OPEN TEACH: A Versatile Teleoperation System for Robotic Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Open-sourced, user-friendly tools form the bedrock of scientific advancement across disciplines. The widespread adoption of data-driven learning has led to remarkable progress in multi-fingered dexterity, bimanual manipulation, and applications ranging from logistics to home robotics. However, existing data collection platforms are often proprietary, costly, or tailored to specific robotic morphologies. We present OPEN TEACH, a new teleoperation system leveraging VR headsets to immerse users in mixed reality for intuitive robot control. Built on the affordable Meta Quest 3, which costs $500, OPEN TEACH enables real-time control of various robots, including multi-fingered hands, bimanual arms, and mobile manipulators, through an easy-to-use app. Using natural hand gestures and movements, users can manipulate robots at up to 90Hz with smooth visual feedback and interface widgets offering closeup environment views.

Keywords: Low Cost Teleoperation, Data Collection, Imitation learning

1 Introduction

The integration of learning-based methods has sparked a revolution in robotics, significantly enhancing capabilities in manipulation [1, 2, 3, 4], locomotion [5, 6, 7, 8], and aerial robotics [9, 10, 11]. More recent work has been making advancements in complex single-task behavior learning [12, 13, 2], multitask scenarios [14, 15], multimodal behavior learning [16, 17, 18, 19], and efficient fine-tuning of pretrained behavior models [20, 21, 22]. A fundamental requirement across all these threads of research is the need to collect data in the form of task demonstrations.

Commonly used teleoperation systems include devices such as joysticks and 3D spacemouses [23, 24], commercial VR headsets [25, 26, 27, 13, 28, 29], kinesthetic teaching [30], and phone teleoperation [31]. Recently proposed exoskeleton-based teleoperation frameworks like ALOHA [2], GELLO [32], and AirExo [33] attempt to alleviate this problem by having the human teleoperator directly control a kinematically isomorphic version of the same robot arm. These frameworks directly impose the kinematic constraints of the robot arm during teleoperation making it more compatible and intuitive to control the robot. Although highly effective, these systems can require an additional robot for each robot being controlled, have high initial setup costs, and are designed for specific robot morphologies.The challenge of easy-to-use teleoperation devices is more apparent in dexterous manipulation problems [34, 35, 27, 13], owing to the high dimensional action space. Such frameworks typically involve the use of expensive gloves [36, 37, 38], extensive calibration processes [34, 27], or are susceptible to monocular occlusions [27].

In this work, we present OPEN TEACH, an open-source framework for robot teleoperation that supports a variety of robots, including bimanual and multi-finger manipulation, all at a price of $500. OPEN TEACH uses a VR headset (e.g. Quest 3) to put users / teachers in an immersive virtual world where they can view a robotic scene both through their eyes, via visual passthrough, as well as realtime streams from the robot's cameras. To control the robot, users can simply use hand gestures, which are detected using onboard hand-pose estimators at 90Hz. We experimentally evaluate OPEN TEACH on 38 tasks across single arm, bimanual, multi-fingered, and mobile manipulation robot setups in both simulation and the real world. The tasks span from tabletop manipulation to contact-rich

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

dexterous manipulation. cross different robot morphologies, we find that users can provide demonstrations at speeds on par with robot-specific teleoperation systems and significantly faster than general-purpose systems like AnyTeleop [35]. Importantly, policies trained on the data collected achieve an average success rate of 86% on 10 tasks in simulation and the real world, validating the utility of policy learning using OPEN TEACH. The contributions of this work is summarized as follows:

1. We present OPEN TEACH, an open-source system for plug-and-play teleoperation framework suitable for collecting demonstrations across different robot morphologies in both simulation and the real world.
2. We experimentally show that the demonstrations collected by OPEN TEACH can be used to train effective, general-purpose manipulation behaviors.
3. Our user study on 15 new users highlights the efficacy of OPEN TEACH for both experienced and new users.

OPEN TEACH will be fully open-sourced with mixed reality API, policy training code, and demonstrations collected using OPEN TEACH available at `https://anon-open-teach.github.io/`.

2 OPEN TEACH

In OPEN TEACH, a user wears a Virtual Reality (VR) headset to provide demonstrations to a robot. This involves creating a virtual world for teaching, retargeting the teacher's hand and wrist pose to the robot joints, and finally controlling the robot. We compare OPEN TEACH with various other teleoperation systems across a variety of robot types and observe that OPEN TEACH is the only framework that enables controlling multiple arms, hands, and mobile manipulators, is calibration-free, and is completely open-source.

2.1 Placing an Operator in a Virtual World

We use the Meta Quest 3 VR headset to place the human teacher in a virtual world. The headset surrounds the human in a virtual environment at a resolution of 2064×2208 and a native refresh rate of 90Hz. The base version of this headset is affordable at $499 and is relatively light at 513g. Compared to the Meta Quest 2 VR headset used in prior work [13], the Quest 3 provides a full-color passthrough allowing the human to get a direct view of the robot setup during teleoperation. These features, especially the full-color passthrough, allow for a comfortable and intuitive operation by the user. Additionally, similar to Arunachalam et al. [13], the Quest 3 API interface allows for creating custom mixed reality worlds that visualize the robotic system along with diagnostic panels in VR. It is important to highlight the exceptional clarity of the scene passthrough visible in Quest 3.

