Bridging Embodiment Gaps: Deploying Vision-Language-Action Models on Soft Robots

Haochen Su
EPFL
Lausanne, Switzerland
haochen.su@alumni.epfl.ch

Cristian Meo
LatentWorlds AI
TUDelft
Delft, Netherlands
cristianmeo@latentworlds.ai

Francesco Stella
Embodied AI SA
EPFL
Lausanne, Switzerland
f.stella@embodiedai.ch

Andrea Peirone
Embodied AI SA
EPFL
Lausanne, Switzerland
andree.peirone@gmail.com

Kai Junge
Embodied AI SA
EPFL
Lausanne, Switzerland
k.junge@embodiedai.ch

Josie Hughes
EPFL
Lausanne, Switzerland
josie.hughes@epfl.ch

Abstract

Robotic systems are increasingly expected to operate in human-centered, unstructured environments where safety, adaptability, and generalization are essential. Vision-Language-Action (VLA) models have been proposed as a language guided generalized control framework for real robots. However, their deployment has been limited to conventional serial link manipulators. Coupled by their rigidity and unpredictability of learning based control, the ability to safely interact with the environment is missing yet critical. In this work, we present the deployment of a VLA model on a soft continuum manipulator to demonstrate autonomous safe human-robot interaction. We present a structured finetuning and deployment pipeline evaluating two state-of-the-art VLA models (OpenVLA-OFT and π_0) across representative manipulation tasks, and show while out-of-the-box policies fail due to embodiment mismatch, through targeted finetuning the soft robot performs equally to the rigid counterpart. Our findings highlight the necessity of finetuning for bridging embodiment gaps, and demonstrate that coupling VLA models with soft robots enables safe and flexible embodied AI in human-shared environments.

1 Introduction

To deploy robots in human-centric, real-world settings, they must interpret human instructions, perceive dynamic environments, and execute robust actions. Vision-Language-Action (VLA) models unify perception, language understanding, and control within a single multimodal policy[23], offering a promising approach to these challenges. Encompassing CLIPort[29], SayCan[1], RT-2[6] and OpenVLA[20], VLA models have progressively improved generalization across tasks and settings. Yet, nearly all existing models and deployment focuses on rigid robotic arms, where predictable kinematics simplify control but limit safety and adaptability in human-centered environments.

Soft robots incorporate compliant or soft structures into their bodies, such that they deform in response to interactions with the environment. This makes them well suited for operating around humans as they provide intrinsic safety to the environment, are resilient to collisions, and can be robustness to environmentally uncertainty[27, 32]. Soft continuum manipulators, in particular, bring these benefits to manipulation[14, 9]. Currently soft arms rely on controllers that account for their underlying non-linear properties and redundancy within the structures. Deploying VLA models on such platforms remains unexplored: existing datasets and benchmarks overwhelmingly rely on rigid, serial-linked robots[11], leaving open questions about embodiment transfer.

This gap poses two key challenges. First, reliance on rigid embodiments restricts VLA applicability to domains where compliance is crucial. Second, the nonlinear, underactuated dynamics of soft robots raise doubts about whether policies trained on rigid arms can generalize effectively. Addressing this challenge is critical to deploying VLAs models on soft robot arms, combining their physical safety with the human-relevant capabilities of VLAs.

In this work, we take a step toward bridging this gap. We propose and implement a finetuning pipeline for deploying VLA models on a custom soft continuum robot, evaluating both OpenVLA-OFT[19] and π_0[5]. Our study systematically benchmarks embodiment transfer across rigid and soft robots and compares the relative strengths of two state-of-the-art VLA models. Concretely, our contributions are:

1. We introduce the first **open-source dataset of soft robot demonstrations**, enabling reproducible research on compliant embodiments.

2. We **benchmark OpenVLA-OFT on both rigid (UR5) and soft robots**, showing that finetuning closes the rigid-to-soft domain gap and yields comparable task success rates.

3. We **compare OpenVLA-OFT and π_0 on the soft robot**: while π_0 demonstrates stronger generalization on rigid embodiments, OpenVLA-OFT achieves superior performance on the compliant platform after finetuning.

2 Related Work

Vision-Language-Action (VLA) models unify perception, language, and control for robotic agents. Early approaches such as CLIPort [29] and SayCan [1] demonstrated the potential of pretrained vision-language models, while large-scale efforts like RT-1 [7] and RT-2 [6] improved task coverage on rigid manipulators. More recent methods focus on temporal reasoning and efficiency, including π_0 [5] with flow-based policies and OpenVLA-OFT [19] with parallel decoding and continuous outputs.

VLA models have also shown transfer between different rigid embodiments [11], suggesting a degree of generality. However, rigid robots share similar inverse kinematics and appearance, making transfer comparatively easier. In contrast, soft continuum manipulators exhibit nonlinear, underactuated dynamics and different morphology, a setting that remains unaddressed in prior VLA benchmarks[21][15]. Our work provides the first systematic evaluation of VLA models on a soft robotic arm. For more details about recent VLA methods, and state-of-the-art soft continuum robots' control, refer to Appendix A.

