yellow cup to serve water at a specified serving position. This means that the water has to be transferred from the blue cup to the yellow cup, which is then lifted to the serving position, while the blue cup is released back to its original position. The cups can be in two different spatial relations: in the same (both right or both left) or in different areas (one on the left and another

⁴<https://github.com/langchain-ai/langchain>

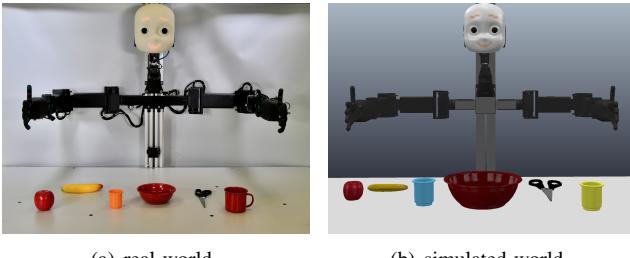


Fig. 5. NICOL workspace with daily objects in real and simulated worlds, from left to right: apple, banana, cup (orange or blue), bowl, scissors, and cup (red or yellow). In this work, experiments are designed and completed in the simulated environment, leaving real-world exploration for the foreseeable future.

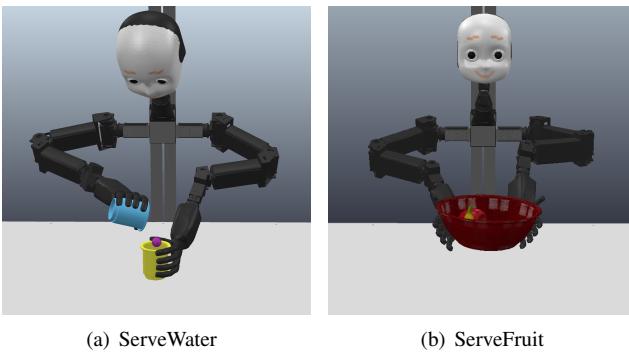


Fig. 6. Critical stages illustrated in the overlap area for two classes of tasks with the NICOL robot: (a) the right hand flips down the water-filled cup to pour out the ‘water’ (small ball) to the empty cup while the left hand is holding the grasped empty cup; (b) left and right hands together are holding the bowl with the apple and banana inside.

on the right). Since the robot cannot reach with its right hand over the *left area*, and vice versa, this task would require two hands to work in cooperation in some cases. However, this relationship information is not passed explicitly to the LLM, which instead receives raw information about the coordinates of the two cups, placed on the table at random positions. The task itself requires a long skill chain to accomplish and coupled with the changing positional relationship between cups and hands, choosing appropriate control policies effectively is even more challenging.

In the ServeFruit task class, there is one large bowl, one apple, and one banana on the work table. The goal of the task is to grasp the fruits, release them to the bowl, and serve the bowl to the human user at the serving point. Similar to the ServeWater task, the positions of the apple, banana, and bowl bring complexity and variety to this task, i.e., the fruits are in the same or different area while the bowl can be at the left or right area, as well as the decomposition of tasks involving uncoordinated and coordinated stages. To be more specific, the grasping and lifting of the objects are independent and uncoordinated, but holding the bowl depends on two hands to hold synchronously. The most complex components also lie in that the bowl is not at the overlap area at first, thus requiring it to be pushed to the overlap area, to receive the objects from another area and be controlled by two hands.

C. Results

In this section, we report the results of the LABOR agent in controlling the NICOL robot to solve the above tasks, including success rate and failure analysis. To evaluate the effectiveness, we repeated 40 tasks for each class, and for each time a variant of the task was randomly chosen. The success rates and failure rates in three aspects of LABOR when using GPT-4o as the LLM on these tasks are listed in Table I. It can be observed that, compared with the baseline method, the LABOR agent demonstrates higher performance in orchestrating bimanual control in long-horizon manipulation tasks, especially in the ServeFruit task.

Regarding the properties of bimanual manipulation and the LLM itself, we found that the failure cases mainly come from two aspects namely *Temporal Coordination* and *Spatial Coordination*. In this sense, *Temporal* failure refers to the fact that the LLM fails to generate skills in the correct order or neglects some processes during the task execution, while *Spatial* failure refers that the LLM mishandles the spatial relationships in the operation, such as the relationship between two objects or between the robot’s hand and an object.

