OpenPyRo-A1: An Open Python-based Low-Cost Bimanual Robot for Embodied AI

Helong Huang[1,*], Christopher E. Mower[1,*], Guowei Huang[1,*], Sarthak Das[1], Magnus Dierking[1], Guangyuan Luo[1], Kai Tan[1], Xi Chen[1], Yehai Yang[1], Yingbing Chen[1], Yiming Zeng[1], Yinchuan Li[1], Zhanpeng Zhang[1], Shuang Wu[1], Yingxue Zhang[1], Weichao Qiu[1], Tongtong Cao[1], Yuzheng Zhuang[1], Guangjian Tian[1], Jianye Hao[1], Jun Wang[2], Haitham Bou-Ammar[1,2,†], Xingyue Quan[1,†]

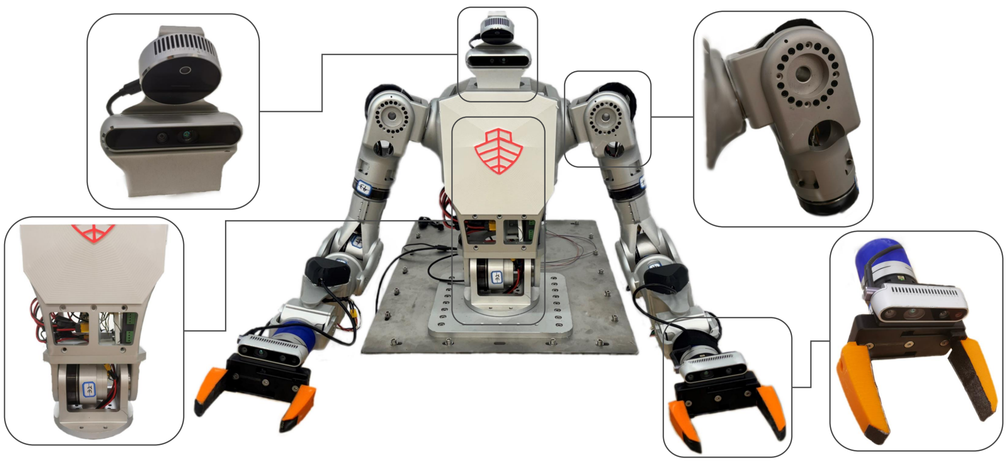

Fig. 1: The OpenPyRo-A1 bimanual humanoid robot.

***Abstract*— Many real-world tasks, such as assembly, cooking, and object handovers, require bi-manual coordination. However, learning such skills via imitation for these systems remains challenging due to dataset scarcity, driven mainly by the high cost of bi-manual robotic platforms and the barriers to entry in robotics software. To address those challenges, this paper contributes OpenPyRo-A1, a low-cost, bi-manual humanoid robot with a Python-first modular softwarse framework for control, planning, and skill learning. Our system supports VR-based data collection, imitation learning from vision and low-level positions, and integration with LLMs and VLMs for high-level task planning. We evaluate OpenPyRo-A1 on seven bi-manual tasks, collecting over 350 demonstrations via VR teleoperation and showcasing an agentic framework for executing tasks from natural language instructions. We hope that the contributions of the OpenPyRo-A1 hardware, the publicly available software stack, and the curated dataset of bi-manual manipulation episodes will advance affordable, scalable dual-arm robotics[1].**

I. INTRODUCTION

AI-based control in robotics has seen many remarkable advancements in recent years [1], [2], [3], [4], [5], [6], [7], [8], [9]. Particularly, methods for imitation learning [10], [11], [12], [13] have generated sophisticated policies by learning from data collected via human demonstrations. While successful, a key bottleneck is the requirement of large datasets to learn effective and scalable controllers.

Unlike natural language processing and computer vision, which have access to vast data sources from the internet, robotics data is limited [14] due to the high hardware cost and the complexity of real-world interactions. To tackle these challenges, large-scale data collection initiatives have been launched [15], and scaling data collection has emerged as an increasingly prominent research direction [16], [11].

While a growing body of data is becoming available in robotics, most of it remains focused on single-arm systems. Although single-arm robots can handle various tasks, many tasks can be performed far more efficiently with a bi-manual arm setup, e.g., in furniture assembly, which may require the robot to steadily hold a part in place with one arm while drilling with the other [17], [18], [19], [20], or in automated cooking, where perfecting chopping requires one arm to stabilize the ingredient while the other performs precise cutting [21], [22].