3 Methodology

To investigate the deployment of VLA models on soft robotic systems, we adopt a structured pipeline spanning task design, data collection, preprocessing, model adaptation, and evaluation. We begin by defining three representative manipulation tasks tailored to the soft robot's capabilities. Next, we set up a data-capturing environment to record multimodal demonstrations, which are then converted into standardized formats. Using these processed datasets, we finetune both models under comparable conditions. Finally, we perform inference and evaluate policy performance on the designed tasks, assessing both success rate and qualitative behavior.

3.1 Robot platforms

As a benchmark, the UR5 robot was used to perform manipulation tasks. A parallel gripper was mounted with a monocular camera mounted above for autonomy (see Fig.1D right).

For the soft counterpart (Fig.1D left), a custom designed continuum robot arm: Embuddy, is used shown in Fig.1A. Embuddy consists of three modular sections, comprised of a standard revolute joint followed by a soft continuum segment (see Fig.1B). The continuum segments (shown in detail in Fig.1C) are tendon driven, and bend in one plane (constrained by an incompressible centerline). The continuum structure is fabricated through 3D printed Thermoplastic Polyurethane (TPU). Two key features of Embuddy allows for inherently safe interactions. Firstly, the underactuated sections mean regardless of the motor positions the sections are always deformable to external forces. Secondly the arm is lightweight (total 5kg), limiting its inertial forces.

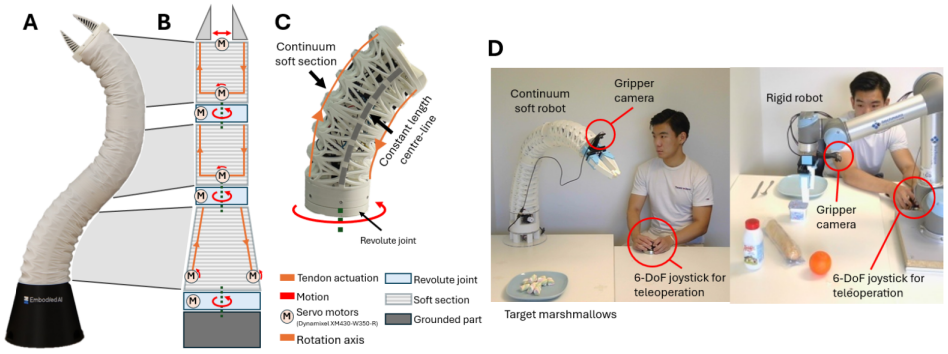


Figure 1: Continuum soft robot used for the experimental study. A: The full view of the continuum robot - Embuddy. B: Actuation and structural schematic of Embuddy, indicating tendons, joints, and motors. C: Detailed view of a single section. D: Demonstration setup for the soft and rigid robot.

Although Embuddy follows a similar scale to a standard serial-link manipulator, with a height of 1m, its workspace is limited to the bending angle of each soft section, whereby the first section can bend 80° and the second and the third up to 50° each. In our experiments, we use the same camera setup and gripper for both the UR5 and Embuddy, ensuring a fair comparison across embodiments.

3.2 Designed tasks

We selected two pick-and-place tasks and one close human-interactive task for the experiments. For simplicity, we denote them as task 1, 2 and 3.

- **Task 1:** "Put the orange in the plate" –> Simple pick-and-place
- **Task 2:** "Put the X in the plate"(X can be orange or milk) –> Pick-and-place with choices
- **Task 3:** "Feed the person with marshmallow" –> Close human-interactive

3.3 Experimental setup

Following the practices in OpenVLA-OFT[19] and π_0[5], we have both 3rd-person and wrist view cameras for capturing the scene. In each task, objects are randomly placed in the workspace. For more details on how the setup is done for each task, refer to Appendix B.

3.4 Data capturing and processing

To capture the dataset, we use a joystick to tele-operate the robots. To teleoperate and control the robot in cartesian space, a Piecewise Constant Curvature (PCC) model is used[28] for the inverse kinematics. By approximating every section as a constant curvature, the tendon lengths can be related to a modeled shape, which is used to determine the end-effector pose.

In each episode of each demonstration, the captured observation consists of 3rd-person image, wrist image, proprioceptive state(end-effector pose) and language instruction(the task). As shown in

Appendix B, the captured images are cropped and scaled. Following the practice of OpenVLA-OFT[19], we filter out the episodes when the robot has almost zero motion(for example when gripper is grabbing or releasing). Finally, we convert and pad the representations of state and action to desired ways and dimensions according to the configuration of the models. RLDS[26] format is used for OpenVLA-OFT and LeRobot[8] format is used for π_0[5]. We open source such datasets. For more details about how we do data capturing and processing for each models and tasks, refer to App. C.

3.5 Model finetuning and inference

For OpenVLA-OFT[19], due to the large number of parameters of the LLM backbone Llama 2 7B[33], the best practice that balances the accuracy and computational cost is to do full finetuing with the low-rank adaptation technique(LoRA)[18]. As for π_0, since the backbone VLM PaliGemma[4] has smaller number of parameters(3B), we conduct full finetuning. For more details, refer to App. D.

During inference, we use the same GPU that is used for finetuning for model prediction. On the local PC that is connected with the robot, we capture observations consisting of 3rd-person view image, wrist view image, proprio state and language instruction in real time. We send such observations to the remote, where the model predicts an action chunk based on the observation and sends the chunk back to local. The local executes the actions and captures observations again. We do this communication non-stop, until the task is done or it reaches maximum steps.