By checking the plans generated by LLMs in each failed task, we found that in the ServeWater task, the *Temporal* failures mainly occurred when the pouring water process was not executed during the task accomplishment, while *Spatial* failures mainly on that such process happened without moving above the yellow cup, causing the water not being transferred to another cup. Similarly in the ServeFruit task, *Spatial* failures mainly come from the incorrect releasing process of the apple or banana, while the *Temporal* failures are mainly due to the lack of such correct processes, causing the bowl to be served to the target point without fruits or only one single fruit.

TABLE I. Experiment results on two classes of bimanual robotic tasks showcase that our method outperforms baseline in terms of success rate.

	ServeWater			ServeFruit		
	Success Rate	Failure		Success Rate	Failure	
		Spatial	Temporal		Spatial	Temporal
Baseline	65.0%	27.5%	7.5%	45.0%	37.5%	17.5%
LABOR	77.5%	17.5%	5.0%	82.5%	12.5%	5.0%

In addition to this, we also see a few cases in Baseline agent where the LLM tries to perform a transfer of an object through two hands through the *overlap area*, instead of performing a manipulation about two objects through the *overlap area*. For example, in the Servewater task, the LLM tries to move the blue cup to another area to pour the water and then transfer it back through the overlap area, which is an interesting phenomenon, but unfortunately due to the long number of steps required for this kind of reasoning, the LLM interrupts prematurely resulting in the task not proceeding to the last step. In contrast, for the LABOR agent, the LLM

innovatively uses the skill of `push_to` to push the blue cup to the overlap area, which is feasible, but then the LLM ignores the fact that the cup is not held by the hand (requiring `move_and_grasp` process), failing in the task eventually.

Excluding these failure cases, one can easily observe that the LABOR agent exceeds the success rate of the Baseline agent in both long-horizon bimanual tasks, especially the ServeFruit task, demonstrating its outstanding efficacy in coordinating two hands in complex tasks.

D. Discussion

Although the guiding prompt is structured to coordinate two-hand control while allowing each hand to operate independently, it inadvertently biases the LLM towards using both hands as much as possible. Due to the generic skills we designed for the NICOL robot and the flexibility provided by the overlap area, the LLM agents occasionally exhibit unexpected behaviors. For instance, when two cups are in different areas, the most efficient approach would be to lift both cups to the overlap area to pour out the water and prepare for future actions. However, the LLM agent might instead transfer the object from the left hand to the right across the overlap area, pouring out the water in a different area.

Experimental results show that these behaviors still achieve the desired outcomes. For example, when fruits are located in different areas, the most efficient method is to lift and release the fruit on the same side as the bowl and push the bowl to the overlap area to add another fruit. However, the LLM occasionally uses the overlap area as a transition point to transfer the fruit, which, although accomplishing the task, slows down progress and incurs a higher likelihood of task failure. While we anticipated some anomalies when designing the skills, these unanticipated behaviors provide valuable insights for developing more complex skills in future real-world applications. For example, designing fine-grained precondition detection mechanisms for each skill or describing these conditions accurately in prompts so that LLMs can better understand the skill.

V. CONCLUSIONS

In this work, we introduce LABOR, an agent that utilizes an LLM to effectively orchestrate bimanual control when performing humanoid manipulation tasks. Under the spatio-temporal control patterns designed in guiding prompts, the LABOR agent decomposes the task into several stages involving uncoordinated and coordinated steps, according to the spatial relationships and task requirement. Based on the generated plan, the LLM then iteratively generates action steps based on the environmental feedback during task executions. Experimental results on the NICOL robot for two classes of long-horizon everyday tasks showcase the superior performance of the LABOR agent compared to the Baseline agent without guiding prompts.