2.2 Pose Estimation with VR Headsets

Similar to Arunachalam et al. [13], we directly use the in-built hand pose estimator [39] of the Quest 3 using 2 monochrome cameras. This is significantly more robust compared to single camera alternatives [40]. Further, since the cameras are internally calibrated, they do not require specialized calibration routines that are needed in prior multi-camera teleoperation frameworks [34, 35]. Also, since the hand-pose estimator is integrated into the device, it can stream real-time hand poses at 90Hz.

2.3 Human to Robot Pose Retargeting

The inbuilt hand pose estimate from the VR headset provides us with the joint positions of all the fingers of the human hand and the wrist. With this information, we can design wrappers that use combinations of these joint positions to map the human hand poses to the robot poses for any given robot morphology. In this work, we use a variety of robot arms, each with either a 2-fingered gripper or a multi-fingered robot hand.

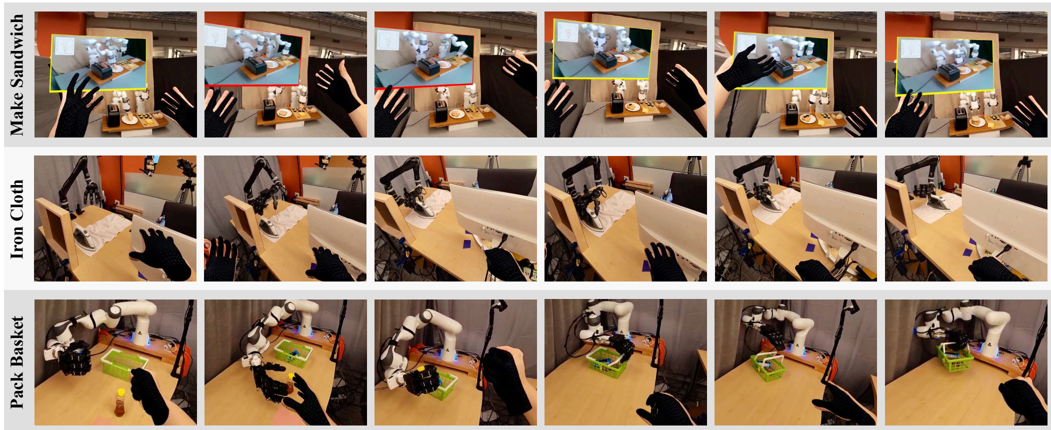


Figure 1: The demonstration collection process as viewed from within the VR application. Shown here is one task being performed for each real-world setup. High resolution images streamed at 90 Hz to the VR application allow for an immersive experience and enable reactive control by the user.

Robot Arm: We establish a 3D coordinate system using the wrist keypoint and knuckle points of the index and pinky fingers to define a 2D plane along the palm and a perpendicular third axis. The wrist position maps to the robot end effector position. Changes in the orientation of this hand coordinate system over time map to adjustments in end effector orientation.

Robot Hand: We use the teacher's hand pose obtained from the VR to compute the individual joint angles in the teacher's hand. Given these joint angles, a straightforward method of retargeting is to directly command the robot's joints to the corresponding angles. In practice, this works well for all fingers except the thumb. To address this, we improve upon Arunachalam et al. [13], where the spatial coordinate of the teacher's thumb tip is mapped to that of the robot hand and then an inverse kinematics solver is used to compute the joint angles of the thumb.

Two-fingered gripper: To detect the opening and closing of the two-fingered gripper, we utilize the pinch between the pinky finger and the thumb. We use a toggle mechanism for opening and closing the gripper where each pinch indicates toggling to the alternate state of the gripper.

Mobile manipulator: The same 3D coordinate system established for controlling robot arms is used for mapping the wrist's movements to actions of the mobile robot. When the wrist moves forward, it extends the robot's arm, enabling it to reach farther. Vertical wrist movements adjust the robot's height, while lateral wrist movements cause the robot to move sideways by controlling its wheels.

3 Experiments

We demonstrate the usefulness of the collected data by training visual and visuotactile policies using behavior cloning [41] and inverse RL [42, 43].

3.1 Imitation Learning with OPEN TEACH Data

Here, we describe the algorithms used for learning policies on data collected through OPEN TEACH.

1. **Franka-Allegro:** We record both visual and tactile data for this setup. The policies are trained using TAVI [44], a demonstration-guided residual RL algorithm that collects a few expert demonstrations and learns a robot policy using both visual and tactile data.
2. **Allegro Sim:** We only record visual data for this setup and train policies using FISH [21].
3. **LIBERO Sim [23]:** We only record visual data for this setup. The policies are trained using transformer-based BC with a GMM head [45] and action chunking [2].

Table 1: Teleoperation Frequency across all robots.

Domain	Robot Setup	Stream Frequency (in Hz)	
		Arm	End Effector
Real	Franka-Allegro	60	60
	Kinova-Allegro	60	60
	Bimanual	90	90
	Stretch	5	5
Sim	Allegro Sim	60	60
	LIBERO Sim	20	20

Table 2: Performance of policies learned on data collected through OPEN TEACH. For Franka-Allegro, Allegro Sim, and Libero Sim, TAVI [44], FISH [21] and BC were respectively used to train policies.