Unfortunately, the problem of limited data resources is even more pronounced in the bi-manual setting, as there is a general lack of data collected for dual-arm systems. Even in the most prominent dataset, OpenX-Embodiment [15], only 3 out of the 22 embodiments (PR2, xArm bi-manual, and Baxter) are bi-manual, accounting for just 4 out of the 60 datasets. We argue that the high cost of bimanual robot platforms is a key reason for the scarcity of real-world data. Although bimanual robots are used in research, their high

[*,†]Equal contribution. [1]Huawei Noah's Ark Lab. [2]University College London.

[1]The hardware, software and dataset will be publicly available upon acceptance.

price and increased complexity limit widespread adoption.

In response, we developed the Open Python Robot A1 (OpenPyRo-A1)—a low-cost, bi-manual humanoid robot. Our design includes both the hardware and a distributed software framework, built from the ground up to ensure seamless system integration. In terms of hardware, OpenPyRo-A1 is designed to be affordable, easy to repair, and scalable for future upgrades. It features two 7-DoF arms (equipped with camera sensors), a torso, a head with an additional installed camera, and custom-built grippers. Our total hardware cost of approximately **$14K** is 2x less than comparable non-hobbyist projects like Reachy 1 from Pollen Robotic[2] and 4.6x less than their latest platform, Reachy 2.

Our modular software framework follows a Python-first design. It leverages the Lightweight Communications and Marshalling (LCM) library for communication to minimize maintenance overhead and reduce complexity. While the Robot Operating System (ROS) remains the dominant robotics framework, its complexity often demands expert-level knowledge even for simple applications. In contrast, our system simplifies dependency management while allowing seamless interchange of sensors, teleoperation interfaces, actuators, and communication protocols without extensive reconfiguration.

Additionally, given the architectural similarities between LCM and ROS, users can implement translation nodes for ROS integration, enabling them to utilize existing drivers while benefiting from our streamlined and flexible system.

Using our system, we focus on the problem of data collection, i.e. how humans can best provide task demonstrations for robotics. In the literature, the two main approaches are kinesthetic teaching and teleoperation. While kinesthetic teaching [23], [24] requires physically guiding the robot—often impractical for bi-manual systems—teleoperation enables remote control from a safe distance. Various teleoperation interfaces exist, such as gamepads or haptic devices [25], [26], but finding intuitive control mappings for effective operation remains challenging [27]. Although our system supports numerous interfaces, this work showcases virtual reality (VR) headsets as the teleoperation interfaces, leveraging their widespread availability in research labs and ability to provide an immersive control scheme that does not necessitate additional high-end haptic hardware.

We evaluated OpenPyRo-A1 on seven bi-manual tasks, ranging from pick-and-place to handover and pouring, demonstrating its versatility in real-world manipulation. Using our VR-based teleoperation system, we collected over 350 demonstrations, providing a rich dataset for training robotic skills. These demonstrations were then leveraged for imitation learning, enabling the robot to acquire and refine manipulation capabilities. To showcase the flexibility of our software framework, we implemented imitation learning at multiple levels: low-level joint position control using Dynamic Movement Primitives (DMPs) [28] and vision-based policy learning through Action Chunking with Transformers (ACT) [29], highlighting OpenPyRo-A1's ability to integrate both classical and modern learning paradigms.

Building on the trained skills, we propose an Embodied AI agent that integrates OpenPyRo-A1 with large language models (DeepSeek) and vision-language models (InternVL [30]) to enable high-level task planning using learned imitation skills. This agentic framework allows users to specify tasks using natural language, with the VLM providing scene descriptions to guide execution. We demonstrate our approach's success in a use case requiring the robot to identify and place specific fruits based on their colors into a basket.

The following is a summary of our main contributions:

- We developed OpenPyRo-A1, a low-cost dual-arm humanoid robot.
- A Python-first modular software framework with libraries for collecting data with VR teleoperation.
- A dataset of bi-manual manipulation episodes, integrated with machine learning algorithms for skill learning from vision and low-level positional data.
- An agentic framework for planning through large language and vision-language models.

II. RELATED WORK

The availability of data for single-arm manipulation tasks has led to the development of numerous successful imitation learning methods [10], [29], [31]. However, bimanual manipulation has not seen the same level of progress due to the lack of data, as stated previously, and due to cost.