4 Results

In this section, we evaluate the performance and behavior of VLA models, more specifically OpenVLA-OFT[19] and π_0[5] on Embuddy with our designed tasks. Following the most common evaluation method, we estimate the model prediction accuracy by success rate in 10 trials. Our first experiment evaluates the performance of vanilla OpenVLA-OFT and π_0 on Embuddy, with particular interest in π_0, which is known for its stronger generalization capability. All out-of-the-box models without finetuning fail in our setting. As expected, the primary cause lies in the discrepancy between soft-robot and rigid-robot dynamics, specifically the mapping from end-effector pose to internal configurations. Due to the maximum bending angle constraints of each section, Embuddy consistently gets stuck mid-execution when the model generates motions suitable for rigid manipulators but incompatible with Embuddy's kinematics. This result highlights the significant domain gap between rigid and soft robots, underscoring the necessity of finetuning for effective policy transfer.

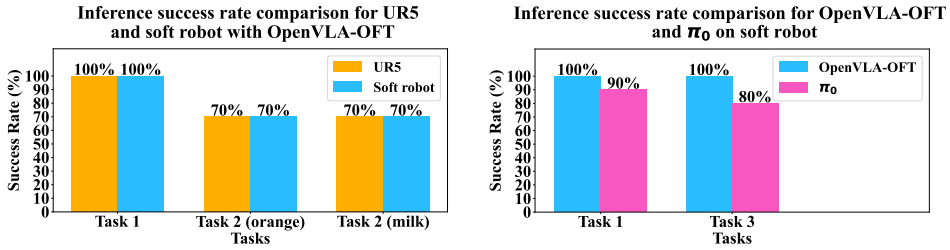


Figure 2: Inference success rate comparisons between OpenVLA-OFT and π_0 on UR5 and Soft Robot embodiments.

As shown in Figure 2 left, applying finetuned OpenVLA-OFT[19] on UR5 and Embuddy achieve exact same success rate on task 1 and 2. This demonstrates that our finetuning strategy successfully bridges the rigid-to-soft domain gap, enabling the models to achieve comparable performance on both soft and rigid robots. Not only OpenVLA-OFT works on soft robot after finetuning, so does π_0. As shown in figure 2 right, π_0 achieves high success rate, though slightly lower than OpenVLA-OFT on soft robot in task 1 and 3. It's notable that while π_0 has better generalization in rigid embodiments, OpenVLA-OFT outperforms π_0 when transferring to a completely new platform with totally different dynamics after proper finetuning. As shown in Table 1, even with a big connection delay, soft robot can still achieve at least 25 Hz in the control loop with OpenVLA-OFT and π_0. Appendix E shows visualizations and more details of our experiments during inference.

5 Conclusion

This paper presents the first systematic deployment of Vision-Language-Action models on a soft continuum robot, directly addressing the embodiment gap between compliant and rigid manipulators. Our experiments reveal that out-of-the-box VLA policies fail due to kinematic and dynamic mismatches. However, we demonstrate that a targeted finetuning pipeline using a small, custom dataset successfully bridges this gap. The adapted policies, particularly OpenVLA-OFT, achieve high success rates on the soft robot, comparable to a rigid UR5 baseline. This work confirms that the advanced reasoning of VLA models can be effectively combined with the intrinsic safety of soft robotics. This work shows a promising direction for developing safe, adaptable, and intelligent embodied agents for human-centered environments. Future research will expand this investigation to a wider range of tasks and compliant platforms.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[2] Carlo Alessi, Camilla Agabiti, Daniele Caradonna, Cecilia Laschi, Federico Renda, and Egidio Falotico. Rod models in continuum and soft robot control: a review. *arXiv preprint arXiv:2407.05886*, 2024.

[3] Costanza Armanini, Frédéric Boyer, Anup Teejo Mathew, Christian Duriez, and Federico Renda. Soft robots modeling: A structured overview. *IEEE Transactions on Robotics*, 39(3):1728–1748, 2023.

[4] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[8] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. `https://github.com/huggingface/lerobot`, 2024.

[9] Xiaoqian Chen, Xiang Zhang, Yiyong Huang, Lu Cao, and Jinguo Liu. A review of soft manipulator research, applications, and opportunities. *Journal of Field Robotics*, 39(3):281–311, 2022.

[10] Xingyu Chen, Jialei Shi, Helge Wurdemann, and Thomas George Thuruthel. Vision-based tip force estimation on a soft continuum robot. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7621–7627. IEEE, 2024.

[11] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

[12] Cosimo Della Santina, Antonio Bicchi, and Daniela Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5(2):1001–1008, 2020.

[13] Cosimo Della Santina, Ryan Landon Truby, and Daniela Rus. Data–driven disturbance observers for estimating external forces on soft robots. *IEEE Robotics and automation letters*, 5(4):5717–5724, 2020.

[14] Weiqiang Dou, Guoliang Zhong, Jinglin Cao, Zhun Shi, Bowen Peng, and Liangzhong Jiang. Soft robotic manipulators: Designs, actuation, stiffness tuning, and sensing. *Advanced Materials Technologies*, 6(9):2100018, 2021.

[15] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024.

[16] Thomas George Thuruthel, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5(2):149–163, 2018.