Robot Setup	Task	Number of Demos	Success Rate
Franka-Allegro	Open Box	3	9/10
	Grasp Sponge	6	7/10
	Pick Up Tea Sachet	4	7/10
	Grasp Object and Twist	6	8/10
Allegro Sim	Flip Cube	6	10/10
	Flip Sponge	6	10/10
	Pinch Grasp	6	7/10
Libero Sim	Close Top Drawer of Cabinet	10	10/10
	Turn on Stove	10	9/10
	Pick and Place Soup into Basket	50	9/10

Table 3: User study comparing OPEN TEACH with baselines when used by experts and new users.

Task	Success Rate				Median completion time for successful demonstrations (in s)			
	New User			Expert	New User			Expert
	Holo-Dex	AnyTeleop	Open Teach	Open Teach	Holo-Dex	AnyTeleop	Open Teach	Open Teach
Flip cube	1	1	1	1	6.58	13.71	5.5	2.85
Pinch Grasp	0	0.2	0.8	1	17.49	18.94	18.72	3.71
Pour	N/A	N/A	0.4	0.8	N/A	N/A	40.97	14.83
Pick and Place	N/A	N/A	0.8	0.8	N/A	N/A	23.57	11.875
Open box of mints	N/A	N/A	0.5	1	N/A	N/A	32.21	20.45

3.2 How versatile is OPEN TEACH across robotic setups?

The primary idea behind OPEN TEACH is that given any robotic setup, a user can purchase an affordable off-the-shelf VR headset (in this case, Quest 3) and plug the headset and robot setup into the proposed framework to start teleoperating the robot without any additional hardware setup cost. To investigate its versatility, we use OPEN TEACH to teleoperate four different real world robotic setups, each having a different combination of a robot arm and end effector type — Franka Allegro, Kinova Allegro, a Bimanual setup with 2 xArm7 robots, and Hello Stretch for mobile manipulation. We also exhibit the applicability of OPEN TEACH in simulation through evaluations on 2 simulated environment suites — Allegro Sim and LIBERO Sim [23]. The frequency of teleoperation for each of the setups has been provided in Table 1.

3.3 How successful are policies trained with OPEN TEACH?

Table 2 provides the success rates of policies learned using imitation learning across both the real-world and simulated setups. We use TAVI [44] to learn visuotactile policies on Franka-Allegro, and FISH [21] to learn visual policies on Allegro Sim. Similar to prior work [44, 21], these policies were learned within 20 minutes and achieved an average success rate of 82%, validating the high quality of the collected observation data. Behavior cloning policies on LIBERO Sim achieve an average success rate of 93%, confirming the high quality of the collected action data. Overall, the learned policies achieve an average success rate of 86% across all tasks and robot morphologies.

4 Conclusion

In this work, we introduce OPEN TEACH, an open-source unified framework designed to facilitate low-latency, high-frequency robot teleoperation. This versatile framework is tailored to accommodate diverse tasks and is compatible with a range of robot morphologies. However, we recognize a few limitations in this work: (a) OPEN TEACH relies on the accuracy of the in-built hand pose detection in the VR headset. Inaccuracies, particularly when fingers are occluded from view, can diminish the quality of hand tracking, posing challenges to teleoperation. (b) In specific instances, the pose detector on the Oculus board may misconstrue finger positions, leading to difficulties in executing gestures like gripper closing, which relies on precise pinches between fingers. Addressing these challenges through future research on hand pose detection and tracking holds the potential to enhance the ease and intuitiveness of teleoperation using VR headsets.

References

[1] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song. Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects. *arXiv preprint arXiv:2203.00663*, 2022.

[2] 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.

[3] 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.

[4] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[5] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Transactions on Robotics*, 2022.

[6] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robotics and Automation Letters*, 7(2):2377–2384, 2022. doi:10.1109/LRA.2022.3143567.

[7] L. Smith, I. Kostrikov, and S. Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*, 2022.

[8] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[9] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 528–535. IEEE, 2016.

[10] D. Gandhi, L. Pinto, and A. Gupta. Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955. IEEE, 2017.

[11] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter. Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, 2(4):2096–2103, 2017. doi:10.1109/LRA.2017.2720851.

[12] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[13] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023.

[14] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[15] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *arXiv preprint arXiv:2309.01918*, 2023.

[16] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[17] Z. J. Cui, Y. Wang, N. Muhammad, L. Pinto, et al. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[18] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[19] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.

[20] S. Haldar, V. Mathur, D. Yarats, and L. Pinto. Watch and match: Supercharging imitation with regularized optimal transport. *arXiv preprint arXiv:2206.15469*, 2022.

[21] S. Haldar, J. Pari, A. Rai, and L. Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *arXiv preprint arXiv:2303.01497*, 2023.

[22] A. Nair, A. Gupta, M. Dalal, and S. Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[23] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[24] N. E. Sian, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanie. Whole body teleoperation of a humanoid robot development of a simple master device using joysticks. *Journal of the Robotics Society of Japan*, 22(4):519–527, 2004.