To bring down the cost of robotics there have been numerous open-source efforts targeting single-arm systems [32], [33], [34], and even full humanoids [35], [36], mainly through platforms like the RoboCup tournament [37], [38], [39]. However there are only a few commercially available upper torsos and even fewer open source models [40], [41]. Due to easier access and broader commercial availability, research labs use humanoid robots for static bimanual tasks.

Using humanoids for these tasks tends to be unnecessarily complex for the specific purpose of object manipulation, often requiring more room and extra hardware to suspend/stabilize the robot. These complex robots make it difficult for smaller labs to do bimanual manipulation, forcing them to turn to other solutions.

One of these solutions involves placing two commercial single-arm robots mounted on a shared torso-like structure [42], [43], [44]. For example, Bi et al. [45] and Shake [46] use dual UR3 arms for tasks such as kitchen and liquid-mixing. Similarly, the APEX system [47] employs dual Franka arms for precision vector alignment. While effective, these DIY setups often void the warranty of the robots, while still bearing a significant cost to buy, e.g. two Franka Emika Panda or Universal robot arms.

The few available open-source fixed-base upper torso platforms are still expensive. For instance, the Pollen robot provides a torso with Python-based teleoperation capabilities, but its price ranges from 33K USD to over 50K USD while additionally relying on ROS. More affordable options, such

[2]https://www.pollen-robotics.com/

as [40], [41] or hobbyist-built robots, often fall short in key areas like payload capacity, accuracy or grasping capabilities. As a result, they do not provide a viable alternative for demanding bimanual tasks.

This leaves a clear gap in robotics for a fixed-base dual arm upper torso that is cheap yet provides reliable performance for data collection and research usage.

III. HARDWARE DESIGN

Our hardware design balances cost, performance, and flexibility. It features two robot arms, a torso, head, and custom grippers. Crafted for precision and modularity, it combines 3D printing for rapid prototyping with an aluminum alloy frame for strength, durability, and lightweight efficiency.

We consider the following three key principles when designing OpenPyRo-A1:

Low-cost: The robotic system is designed to be cost-effective while maintaining high performance. To minimize manufacturing expenses, the structure integrates 3D-printed components for non-load-bearing parts and CNC-machined aluminum alloy (which are also designed to be 3D printed) for critical structural elements, balancing durability with affordability. The system also employs off-the-shelf actuators, sensors, and electronics, reducing reliance on custom hardware and facilitating easier procurement. Furthermore, the design prioritizes assembly with minimal specialized tools, making it accessible to a broader range of users, including research institutions and small-scale industrial settings.

Ease of repair: A modular architecture ensures that individual components can be easily replaced or upgraded without requiring extensive disassembly. Key features include easy-access access panels (located in the torso), standardized fasteners, and color-coded wiring harnesses for simplified troubleshooting. Furthermore, all electrical and mechanical connections are quick-release or plug-and-play, minimizing downtime in case of failures. The gripper and end-effector mounting system use universal attachment points, allowing users to swap between different parts without requiring modifications.

Scalability: The system is designed to be easily upgraded as new technologies become available. The electronics and software architecture support modular firmware updates, ensuring compatibility with future peripherals. The frame includes pre-drilled expansion points, allowing users to attach additional sensors, cameras, or actuators without requiring extensive modifications. Furthermore, the power and communication architecture are designed with additional capacity, ensuring that new components can be integrated seamlessly without overloading the system.

A. Overall Specification

OpenPyRo-A1 has 18-DoF including the two arms, base joints and grippers. The robot has two custom grippers attached at the arm end-effectors.

Dimensions: The robot footprint is 70×70cm. The height from the base to the head is 65cm.

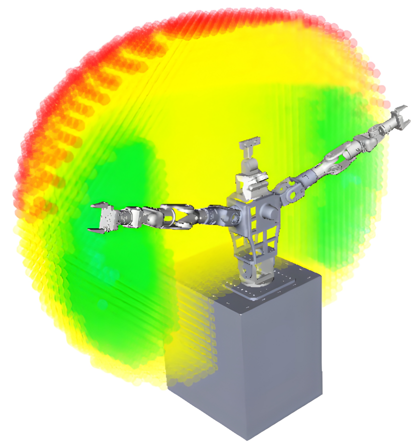

Fig. 3: Robot reachability map showing the right arm's workspace, including torso joints. Green indicates reachable areas, red indicates limited reach, and yellow represents intermediate zones.

Weight: The total weight of the robot is 47.5kg including torso and two arms.