[17] Qinghua Guan, Francesco Stella, Cosimo Della Santina, Jinsong Leng, and Josie Hughes. Trimmed helicoids: an architectured soft structure yielding soft robots with high precision, large workspace, and compliant interactions. *npj Robotics*, 1(1):4, 2023.

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[19] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025.

[20] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024.

[21] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023.

[22] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2025.

[23] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai, 2025.

[24] Microsoft Corporation. Microsoft azure, 2024. Accessed: April 1, 2025.

[25] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[26] Sabela Ramos, Sertan Girgin, Léonard Hussenot, Damien Vincent, Hanna Yakubovich, Daniel Toyama, Anita Gergely, Piotr Stanczyk, Raphael Marinier, Jeremiah Harmsen, Olivier Pietquin, and Nikola Momchev. Rlds: an ecosystem to generate, share and use datasets in reinforcement learning, 2021.

[27] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[28] C. Della Santina, A. Bicchi, and D. Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5:1001–1008, April 2020.

[29] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation, 2021.

[30] Francesco Stella, Cosimo Della Santina, and Josie Hughes. Soft robot shape estimation with imus leveraging pcc kinematics for drift filtering. *IEEE Robotics and Automation Letters*, 9(2):1945–1952, 2023.

[31] Francesco Stella, Qinghua Guan, Cosimo Della Santina, and Josie Hughes. Piecewise affine curvature model: a reduced-order model for soft robot-environment interaction beyond pcc. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 1–7. IEEE, 2023.

[32] Francesco Stella and Josie Hughes. The science of soft robot design: A review of motivations, methods and enabling technologies. *Frontiers in Robotics and AI*, 9:1059026, 2023.

[33] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

A Related Work

In this section we introduce more about the two state-of-the-art methods π_0[5] and OpenVLA-OFT[19] we apply in our work.

A.1 π_0

π_0 [5] is a Vision-Language-Action (VLA) flow model built on top of a pretrained vision-language model (VLM) backbone PaliGemma[4]. It supports cross-embodiment learning by training on data from multiple robotic platforms with varying kinematics and action spaces. Key architectural and training aspects include:

- **Flow-matching action expert:** Rather than discretizing actions, π_0 uses conditional flow matching to predict continuous action chunks. During training, noisy action sequences are generated, and the model learns to predict the "denoising" flow that maps noise back to true actions.
- **Cross-embodiment generality:** π_0 is pretrained on diverse datasets comprising seven distinct robot configurations (e.g., single-arm, dual-arm, mobile manipulators) and over 68 manipulation tasks. This enables zero-shot control across different rigid platforms.
- **Action chunking for temporally extended tasks:** At inference, the model outputs sequences of actions via flow trajectories, allowing temporally coherent planning and execution of complex, extended tasks.
- **Pretraining and fine-tuning recipe:** π_0 adopts a two-stage training paradigm: broad pretraining on large-scale diverse robot data followed by task-specific fine-tuning, analogous to modern language-model training practices.

Together, these design choices enable π_0 to perform complex robotic manipulation tasks—such as laundry folding, object assembly, and mobile manipulation—via both direct prompting and fine-tuning, achieving strong generalization across embodiments and task domains.

A.2 OpenVLA-OFT

OpenVLA-OFT [19] is a recent state-of-the-art Vision-Language-Action model designed to improve both performance and inference efficiency over prior VLA systems. It builds upon the OpenVLA framework [20], using a ViT-based visual encoder and Llama 2 7B[33] as the language backbone, but introduces several key innovations:

- **Parallel decoding with action chunking:** Instead of autoregressive token-by-token prediction, OpenVLA-OFT maps multimodal inputs directly to a sequence of actions in a single forward pass. By conditioning on empty action embeddings of length K with bidirectional attention, the model predicts K consecutive actions simultaneously, enabling fast execution without intermediate replanning.
- **Continuous action outputs:** Unlike the discrete tokenized actions in OpenVLA, OpenVLA-OFT directly regresses continuous control vectors. An MLP action head replaces the output embedding layer, trained via an L1 objective to match ground-truth trajectories. This design improves precision and avoids discretization artifacts.
- **Flexible multimodal inputs:** Beyond single-view images, the model supports multi-camera observations and low-dimensional robot states. These embeddings are projected into the shared language space and concatenated for decoding, enabling richer context awareness.
- **Language-conditioned modulation:** To strengthen grounding, OpenVLA-OFT applies FiLM [25] layers that inject task-language embeddings into visual features at each transformer block, improving instruction following in visually ambiguous settings.

These modifications allow OpenVLA-OFT to outperform prior policies such as π_0[5] and diffusion-based RDT-1B[22] on benchmarks including LIBERO [21] and ALOHA [15], while maintaining competitive inference speed.

A.3 Soft robot control

A more conventional approach towards the control of soft continuum robots have been explored in the past[2, 16]. A key challenge lies in modeling and perception of soft robots due to their nature of large deformation[3]. While methods such as finite element analysis can describe its deformation, for real-time control, simplified mathematical models such as piecewise constant curvature (PCC)[12] or affine curvature models[31] have been developed. Through combination with proprioceptive sensing method(through tendon lengths[17], strain sensing[13], inertial measurement units[30], vision[10]), such models can be updated in real time to estimate and control its cartesian position.