[25] Z. Gharaybeh, H. Chizeck, and A. Stewart. Telerobotic control in virtual reality. In *OCEANS 2019 MTS/IEEE SEATTLE*, pages 1–8, 2019. doi:10.23919/OCEANS40490.2019.8962616.

[26] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *ICRA*, 2018.

[27] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. *arXiv preprint arXiv:2203.13251*, 2022.

[28] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training, 2022. URL `https://arxiv.org/abs/2210.03109`.

[29] A. George, A. Bartsch, and A. B. Farimani. Openvr: Teleoperation for manipulation. *arXiv preprint arXiv:2305.09765*, 2023.

[30] A. G. Billard, S. Calinon, and F. Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54(5):370–384, 2006.

[31] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[32] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. *arXiv preprint arXiv:2309.13037*, 2023.

[33] H. Fang, H.-S. Fang, Y. Wang, J. Ren, J. Chen, R. Zhang, W. Wang, and C. Lu. Low-cost exoskeletons for learning whole-arm manipulation in the wild. *arXiv preprint arXiv:2309.14975*, 2023.

[34] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170, 2020. doi:10.1109/ICRA40945.2020.9197124.

[35] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023.

[36] M. Caeiro-Rodríguez, I. Otero-González, F. A. Mikic-Fonte, and M. Llamas-Nistal. A systematic review of commercial smart gloves: Current status and applications. *Sensors*, 2021. ISSN 1424-8220. doi:10.3390/s21082667.

[37] V. Kumar and E. Todorov. Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 657–663, 2015. doi:10.1109/HUMANOIDS.2015.7363441.

[38] S. Li, J. Jiang, P. Ruppel, H. Liang, X. Ma, N. Hendrich, F. Sun, and J. Zhang. A mobile robot hand-arm teleoperation system by vision and imu. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10900–10906. IEEE, 2020.

[39] S. Han, B. Liu, R. Cabezas, C. D. Twigg, P. Zhang, J. Petkau, T.-H. Yu, C.-J. Tai, M. Akbay, Z. Wang, A. Nitzan, G. Dong, Y. Ye, L. Tao, C. Wan, and R. Wang. Megatrack: Monochrome egocentric articulated hand-tracking for virtual reality. 2020.

[40] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann. Mediapipe hands: On-device real-time hand tracking, 2020.

[41] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *NeurIPS*, volume 1. Morgan-Kaufmann, 1988.

[42] A. Y. Ng, S. J. Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, 2000.

[43] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

[44] I. Guzey, Y. Dai, B. Evans, S. Chintala, and L. Pinto. See to touch: Learning tactile dexterity through visual incentives. *arXiv preprint arXiv:2309.12300*, 2023.

[45] D. A. Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[46] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, and J. Zhang. Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 416–422. IEEE, 2019.

[47] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube, 2022.

[48] Y. Qin, H. Su, and X. Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. *arXiv preprint arXiv:2204.12490*, 2022.

[49] S. Li, N. Hendrich, H. Liang, P. Ruppel, C. Zhang, and J. Zhang. A dexterous hand-arm teleoperation system based on hand pose estimation and active vision. *IEEE Transactions on Cybernetics*, 2022.

[50] M. Mosbach, K. Moraw, and S. Behnke. Accelerating interactive human-like manipulation learning with gpu-based simulation and high-quality demonstrations. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 435–441. IEEE, 2022.

[51] 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.

5 Appendix

Figure 2: We present OPEN TEACH, a unified robot teleoperation framework that supports multiple arms and hands, allows mobile manipulation, is calibration-free, and works across both simulation and real-world environments. OPEN TEACH uses a VR headset for teleoperation, offers low latency and high-frequency visual feedback. This high-frequency operation allows human users to correct for robot errors in real time, facilitating the execution of intricate and long-horizon tasks. From *making a sandwich* and *ironing cloth* to *placing items in a basket and lifting it* and *approaching a cabinet and opening it*, OPEN TEACH delivers a comprehensive, user-friendly teleoperation experience for a wide range of applications. OPEN TEACH is fully open-source.

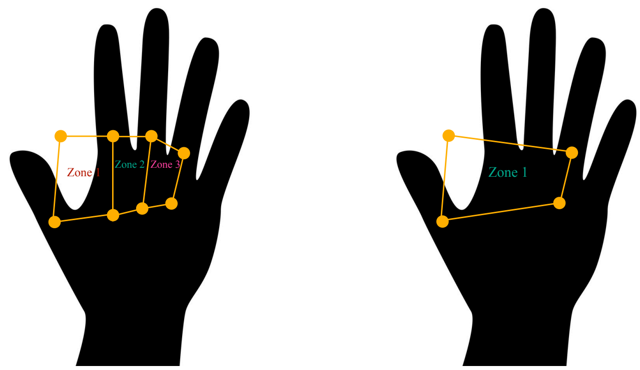
5.1 Framework details

5.1.1 Structure of the framework

We use ZeroMQ for networking between nodes. The OPEN TEACH framework is divided into two parts - *teleoperation* and *data collection*.

Teleoperation: The teleoperator is divided into 5 components - Detector, Keypoint Transformer, Operator, Controller, and Visualizer. A brief description of each has been provided below.