Power: The robot is powered by a Mestek 1800W DC power supply, providing the necessary electrical energy for its operation.

Sensors: The joints of the robot report joint positions, velocities, and current. We have designed mounts for Intel Realsense cameras that can be attached on each gripper and two on the head. We maintain a modular design throughout the robot, so these cameras can be swapped out with others, not included, or re-positioned.

B. Arm and Gripper

Each arm on the robot is 7-DoF. A reachability analysis [48], [49] for the right robot arm including the torso joints is shown in Figure 3 – the kinematic model is symmetric so the same image is seen for the left arm (just inverted). The arm end-effector is designed with standard fittings so that any gripper—either the OpenPyRo default gripper or another custom design or commercial alternative—can be swapped out easily. The robot arm can be controlled in position mode at a maximum operating frequency of 100Hz.

The robotic arm features a 73 cm reach and is constructed from aluminum for durability and lightweight operation. Its actuators are powered by a 48V DC supply, with power transmission facilitated through a serial connection. The system's control circuitry directly supplies 48V to the motors, ensuring efficient power delivery.

A custom-designed parallel gripper has been developed for the robotic arm. The gripper operates at a frequency of 20 Hz and has a mass of 0.4 kg. It features a maximum finger width of 9.5 cm and a finger length of 8 cm, with fingers fabricated from TPU95A for enhanced flexibility and durability. The gripper's actuators are identical to those used in the arm, and motor communication follows the same protocol, ensuring seamless integration within the system.

C. Torso

The torso of the humanoid robot features a pan-and-tilt configuration with two joints at the base. The central section of the torso houses the motor controllers and power boxes. The robot utilizes EYOU motors, which offer quick response times, power-off switches, and a communication baud rate optimized for stable data transmission via CAN bus. The motors in the torso are powered and controlled in the same manner as the robotic arms, and the material used for the torso is identical to that of the arms. Communication with the motors is handled through CAN bus, with the system operating at a maximum frequency of 100 Hz for synchronized and efficient control.

D. Onboard processing unit

The robot is equipped with an onboard processing unit, using an OrangePi for control, the OrangePi board manages the robot's movements and coordinates communication between various components. Motor control is handled via a PD control node, operating at a frequency of 100 Hz. The OrangePi is capable of functioning within a distributed system setup, offering a reliable framework for robot control and facilitating seamless communication between different computers. Additionally, up to two cameras are integrated with the PD control node, providing synchronized visual feedback during operation.

IV. SOFTWARE ARCHITECTURE

To complement OpenPyRo-A1, we introduce `noahr`, a Python-first modular software framework designed for accessibility, even for those without advanced robotics knowledge. It follows a distributed architecture with a publisher-subscriber communication model, simplifying robot control, perception, and data collection. The framework is built to be lightweight and flexible enough to integrate with existing robotics ecosystems while being easy to install. A single `pip install` command sets up all required dependencies and the onboard OrangePi processing unit and external computing devices can be connected via a single Ethernet connection, enabling swift development process.

A. Communication and Robot Interface

Efficient communication and control are essential for real-time robotic operations. To achieve this, we leverage the Lightweight Communications and Marshalling (LCM) library [50], which supports custom message types and ensures low-latency, high-reliability communication. LCM is well-suited for transmitting extensive data at high frequencies, making it ideal for distributed robotics systems.

At the system's core, we provide a structured node class (`BaseNode`) that simplifies the creation of publishers and subscribers. This enables users to define message processing logic while managing execution frequencies efficiently.

We provide a high-level robot interface class to simplify accessing the robot, abstracting low-level communication details. This class enables users to: (i) access real-time robot state information (joint positions, velocities, and onboard sensors); (ii) execute joint position commands, whether streaming or trajectory-based; and (iii) utilize task-space control for higher-level robotic manipulation tasks. This class enables users to: (i) access real-time robot state information (joint positions, velocities, and onboard sensors); (ii) execute joint position commands, whether streaming or trajectory-based; and (iii) utilize task-space control for higher-level robotic manipulation tasks.

B. Kinematics and Control

To map goal states defined in task space to joint commands, we implement forward kinematics using Kinpy [51], modified to support CasADi backend for array, variable, and parameter representation, alongside the Spatial-CasADi library [52] for spatial transformations.

Motor control is performed through a custom CAN bus interface, built on Python's `ctypes` library. A PD controller regulates joint positions, enforces safety limits, and ensures smooth motion, publishing joint targets and states at 100 Hz.