B Experimental setup

Here we provide extra details about our experimental setup. Figure 3 shows the setup for UR5 baseline experiments. And figure 4 shows the setup for soft robot experiments. Note that the workplace of two setups have same area(1200 cm^3), but different shapes, due to the special workspace of Embuddy. For both robots, we use the same 1 DoF gripper. And the initial end-effector pose is fixed and predefined for all tasks.

B.1 Details for each tasks

Task 1: "Put the orange in the plate" There are four common food objects(orange, milk, yogurt and baguette) that are randomly placed in the workspace. The plate is placed apart, roughly at the same place in each demonstration.

Task 2: "Put the X in the plate"(X can be orange or milk) Same as Task 1.

Task 3: "Feed the person with marshmallow" A plate of marshmallows is placed randomly in the workspace. The person in the scene stays roughly at the same position in each demonstration.

C Dataset capturing and processing

To achieve real-time flexible 6 DoF controlling of both robots(UR5 and soft robot), we use a 3dconnexion space mouse as the joystick controller. The open/close of the gripper is controlled by the buttons on the joystick. Since we choose a relatively small gain for transformation and rotation, our capture frequency is also relatively low(5Hz). For all tasks, number of episodes in each demonstration is in the range of 50 to 200.

As shown in figure 3 and 4, we crop and down-sample the images to the resolution of 256 * 256. We further flip the wrist view image to make it more intuitive.

We represent the **proprioceptive state** (pose) as an 8-dimensional vector

$$s = \big[\, x,\, y,\, z,\, r,\, p,\, y,\, \text{pad},\, g \,\big],$$

where (x, y, z) denotes Cartesian position, (r, p, y) denotes orientation in roll–pitch–yaw, pad is a padding dimension, and $g \in \{0, 1\}$ denotes the gripper state (open/closed).

The corresponding **action** is defined as the delta between adjacent poses, represented as a 7-dimensional vector

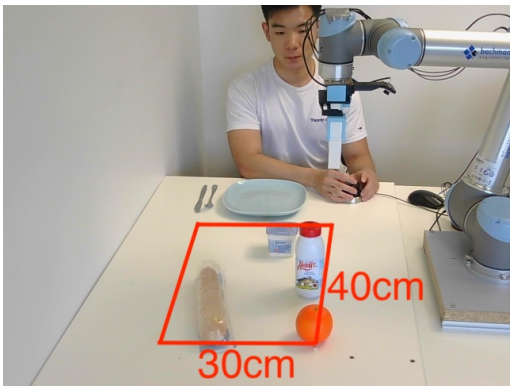
$$a = \big[\, \Delta x,\, \Delta y,\, \Delta z,\, \Delta r,\, \Delta p,\, \Delta y,\, g \,\big],$$

where the first six dimensions specify Cartesian and orientation increments, and $g \in \{0, 1\}$ indicates the gripper command.

Note that roll-pitch-yaw lies within the range of $[-\pi, \pi]$. When computing the delta, directly subtracting two values near the boundaries can lead to incorrect large values. For example, the difference between $-\pi + \epsilon$ and $\pi - \epsilon$ (with small ϵ) should be close to -2ϵ, but a naive subtraction yields nearly 2π. Therefore, the delta is handled by

$$\Delta = ((\Delta + \pi) \bmod 2\pi) - \pi \tag{1}$$

Here's the number of demonstrations captured for each task:



(a) 3rd-person view

(b) Wrist view

(c) 3rd-person view after cropping and scaling

(d) Wrist view after cropping, scaling, and flipping

Figure 3: Setup and processed image views for UR5 experiments(baseline). Top row: original views, with resolution 640x480; Bottom row: processed views, with resolution 256x256.

- Task 1: 50
- Task 2: 100 (50 for orange; 50 for milk)
- Task 3: 20

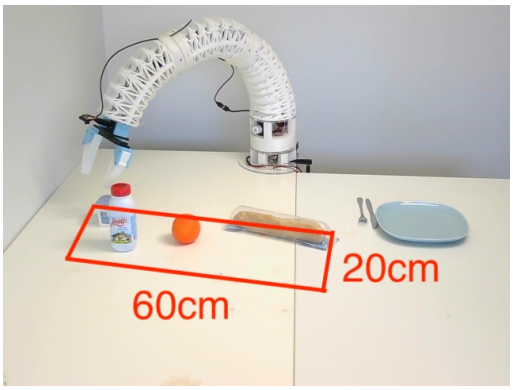
The open source soft robot dataset can be found on HuggingFace HCSuMoss/soft_orange and HCSuMoss/soft_feed .

D Finetuning details

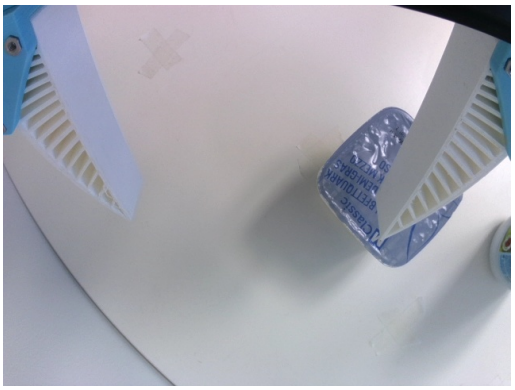
Figure 5 shows the training loss curves of task 1 and 3 with OpenVLA-OFT and π_0.