1. **Detector:** Receives the hand keypoints from the Meta Quest 3 and publishes them to ZMQ sockets.
2. **Keypoint Transformer:** Subscribes the keypoints published by the detector and maps them to the robot pose.
3. **Operator:** Receives the robot pose from the keypoint transformer and the current robot state from the controller. The operator computes the robot's actions which are published to a ZMQ socket.



(a) Holo-Dex (b) Open Teach

Figure 3: Thumb retargeting difference

4. **Controller:** Subscribes an action from the operator and takes an action in the real or simulated environment. After taking the action, the controller publishes the current states of the environment for use by the operator.

5. **Visualizer:** Subscribes the RGB images from the camera process (or the environment in case of simulations) and puts it on the screen inside the VR app for visualization during teleoperation.

Data Collection: A data recorder process subscribes sensor information (RGB and Depth images, tactile readings, timestamps) and robot-specific information (joint states, gripper states, timestamps) from the corresponding sockets and logs them in corresponding files. The data is then compiled together by matching the timestamps between the sensor information and robot-specific data.

5.1.2 Thumb Retargeting for Robot Hand

Section 2.3 provides details about the design of the OPEN TEACH wrapper for the robot hand. To recap, given the individual joint angles in the teacher's hand from the VR headset, the joint angles for the robot hand can be computed by directly commanding the robot's joints to the corresponding angles. This works well for all fingers except the thumb. Holo-Dex[13] deals with this by mapping the spatial coordinate of the teacher's thumb tip to that of the robot hand. Then an inverse kinematics solver is used to compute the joint angles of the thumb. In this case, the retargeting of the thumb is done in 2D space. These bounds, depicted in Fig. 3(a), define the thumb's reach limits. During retargeting, the thumb tip's zone on the 2D palm plane is detected, and a perspective transform from the human hand to the robot hand is applied, aligning the human thumb tip with the robot thumb tip on the 2D plane. However, using three separate bounds introduces jitters when the thumb tip transitions between zones and results in stagnancy when outside the bounds. Further, in Holo-Dex, the height of the robot thumb tip is fixed, allowing it to only move along the 2D space.
To address these challenges, OPEN TEACH employs a single, large zone spanning the entire thumb's workspace in 2D space(refer to Fig. 3(b)). When the thumb is within bounds, a perspective transformation retargets the human thumb tip to the robot thumb tip. In cases where the thumb goes out of bounds, the closest point within the bound is estimated and used for retargeting, avoiding stagnation. Additionally, instead of a fixed height, the thumb is allowed to move perpendicular to the 2D surface along the palm, mapping the height of the human thumb tip to the robot thumb tip based on maximum and minimum height bounds. This approach ensures smoother thumb motion and enables the performance of more complex tasks compared to Holo-Dex [13].

6 Baseline Comparisons

Table 4 provides a comparison between OPEN TEACH and prior teleoperation systems considering features such as being calibration-free, compatible with multi-fingered hands, bimanual arms, and mobile manipulators, and being open-sourced.

Table 4: Comparison of OPEN TEACH's capabilities with prior teleoperation systems on features such as being calibration-free, compatible with multi-fingered hands, bimanual arms, and mobile manipulators, and being open-sourced.

	Calibration Free	Hands	Arms	Bimanual	Mobile Manipulation	Open-source
Joystick	✓	✗	✓	✗	✗	✓
Spacemouse	✓	✗	✓	✗	✗	✓
Phone Teloperation [31]	✓	✗	✓	✗	✗	✗
DexPilot [34]	✗	✓	✓	✗	✗	✗
Holo-Dex [13]	✓	✓	✗	✗	✗	✓
DIME [27]	✗	✓	✗	✗	✗	✓
TeachNet [46]	✓	✓	✗	✗	✗	✓
Telekinesis [47]	✓	✓	✓	✗	✗	✗
Qin et al. [48]	✓	✓	✗	✓	✗	✓
MVP-Real [28]	✗	✓	✓	✗	✗	✗
Transteleop [49]	✗	✓	✓	✗	✗	✗
Mosbach et al. [50]	✗	✓	✓	✗	✗	✓
AnyTeleop [35]	✓	✓	✓	✗	✗	✗
ALOHA [2]	✓	✗	✓	✓	✗	✓
Mobile ALOHA [51]	✓	✗	✓	✓	✓	✓
GELLO [32]	✓	✗	✓	✓	✗	✓
AirExo [33]	✓	✗	✓	✓	✗	✓
Dobb-E [4]	✓	✗	✓	✗	✓	✓
OPEN TEACH	✓	✓	✓	✓	✓	✓

7 Experimental Setup

We evaluate the versatility of OPEN TEACH by using it to collect demonstrations on six different setups — four in the real world and two in simulation. Each setup is a combination of a variant of a robot arm with either an Allegro Hand or a 2-fingered gripper. The real-world setups include:

1. **Franka-Allegro:** A Franka Arm with an Allegro Hand having the Xela tactile sensors.
2. **Kinova-Allegro:** A Kinova Jaco Arm with an Allegro Hand with the Xela tactile sensors.
3. **Bimanual:** 2 xArm7 robot arms with 2-fingered grippers.
4. **Stretch:** Hello Stretch mobile manipulator with a 2-fingered gripper.