C. Teleoperation and Data Collection

We provide an immersive, real-time control experience using the Meta Quest 3 headset for teleoperation. Additionally, our system supports alternative control methods, including PlayStation 4 controllers and keyboard-based operation. The framework streamlines data collection with LCM-based logging, enabling seamless recording of all network messages. A conversion script allows users to export raw data into CSV format for further analysis.

D. Debugging and Visualization

To support debugging and visualization, we provide live joint state visualization for real-time monitoring of robot movements, message inspection tools to track communication between components, and a graphical interface for network analysis built upon LCM's native tools, facilitating efficient system diagnostics.

E. Safety

Safety is a fundamental concern in robotic systems, and our framework incorporates multiple mechanisms to mitigate risks. A dedicated safety node acts as an intermediary between user commands and motor execution, ensuring that both joint and task-space limits are enforced. For additional protection, an optional workspace restriction mode prevents accidental self-collisions, allowing experienced users to disable it if needed. An emergency stop (E-stop) system is also being developed to enhance operational safety further.

V. DATASET AND DEMONSTRATIONS

This section overviews the dataset we plan to release and highlights several teleoperation demonstrations that showcase our dual-arm robot's capability to perform diverse and dexterous tasks.

Fig. 4: Snapshots from teleoperation demonstrations on OpenPyRo-A1. Tasks demonstrated are (from top-row to bottom): (i) fold clothes, (ii) open bag and pick out item, (iii) slice the apple, and (iv) wash the dish.

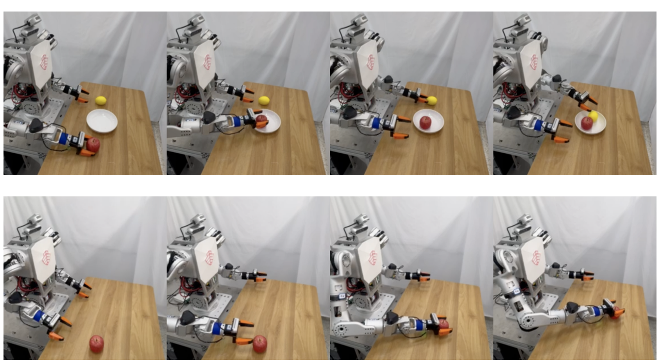

Fig. 5: Snapshots of imitation learning tasks with ACT. Tasks demonstrated are (from top-row to bottom): (i) dual-arm pick-place apple then lemon into the bowl, (ii) push apple to target.

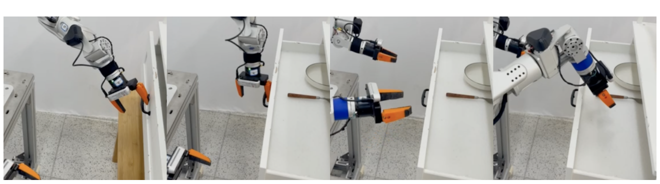

Fig. 6: Snapshots from imitation learning tasks with DMP for the task open cupboard and reach to pre-grasp trained

A. Dataset

The dataset accompanying this paper consists of over 350 demonstrations with 6 tasks, including (values in square brackets are the number of episodes per task). (i) **Exchange pepper** [60]: the robot picks up a pepper with one hand, exchanges it to the other, and places it down. (i) **Pick a cup and place** [60]: the robot picks a cup from the table, and places it into a bowl held by the other hand. (iii) **Pick lemon and apple, then place into bowl** [80]: the robot picks with one arm an apple, places it into the bowl, then uses the other arm to pick a lemon and places it into the same bowl. (iv) **Tip rice into bowl** [60]: the robot holds a bowl, then grasp a cup filled with rice and tips it into the bowl. (v) **Pour water** [50]: the robot uses one arm to hold a bowl, and then grasps a cup filled with water and pours the water into the bowl. (vi) **Push fruit to target** [50]: the robot pushes two pieces of fruit placed on the table to a target. Each episode is collected as an LCM-log file. We have provided helper scripts to load these as Pandas data-frames, or convert them to HDF5 format.

Every log file includes the following data streams. (a) Data from the VR headset. (b) The robot's current joint states. (c) Camera feeds from four Intel RealSense cameras. The L515 model is mounted in the head, the D455 model in the neck, and two D435 models are mounted on each gripper.

Fig. 7: Snapshots from the embodied AI use-case for the task “fill the basket with yellow fruit”.