Limitations and Future Work. Similar to other LLM-based methods, the LABOR agent is dependent on the performance of the LLM like the GPT-4o model, requiring

its reasoning and understanding of the correlations of primitive skills and task solutions. Besides, since the agent is currently solely running in the simulated environment instead of the real world, some potential differences in environmental dynamics might not be sufficiently analyzed or considered, like self-collision detection between the hands, which brings some unknowns in transferring the agent. In future work, the LLM can be further used to guide the learning of versatile skills in controlling humanoid hands [31], [35], which can endow the LABOR agent with fine-grained manipulation skills for everyday objects with various shapes. Moreover, vision-based foundation models [36], [37] can enhance the scene understanding capabilities of the LABOR agent when dealing with real-world tasks. As skills expand, the LABOR agent should be able to control bimanual robots to perform more complex long-horizon tasks.

VI. ACKNOWLEDGMENT

The authors would like to thank OpenAI and their Researcher Access Program for generously providing GPT-4o API tokens support, and Jan-Gerrit Habekost and Lennart Clasmeier for their work and support for the NICOL robot in the simulation.

REFERENCES

- [1] N. Vahrenkamp, M. Przybylski, T. Asfour, and R. Dillmann, “Bimanual grasp planning,” in *2011 IEEE-RAS 11th International Conference on Humanoid Robots (Humanoids)*, 2011, pp. 493–499.
- [2] S. S. Mirrazaei Salehian, N. B. Figueira Fernandez, and A. Billard, “Dynamical system-based motion planning for multi-arm systems: Reaching for moving objects,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 4914–4918.
- [3] E. Ng, Z. Liu, and M. Kennedy, “It takes two: Learning to plan for human-robot cooperative carrying,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7526–7532.
- [4] K. S. Luck and H. B. Amor, “Extracting bimanual synergies with reinforcement learning,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4805–4812.
- [5] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta, “Efficient bimanual manipulation using learned task schemas,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1149–1155.
- [6] R. Zollner, T. Asfour, and R. Dillmann, “Programming by demonstration: dual-arm manipulation tasks for humanoid robots,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2004, pp. 479–484.
- [7] S. Stepputtis, M. Bandari, S. Schaal, and H. B. Amor, “A system for imitation learning of contact-rich bimanual manipulation policies,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 810–11 817.
- [8] C. Smith, Y. Karayannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, “Dual arm manipulation—a survey,” *Robotics and Autonomous systems*, vol. 60, no. 10, pp. 1340–1353, 2012.
- [9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, May 2022. [Online]. Available: https://openreview.net/forum?id=_VjQlMeSB_J
- [10] X. Zhao, M. Li, W. Lu, C. Weber, J. H. Lee, K. Chu, and S. Wermter, “Enhancing zero-shot chain-of-thought reasoning in large language models through logic,” in *2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, May 2024.