The Franka-Allegro and Kinova-Allegro comprise a single Intel Realsense camera for data collection, whereas the Bimanual setup collects data from 5 different cameras. The Stretch has an iPhone attached to the wrist for data collection [4]. The simulated environments include:

1. **Allegro Sim:** A floating Allegro Hand capable of performing static and dynamic tasks.
2. **LIBERO Sim [23]:** A Franka Arm with a 2-fingered gripper placed in varied scenes.

8 Task Details

8.1 Demo Collections times

Table 5 provides the average times required to collect a demonstration for 16 tasks across 3 real-world setups (Franka-Allegro, Kinova-Allegro, Bimanual) and 2 simulated environments(Allegro sim, LIBERO sim).

8.2 Task Descriptions

Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9 provide rollouts of all the tasks performed both in the real world and in simulated environments. Each task rollout is labeled with the name of the task and a task description.

8.3 Task Details

8.3.1 Demo Collections times

Table 5 provides the average times required to collect a demonstration for 16 tasks across 3 real-world setups (Franka-Allegro, Kinova-Allegro, Bimanual) and 2 simulated environments(Allegro sim, LIBERO sim).

Table 5: Time

Robot Setup	Task	Average time to collect a demo (in s)
Franka-Allegro	Open box	45
	Grasp sponge	60
	Pick up tea satchet	60
	Grasp object and twist	35
Kinova-Allegro	Unfold towel	40
	Open a pack of cream	10
	Open ketchup bottle	40
Bimanual	Uncap marker	60
	Sweep table	60
	Pour sprinkles in a bowl	40
Allegro Sim	Flip cube	3
	Flip sponge	20
	Pinch Grasp	15
LIBERO Sim	Close top drawer of cabinet	10
	Turn on stove	25
	Pick up and put soup can in the basket	30

8.3.2 Task Descriptions

Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9 provide rollouts of all the tasks performed both in the real world and in simulated environments. Each task rollout is labeled with the name of the task and a task description.

8.4 User Study

Following up from Section **??**, we provide the success rate and average completion times for all 15 users for each task performed in Table 6 and Table 7 respectively. Each user roughly performed 3 tasks on average, with 5 trials for each task. As mentioned in Section **??**, since the Holo-Dex [13] and AnyTeleop [35] baselines lack open-source code for arm retargeting, we were unable to evaluate them on tasks involving arm movements. We observe a wide range of differences in success rates and average completion times demonstrating the inherent variations across users.

Table 6: Success rates for the user study conducted across 15 individuals. Each user roughly performs 3 tasks on average.

User	Method	Success Rate (in 5 trials)				
		Flip Cube	Pinch Grasp	Pour	Pick and Place	Open Box of Mints
User 1	Holo-Dex	1	0	-	-	-
	AnyTeleop	0.8	0.2	-	-	-
	Open Teach	1	0.8	0.2	-	-
User 2	Holo-Dex	-	0.2	-	-	-
	AnyTeleop	-	0.2	-	-	-
	Open Teach	-	0.8	-	0.8	0.8
User 3	Holo-Dex	1	0	-	-	-
	AnyTeleop	1	0.2	-	-	-
	Open Teach	1	0.8	-	-	0.2
User 4	Holo-Dex	1	0	-	-	-
	AnyTeleop	1	0.2	-	-	-
	Open Teach	1	0.8	-	0.6	0.4
User 5	Holo-Dex	-	0	-	-	-
	AnyTeleop	-	0.6	-	-	-
	Open Teach	-	0.2	0.4	1	-
User 6	Holo-Dex	-	0	-	-	-
	AnyTeleop	-	0.6	-	-	-
	Open Teach	-	0.8	-	0.2	-
User 7	Holo-Dex	-	0	-	-	-
	AnyTeleop	-	0	-	-	-
	Open Teach	-	0.6	0.8	0.8	0.4
User 8	Holo-Dex	1	-	-	-	-
	AnyTeleop	1	-	-	-	-
	Open Teach	1	-	-	-	-
User 9	Holo-Dex	-	0	-	-	-
	AnyTeleop	-	0.4	-	-	-
	Open Teach	-	0.8	0	-	0.6
User 10	Holo-Dex	-	0	-	-	-
	AnyTeleop	-	0.2	-	-	-
	Open Teach	-	0.6	0.4	1	1
User 11	Holo-Dex	1	-	-	-	-
	AnyTeleop	1	-	-	-	-
	Open Teach	1	-	-	0.8	0.4
User 12	Holo-Dex	1	-	-	-	-
	AnyTeleop	1	-	-	-	-
	Open Teach	1	-	-	-	-
User 13	Holo-Dex	1	-	-	-	-
	AnyTeleop	1	-	-	-	-
	Open Teach	1	-	0.6	-	-
User 14	Holo-Dex	-	0	-	-	-
	AnyTeleop	-	0.4	-	-	-
	Open Teach	-	0.6	-	-	0.8
User 15	Holo-Dex	1	-	-	-	-
	AnyTeleop	1	-	-	-	-
	Open Teach	1	-	0.4	-	-

Table 7: Average completion times for successful trials for the user study conducted across 15 individuals. Each user roughly performs 3 tasks on average. *NS* denotes cases where no successes were achieved.