D.1 OpenVLA-OFT[19] finetuning

According to the studies in OpenVLA[20] paper, applying LoRA[18] with a rank of 32 is the best way to finetune on the OpenVLA-7b model, in terms of both prediction accuracy, and computational cost. To do the finetuning with such practice, we require a GPU with 80 GB(or more) memory. We utilize an A100 card on Virtual Machines of Microsoft Azure[24] cluster for UR5's experiments and an H100 card on a remote HPC cluster for soft robot's experiments.



(a) 3rd-person view



(b) Wrist view

(c) 3rd-person view after cropping and scaling

(d) Wrist view after cropping, scaling, and flipping

Figure 4: Setup and processed image views for soft robot experiments. Top row: original views, with resolution 640x480; Bottom row: processed views, with resolution 256x256.

By default data augmentation is applied on the input images, which includes augmentation with random cropping, adjustment on brightness, contrast, saturation, and hue. All the parameters are applied with default settings in the original work.

For hyper-parameters, we mostly follow the default setting of the model. We include proprio state(pose) and two image views(3rd-person and wrist) in the input, and train continuous action head with L1 regression objective with LoRA(rank=32). The model was trained with the following hyperparameter settings:

- **Action Chunk:** 8
- **Batch size:** 8 with one device
- **Learning rate:** 5×10^{-4}
- **Warm-up steps:** No warm-up
- **Learning rate decay:** After 120,000 steps, the learning rate decayed by a factor of 10.
- **Gradient accumulation:** Gradients were accumulated for 1 step, effectively applying updates at every step.
- **Maximum training steps:** The training process was run for a total of 200,000 steps.
- **GPU memory allocated** 63 GB

For task 2 "Put the X in the plate", we enable the FiLM module to enhance language understanding, so that the model is capable to handle the task for both orange and milk. Due to this modification and larger dataset size, we increase the maximum training steps to 240k and adjust the learning rate decay to happen at step 180k for task 2.

And for task 3, due to a smaller amount of demonstrations included in the dataset, we reduce the maximum training steps to 150k and adjust the learning rate decay to at step 100k.

When the training loss is stabilised around 0.01, the training is done. For updating 150k steps on a single A100 card, it takes around 56 hours.

D.2 π_0[5] finetuning

Since the backbone of π_0 is much smaller than OpenVLA-OFT[19], to make fair comparison, we use the full finetuning recipe for our experiments. We utilize the same H100 card on a remote HPC cluster as the experiments of OpenVLA-OFT's experiments on soft robot.

Once again, we follow the default setting and hyper-parameters of the model. To make the action chunk size same as previous experiments, we modify the action chunk size to be 8.

- **Action Chunk:** 8
- **Batch size:** 32 with one device
- **Learning rate:** 2.5×10^{-5}
- **Warm-up steps:** 1000
- **Learning rate decay:** Cosine decay from warm-up to maximun training step. The final LR at the end of decay is 2.5×10^{-6}
- **Gradient accumulation:** Gradients were accumulated for 1 step, effectively applying updates at every step.
- **Maximum training steps:** The training process was run for a total of 30,000 steps.
- **Number of workers:** 2
- **GPU memory allocated** 91 GB (XLA_PYTHON_CLIENT_MEM_FRACTION=0.9 –> this enables JAX to use up to 90% of the GPU memory)

For both task 1 and 3, we run the same amount of steps. It takes around 11 hours on an H100 to update 30k steps.

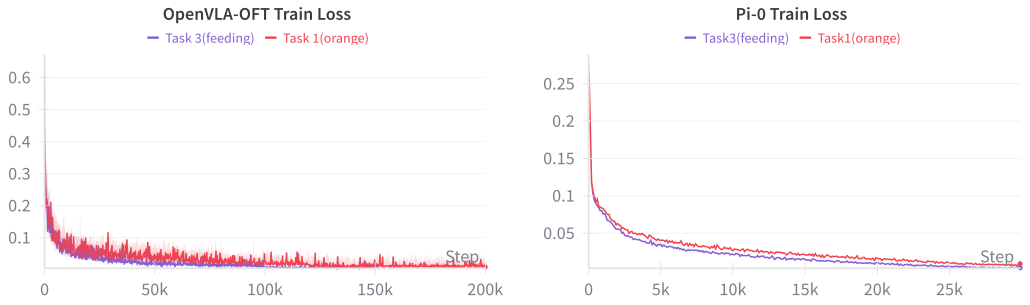


Figure 5: The training losses for task 1 and 3 on soft robot with OpenVLA-OFT and π_0

E Inference

Figure 6 and 7 and show visualizations of our UR5 experiments and soft robot experiments on different tasks.

E.1 Inference perturbation

Apart from the experiments we show in the paper, we also study about other conditions that may happen during inference in practice.

E.1.1 With human showing in the scene

For both experiments on UR5 and soft robot, we involve human moving freely in the scene during inference. It turns out that it has no influence on the model's performance and the model has its focus on the workspace. What's more, human can also be involved in the training set, and this is verified by both UR5 and soft robot experiments. The movement and appearance of human has zero influence in the scene as long as the workspace is not covered or interrupted. These results confirm the strong robustness of VLA models to human presence, ensuring reliable performance in human-shared environments.