- [11] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang, “Sparks of Artificial General Intelligence: Early experiments with GPT-4,” no. arXiv:2303.12712, p. 2303.12712, Apr. 2023. [Online]. Available: <http://arxiv.org/abs/2303.12712>
- [12] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, et al., “Do as I can, not as I say: Grounding language in robotic affordances,” in *Proceedings of The 6th Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 287–318.
- [13] X. Zhao, M. Li, C. Weber, M. B. Hafez, and S. Wermter, “Chat with the Environment: Interactive Multimodal Perception using Large Language Models,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. arXiv:2303.08268. arXiv, Oct. 2023, p. 2303.08268. [Online]. Available: <http://arxiv.org/abs/2303.08268>
- [14] J. Zhang, J. Zhang, K. Pertsch, Z. Liu, X. Ren, M. Chang, S.-H. Sun, and J. J. Lim, “Bootstrap Your Own Skills: Learning to Solve New Tasks with Large Language Model Guidance,” in *7th Annual Conference on Robot Learning*. arXiv, Oct. 2023, p. 2310.10021. [Online]. Available: <http://arxiv.org/abs/2310.10021>
- [15] M. Kerzel, P. Algeuer, E. Strahl, N. Frick, J.-G. Habekost, M. Eppe, and S. Wermter, “NICOL: A neuro-inspired collaborative semi-humanoid robot that bridges social interaction and reliable manipulation,” *IEEE Access*, vol. 11, pp. 123 531–123 542, 2023.
- [16] Y. Koga and J.-C. Latombe, “Experiments in dual-arm manipulation planning,” in *Proceedings 1992 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Computer Society, 1992, pp. 2238–2239.
- [17] B. Nemec, N. Likar, A. Gams, and A. Ude, “Bimanual human robot cooperation with adaptive stiffness control,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 607–613.
- [18] P. Lertkultanon and Q.-C. Pham, “A certified-complete bimanual manipulation planner,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1355–1368, 2018.
- [19] D. P. Losey, M. Li, J. Bohg, and D. Sadigh, “Learning from my partner’s actions: Roles in decentralized robot teams,” in *Conference on robot learning*. PMLR, 2020, pp. 752–765.
- [20] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, Z. Lu, S. McAleer, H. Dong, S.-C. Zhu, and Y. Yang, “Towards human-level bimanual dexterous manipulation with reinforcement learning,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 5150–5163, 2022.
- [21] F. Amadio, A. Colomé, and C. Torras, “Exploiting symmetries in reinforcement learning of bimanual robotic tasks,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1838–1845, 2019.
- [22] R. Lioutikov, O. Kroemer, G. Maeda, and J. Peters, “Learning manipulation by sequencing motor primitives with a two-armed robot,” in *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference (IAS-13)*. Springer, 2016, pp. 1601–1611.
- [23] L. P. Ureche and A. Billard, “Constraints extraction from asymmetrical bimanual tasks and their use in coordinated behavior,” *Robotics and Autonomous Systems*, vol. 103, pp. 222–235, 2018.
- [24] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *Journal of Machine Learning Research*, vol. 22, no. 30, pp. 1–82, 2021.
- [25] M. Li*, X. Zhao*, J. H. Lee, C. Weber, and S. Wermter, “Internally Rewarded Reinforcement Learning,” in *40th International Conference on Machine Learning (ICML)*, July 2023, p. 2302.00270. [Online]. Available: <http://arxiv.org/abs/2302.00270>
- [26] X. Zhao, C. Weber, M. B. Hafez, and S. Wermter, “Impact Makes a Sound and Sound Makes an Impact: Sound Guides Representations and Explorations,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2512–2518.
- [27] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans, “Foundation Models for Decision Making: Problems, Methods, and Opportunities,” *arXiv preprint arXiv:2303.04129*, p. 2303.04129, 2023.
- [28] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, “VoxPoser: Composable 3D value maps for robotic manipulation with language models,” in *7th Annual Conference on Robot Learning (CoRL 2023), Atlanta, Georgia USA*, 2023, p. 2307.05973. [Online]. Available: https://openreview.net/forum?id=9_8LF30mOC
- [29] G. Wang, Y. Xie, Y. Jiang, A. Mandlikar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, “Voyager: An Open-Ended Embodied Agent with Large Language Models,” no. arXiv:2305.16291, p. 2305.16291, Oct. 2023. [Online]. Available: <http://arxiv.org/abs/2305.16291>
- [30] 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)*, 2023, pp. 9493–9500.
- [31] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, “Eureka: Human-level reward design via coding large language models,” in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: <https://openreview.net/forum?id=IEduRUO55F>
- [32] K. Chu, X. Zhao, C. Weber, M. Li, and S. Wermter, “Accelerating reinforcement learning of robotic manipulations via feedback from large language models,” in *7th Conference on Robot Learning (CoRL 2023) Workshop, Atlanta, Georgia USA*, Nov. 2023.
- [33] J. R. Boehm, N. P. Fey, and A. M. Fey, “Online recognition of bimanual coordination provides important context for movement data in bimanual teleoperated robots,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6248–6255.
- [34] F. Krebs and T. Asfour, “A bimanual manipulation taxonomy,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 031–11 038, 2022.
- [35] X. Zhao, C. Weber, and S. Wermter, “Agentic skill discovery,” *arXiv preprint arXiv:2405.15019*, 2024.
- [36] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao, “Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4v,” *arXiv preprint arXiv:2310.11441*, 2023.
- [37] M. Oquab, T. Darcret, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, “DINOv2: Learning robust visual features without supervision,” *Transactions on Machine Learning Research*, 2024. [Online]. Available: <https://openreview.net/forum?id=a68SUt6zFt>