User	Method	Average completion time for successful demonstrations (in s)				
		Flip Cube	**Pinch Grasp**	**Pour**	**Pick and Place**	**Open Box of Mints**
User 1	Holo-Dex	4.6	NS	-	-	-
	AnyTeleop	20.2	22.5	-	-	-
	Open Teach	5.4	18.6	66	-	-
User 2	Holo-Dex	-	17.5	-	-	-
	AnyTeleop	-	18.9	-	-	-
	Open Teach	-	20.6	-	29.7	12.2
User 3	Holo-Dex	5.4	NS	-	-	-
	AnyTeleop	18.3	7.8	-	-	-
	Open Teach	5.1	12.6	-	-	11.3
User 4	Holo-Dex	11	NS	-	-	-
	AnyTeleop	13.2	31.4	-	-	-
	Open Teach	6.2	7.5	-	16.9	48.4
User 5	Holo-Dex	-	NS	-	-	-
	AnyTeleop	-	11.4	-	-	-
	Open Teach	-	10.9	41.6	12.4	-
User 6	Holo-Dex	-	NS	-	-	-
	AnyTeleop	-	12.7	-	-	-
	Open Teach	-	10.5	-	23.57	-
User 7	Holo-Dex	-	NS	-	-	-
	AnyTeleop	-	NS	-	-	-
	Open Teach	-	19.1	21.49	49	37.8
User 8	Holo-Dex	6.5	-	-	-	-
	AnyTeleop	5.4	-	-	-	-
	Open Teach	4.7	-	-	-	-
User 9	Holo-Dex	-	NS	-	-	-
	AnyTeleop	-	49.9	-	-	-
	Open Teach	-	65.3	NS	-	32.21
User 10	Holo-Dex	-	NS	-	-	-
	AnyTeleop	-	48	-	-	-
	Open Teach	-	30.8	40.3	48.7	21.3
User 11	Holo-Dex	6.7	-	-	-	-
	AnyTeleop	11.5	-	-	-	-
	Open Teach	5.6	-	-	21.8	15.7
User 12	Holo-Dex	6.2	-	-	-	-
	AnyTeleop	11	-	-	-	-
	Open Teach	3.8	-	-	-	-
User 13	Holo-Dex	8.9	-	-	-	-
	AnyTeleop	14.2	-	-	-	-
	Open Teach	5.8	-	18.1	-	-
User 14	Holo-Dex	-	NS	-	-	-
	AnyTeleop	-	49.9	-	-	-
	Open Teach	-	65.3	-	-	132.5
User 15	Holo-Dex	13.2	-	-	-	-
	AnyTeleop	14.6	-	-	-	-
	Open Teach	6.3	-	53.1	-	-