E.1.2 With unseen objects

When there are some objects that are not shown in the training set, the model might be confused by chance. In our experiments, the model is confused once in 10 trials.

E.1.3 When object is placed outside of workspace

When the tasked object is placed outside of workspace, the model fails all the time, even if the object is only placed slightly(10cm) away from the region. From this test, we find that the workspace occurred in the training set is a deterministic factor, which defines the region of where the inference may succeed.

E.2 Verification of Language Instruction

We evaluate whether the models correctly ground their actions in the provided language instructions.

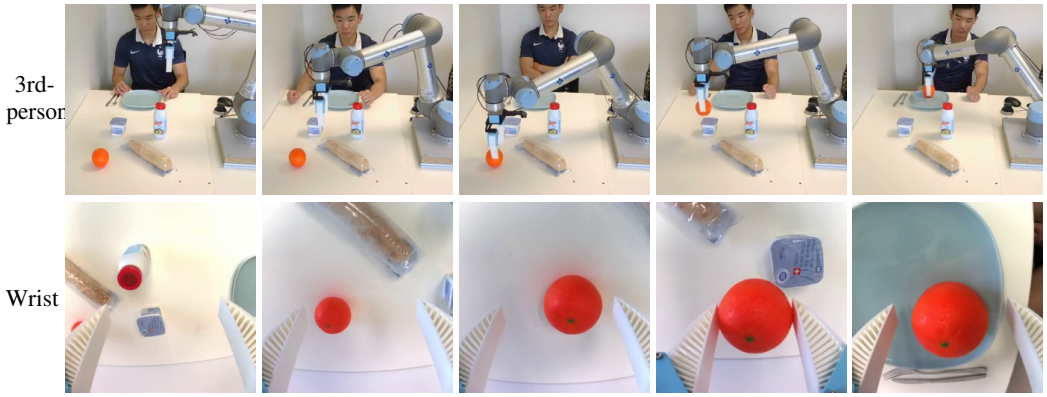
Task 2 (Pick-and-Place with Choices): In the "Put X in the plate" task, OpenVLA-OFT[19] achieves a 70% success rate. The inclusion of the FiLM[25] module directs the model's attention to the object specified in the instruction, rather than selecting objects arbitrarily, indicating effective language-conditioned object selection.

Task 3 (Human-Interactive Feeding): In a controlled modification, we place an orange in the plate instead of marshmallows. The models(OpenVLA-OFT and π_0[5]) appropriately refrain from executing the pick-and-place action, terminating the task mid-execution rather than incorrectly interacting with the available object. This confirms that the models' actions are semantically guided by the instruction rather than by visual salience alone.

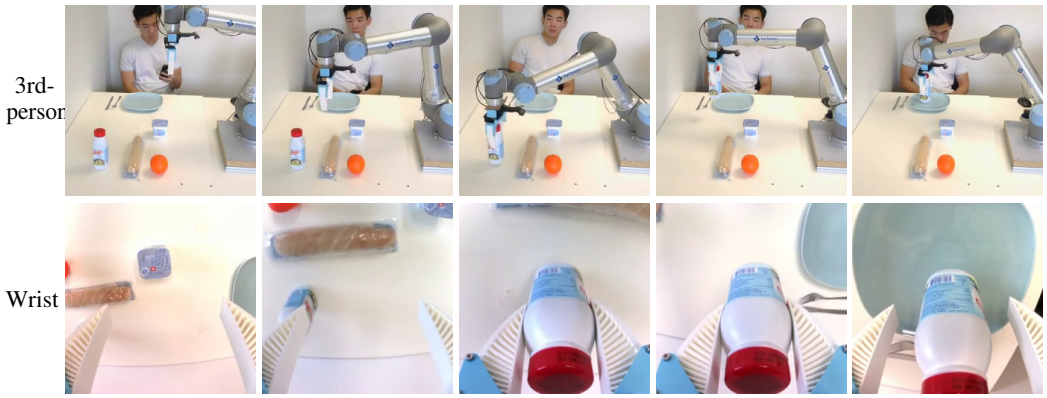
These results collectively demonstrate that OpenVLA-OFT and π_0 reliably interprets and adheres to task-specific language instructions, supporting its robustness in language-conditioned manipulation tasks.

E.3 Robustness of Embuddy against manual movement

As one of the main advantages of soft robot in close human-interactive task is that it's bendable and easily stoppable by person. We also study the behavior of Embuddy in the VLA control loop when a person manually stops it or pushes it away. As shown in Figure 8, when Embuddy is manually pushed away during OpenVLA-OFT's inference stage, it can recover its original pose, continue to follow the correct trajectory and finish the task successfully without influence. In our experiments with task 3, the whole process of one trial lasts around 2 to 3 minutes. We manually stop or push away Embuddy twice, each lasts for around 5 seconds. Under such perturbation, we observe no performance degradation.