Additionally, each message includes time stamps.

B. Teleoperation demonstrations

We showcase the robot’s ability to perform a variety of dexterous tasks using our VR teleoperation setup, which controls the end-effectors via the VR controllers. A custom IK solver maps the controller target poses to joint states, which are then sent to the robot as commands. We perform four tasks, shown in Figure 4: (i) fold clothes, (ii) open bag and pick out item, (iii) slice the apple, and (iv) wash the dish. Please See the accompanying video where we showcase several more teleoperation tasks.

VI. Use cases

This section describes two use-cases of our robot. First, we have trained several policies using imitation learning, and then integrated these in an agentic framework connected with an LLM (Deepseek) to chose actions, and a VLM (InternVL) to describe the scene.

A. Imitation Learning

Two common imitation learning methods were used in our use-case. We implemented Dynamic Movement Primitives (DMP) [28], [31] for joint position control and Action Chunking with Transformers (ACT) [29] for vision-based action generation.

For each task we collected 80 episodes using our VR teleoperation system. Two tasks were trained with ACT: (i) pick an apple and place into the bowl, then with the other arm, pick a lemon and place into the bowl, and (ii) push an apple to a target position. The snapshots from the experiments are shown in Figure 5.

We also used a DMP for the task: open cupboard with one arm and reach to a target pose with the other arm. This task is shown in Figure 6.

B. Embodied AI

The previous section demonstrated that our robot can learn and deploy multiple skills. Building on this, we further evaluate the system’s adaptability by integrating different imitation learning approaches, enabling both low-dimensional and vision-based action spaces. Here, we present an agentic framework for Embodied AI (Figure 8). Our goal is to show here that our system is capable to utilize the learned skills (from imitation learning) and orchestrate them to conduct a task defined by natural spoken language.

First, a prompt for the LLM is generated. A human records a task description using a microphone, and the audio is converted to text [53]. An image of the environment is then processed by a VLM to generate a scene description. Both

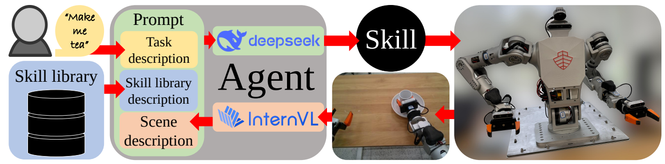

Fig. 8: Embodied AI agent framework on OpenPyRo-A1.

the task and scene descriptions are included in the prompt, along with a description of available skills and a history of previously enacted skills. At each step, the LLM selects the next skill to enact. We integrate the LLM (Deepseek) and VLM (InternVL) following the approach of Zeng et al. [30].

We learned two additional policies with DMP: reach to fruit and reach to basket. Then, we collected a joint configuration for grasping each fruit. The “reach to fruit” demonstration was then able to generalize by passing the collected joint configuration Two additional actions were implemented for opening and closing the gripper.

In total, the skills available for the agent are as follows: (1) open/close gripper, (2) reach to fruit (parameterized on which fruit), (3) reach to basket, (4) open-cupboard and reach to target.

We successfully demonstrated our Embodied AI framework on the task: “fill the basket with yellow fruit.” In each case, the skill library contained redundant skill, illustrating that the LLM was able to reason about skills that were unnecessary for the task. Furthermore, by integrating imitation learning methods, we highlight the framework’s ability to generalize across different skill execution strategies, combining high-level reasoning with learned low-level control policies.

VII. Conclusions

In this work, we introduce OpenPyRo-A1, a low-cost, modular dual-arm humanoid robot. Additionally, we present a distributed, Python-based framework for interfacing with the robot, as well as a pipeline for data collection, skill learning, deployment, and Embodied AI. We validate the hardware, software, and imitation learning pipeline through a series of experiments demonstrating the robot’s capability to complete complex manipulation tasks. Future work will focus on enhancing the software, developing integrated force control, adding a moving chassis or legs to the robot and testing alternative materials to further reduce cost and improve durability.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Open-vla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[5] S. Zhou, Y. Du, J. Chen, Y. Li, D.-Y. Yeung, and C. Gan, "Robo-dreamer: Learning compositional world models for robot imagination," *arXiv preprint arXiv:2404.12377*, 2024.

[6] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, "Unleashing large-scale video generative pre-training for visual robot manipulation," *arXiv preprint arXiv:2312.13139*, 2023.

[7] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, *et al.*, "Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation," *arXiv preprint arXiv:2410.06158*, 2024.