(a) Inference for task 1 "put the orange in the plate" on UR5

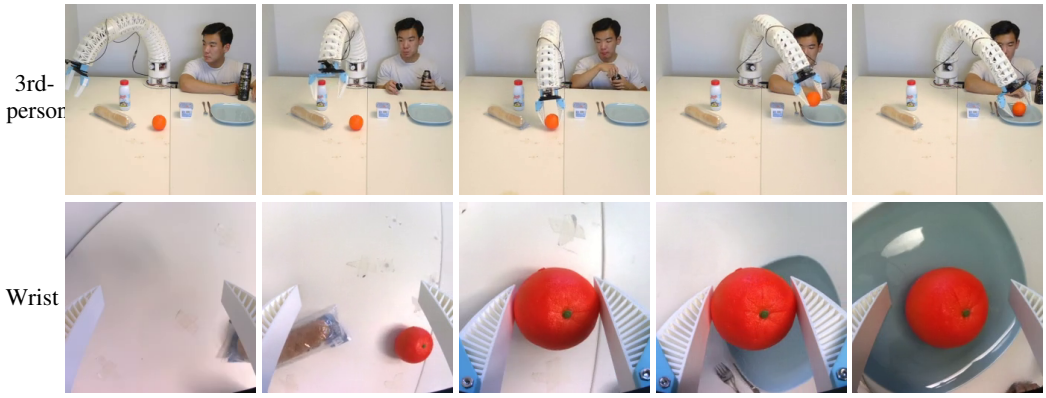


(b) Inference for task 2 "put the X in the plate" on UR5(when X is milk)

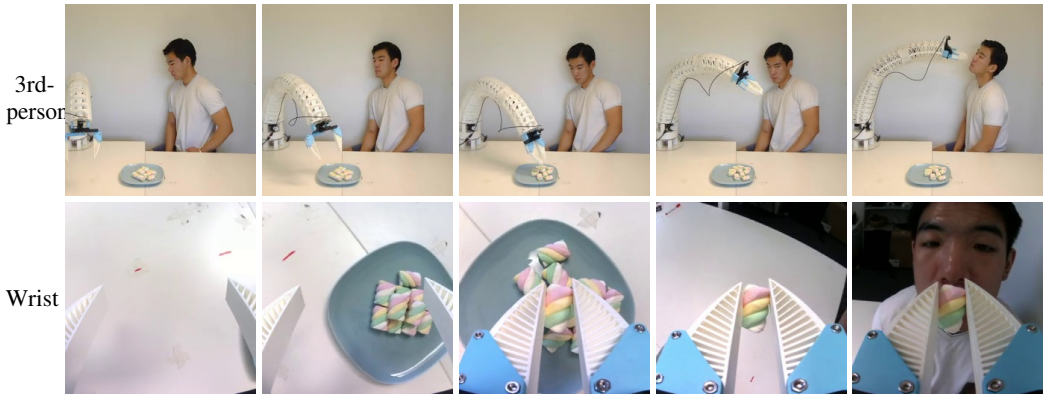
Figure 6: Visualization of the UR5 performing task 1 and 2 during inference. Each row shows 3rd-person or wrist camera views, and columns show different time steps.

Table 1: Table for inference average off-board frequency(network communication latency included). Note that within soft robot experiments, the communication latency is the same. But their latency is higher than the one in UR5 experiments. All the inference uses action chunk of size 8.

Platform	Model	Device	Frequency (Hz)
UR5	OpenVLA-OFT	A100(Azure VM)	32.3
Embuddy	OpenVLA-OFT	H100(Remote cluster)	25.1
Embuddy	π_0	H100(Remote cluster)	38.0



(a) Inference for task 1 "put the orange in the plate" on soft robot



(b) Inference for task 3 "feed the person with marshmallow" on soft robot

Figure 7: Visualization of Embuddy performing task 1 and 3 during inference. Each row shows 3rd-person or wrist camera views, and columns show different time steps.

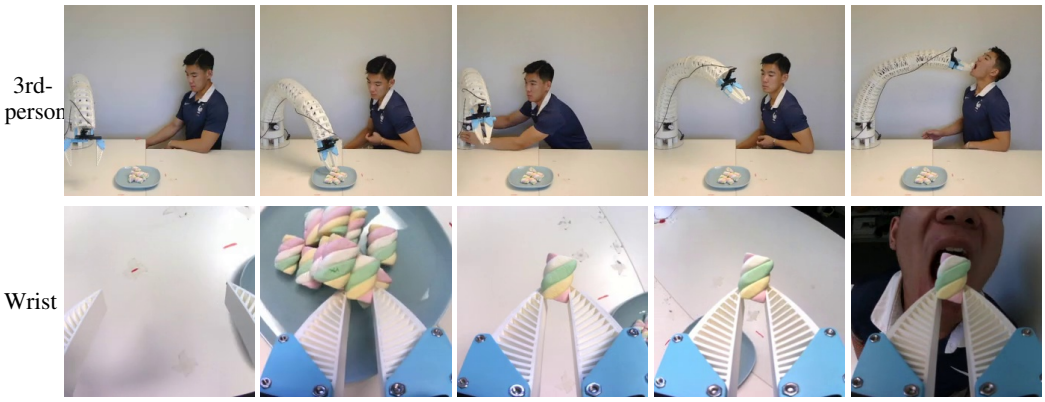


Figure 8: Visualization of Embuddy performing task 3 during inference with human interaction. As shown in the third moment, the robot's pose is manually changed by human force in the middle of inference. However, Embuddy is capable to recover it's pose and trajectory, and still complete the task successfully.