[8] J. Li, Q. Gao, M. Johnston, X. Gao, X. He, S. Shakiah, H. Shi, R. Ghanadan, and W. Y. Wang, "Mastering robot manipulation with multimodal prompts through pretraining and multi-task fine-tuning," *arXiv preprint arXiv:2310.09676*, 2023.

[9] C. E. Mower, Y. Wan, H. Yu, A. Grosnit, J. Gonzalez-Billandon, M. Zimmer, J. Wang, X. Zhang, Y. Zhao, A. Zhai, P. Liu, D. Palenicek, D. Tateo, C. Cadena, M. Hutter, J. Peters, G. Tian, Y. Zhuang, K. Shao, X. Quan, J. Hao, J. Wang, and H. Bou-Ammar, "Ros-llm: A ros framework for embodied ai with task feedback and structured reasoning," 2024. [Online]. Available: https://arxiv.org/abs/2406.19741

[10] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649241273668, 2024. [Online]. Available: https://doi.org/10.1177/02783649241273668

[11] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[12] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=agTr-vRQsa

[13] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=rif3a5NAxU6

[14] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=JrsfBJtDFdI

[15] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin, "Open X-Embodiment: Robotic learning datasets and RT-X models," https://arxiv.org/abs/2310.08864, 2023.

[16] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: Towards unleashing infinite data for automated robot learning via generative simulation," 2024. [Online]. Available: https://arxiv.org/abs/2311.01455

[17] Y. Cui, Z. Xu, L. Zhong, P. Xu, Y. Shen, and Q. Tang, "A task-adaptive deep reinforcement learning framework for dual-arm robot manipulation," *IEEE Transactions on Automation Science and Engineering*, 2024.

[18] Y. Liu, W. Su, Z. Li, G. Shi, X. Chu, Y. Kang, and W. Shang, "Motor-imagery-based teleoperation of a dual-arm robot performing manipulation tasks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 3, pp. 414–424, 2018.

[19] G. Tang, S. Liu, M. Sun, Y. Wang, W. Zhu, D. Wang, X. Li, H. Wu, S. Men, L. Zhang, *et al.*, "High-precision all-in-one dual robotic arm strategy in oral implant surgery," *BDJ open*, vol. 10, no. 1, p. 43, 2024.

[20] Y. Li, C. Yang, A. Tzemanaki, A. Bahl, R. Persad, and C. Melhuish, "A dual-arm robotics teleoperation system for needle steering in prostate brachytherapy," in *2024 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2024, pp. 1–6.

[21] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, "Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5159–5166, 2022.

[22] D. Noh, H. Nam, K. Gillespie, Y. Liu, and D. Hong, "Yori: Autonomous cooking system utilizing a modular robotic kitchen and a dual-arm proprioceptive manipulator," *arXiv preprint arXiv:2405.11094*, 2024.

[23] A. Correia and L. A. Alexandre, "A survey of demonstration learning," 2023. [Online]. Available: https://arxiv.org/abs/2303.11191

[24] P. Aliasghari, M. Ghafurian, C. Nehaniv, and K. Dautenhahn, *Kinesthetic Teaching of a Robot over Multiple Sessions: Impacts on Speed and Success*, 02 2023, pp. 160–170.

[25] D. J. Brooks and H. A. Yanco, "Design of a haptic joystick for shared robot control," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, pp. 113–114.

[26] R. Pettitt, E. Redden, and C. Carstens, *Scalability of robotic controllers: An evaluation of controller options*, 01 2011, pp. 51–113.

[27] C. E. Mower, W. Merkt, A. Davies, and S. Vijayakumar, "Comparing alternate modes of teleoperation for constrained tasks," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1497–1504.

[28] S. Schaal, "Dynamic movement primitives-a framework for motor

control in humans and humanoid robotics," in *Adaptive motion of animals and machines*. Springer, 2006, pp. 261–280.
[29] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.
[30] A. Zeng, M. Attarian, brian ichter, K. M. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. S. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, and P. Florence, "Socratic models: Composing zero-shot multimodal reasoning with language," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=G2Q2Mh3avow
[31] A. Fabisch, "movement_primitives: Imitation learning of cartesian motion with movement primitives," *Journal of Open Source Software*, vol. 9, no. 97, p. 6695, 2024.
[32] D. Hanson, A. Imran, G. Morales, V. Krisciunas, A. Sagi, A. Malali, R. Mohbe, and R. Upadrashta, "Open arms: Open-source arms, hands & control," 05 2022.
[33] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: personalized robot assistance with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1087–1102, Dec 2023.
[34] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg, "Tidybot++: An open-source holonomic mobile manipulator for robot learning," in *Conference on Robot Learning*, 2024.
[35] C. Herron, A. Fuge, B. Beiter, Z. Fuge, N. Tremaroli, S. Welch, M. Stelmack, M. Kogelis, P. Hancock, I. Simoes, C. Runyon, I. Pressgrove, and A. Leonessa, "Pandora: The open-source, structurally elastic humanoid robot," 07 2024.
[36] P. Allgeuer, H. Farazi, M. Schreiber, and S. Behnke, "Child-sized 3d printed igus humanoid open platform," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 33–40.
[37] G. Metta, L. Natale, F. Nori, and G. Sandini, "The icub project: An open source platform for research in embodied cognition," in *Advanced Robotics and its Social Impacts*, 2011, pp. 24–26.
[38] G. Ficht, P. Allgeuer, H. Farazi, and S. Behnke, "Nimbro-op2: Grown-up 3d printed open humanoid platform for research," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE Press, 2017, p. 669–675. [Online]. Available: https://doi.org/10.1109/HUMANOIDS.2017.8246944
[39] G. Ficht, H. Farazi, D. Rodriguez, D. Pavlichenko, P. Allgeuer, A. Brandenburger, and S. Behnke, "Nimbro-op2x: Affordable adult-sized 3d-printed open-source humanoid robot for research," *International Journal of Humanoid Robotics*, vol. 17, no. 05, p. 2050021, 2020. [Online]. Available: https://doi.org/10.1142/S0219843620500218
[40] B. Johansson, T. A. Tjøstheim, and C. Balkenius, "Epi: An open humanoid platform for developmental robotics," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420911498, 2020. [Online]. Available: https://doi.org/10.1177/1729881420911498
[41] M. Lapeyre, P. Rouanet, J. Grizou, S. Nguyen, F. Depraetre, A. Le Falher, and P.-Y. Oudeyer, "Poppy project: open-source fabrication of 3d printed humanoid robot for science, education and art," in *Digital Intelligence 2014*, 2014, p. 6.
[42] P. Liu, D. Tateo, H. Bou-Ammar, and J. Peters, "Efficient and reactive planning for high speed robot air hockey," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. [Online]. Available: https://www.ias.informatik.tu-darmstadt.de/uploads/Team/PuzeLiu/IROS_2021_Air_Hockey.pdf
[43] C. C. Johnson, A. Clawson, and M. D. Killpack, "Baloo: A large-scale hybrid soft robotic torso for whole-arm manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2409.08420
[44] Y. Mu, T. Chen, S. Peng, Z. Chen, Z. Gao, Y. Zou, L. Lin, Z. Xie, and P. Luo, "Robotwin: Dual-arm robot benchmark with generative digital twins (early version)," 2024. [Online]. Available: https://arxiv.org/abs/2409.02920
[45] K. F. Gbagbe, M. A. Cabrera, A. Alabbas, O. Alyunes, A. Lykov, and D. Tsetserukou, "Bi-vla: Vision-language-action model-based system for bimanual robotic dexterous manipulations," *arXiv preprint arXiv:2405.06039*, 2024.
[46] M. H. Khan, S. Asfaw, D. Iarchuk, M. A. Cabrera, L. Moreno, I. Tokmurziyev, and D. Tsetserukou, "Shake-vla: Vision-language-action model-based system for bimanual robotic manipulations and liquid mixing," *arXiv preprint arXiv:2501.06919*, 2025.
[47] A. Dastider, H. Fang, and M. Lin, "Apex: Ambidextrous dual-arm robotic manipulation using collision-free generative diffusion models," *arXiv preprint arXiv:2404.02284*, 2024.
[48] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1970–1975.
[49] S. Jauhri, J. Peters, and G. Chalvatzaki, "Robot learning of mobile manipulation with reachability behavior priors," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8399–8406, 2022.
[50] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4057–4062.
[51] Kenta-Tanaka et al., "kinpy." [Online]. Available: https://github.com/neka-nat/kinpy
[52] C. E. Mower, "Spatial CasADi: A compact library for manipulating spatial transformations," 2023. [Online]. Available: https://github.com/cmower/spatial-casadi
[53] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.