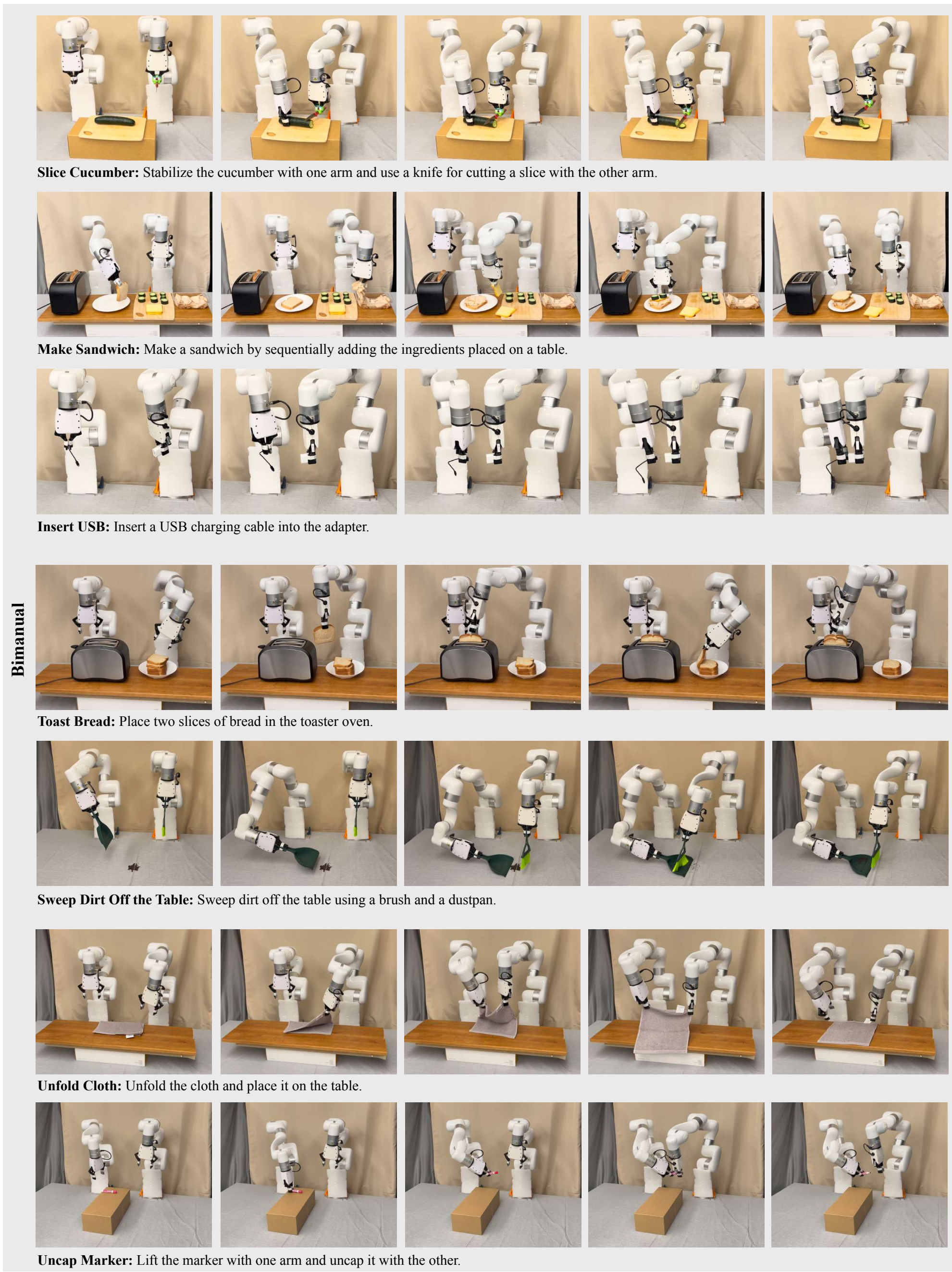


Figure 4: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

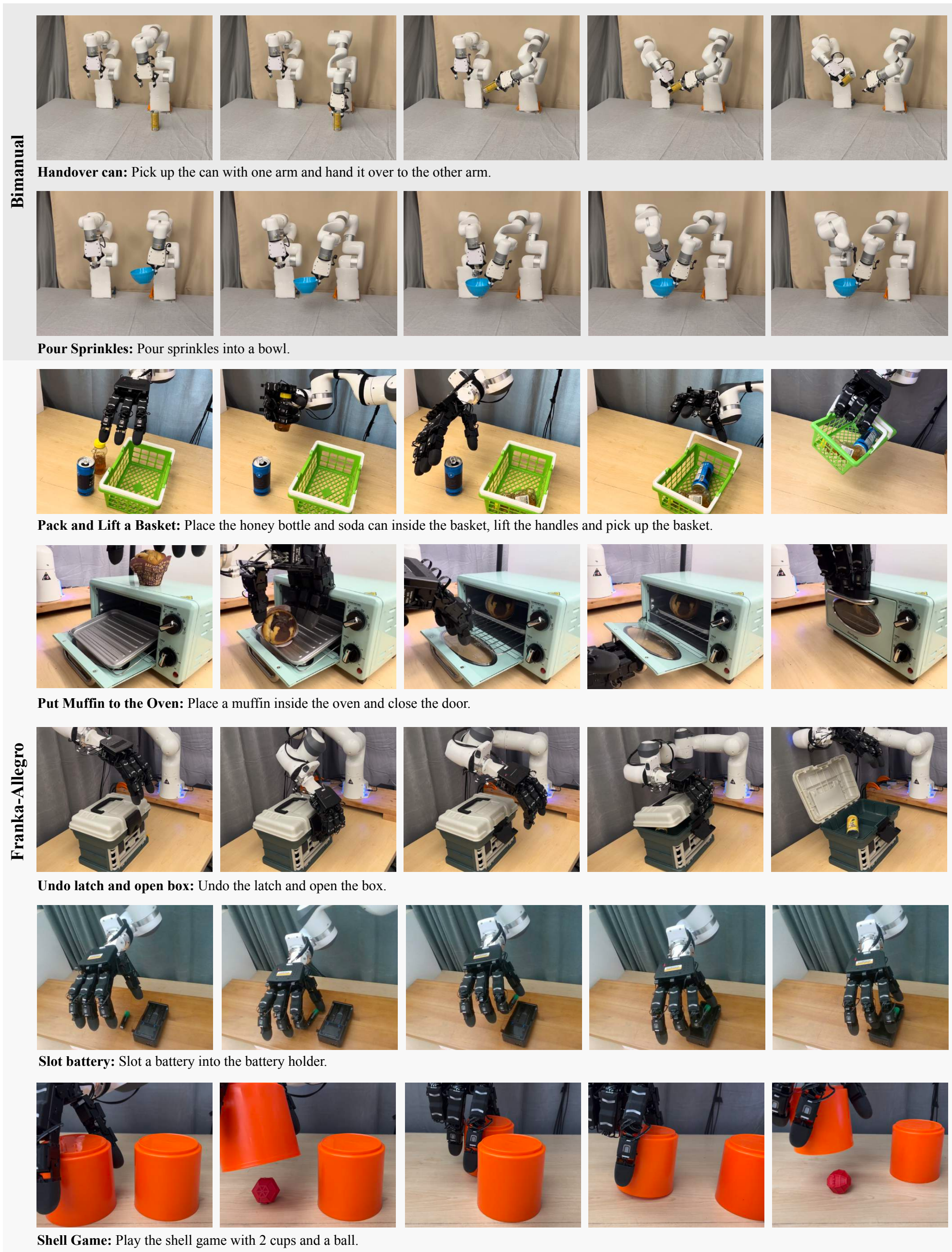


Figure 5: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

Figure 6: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

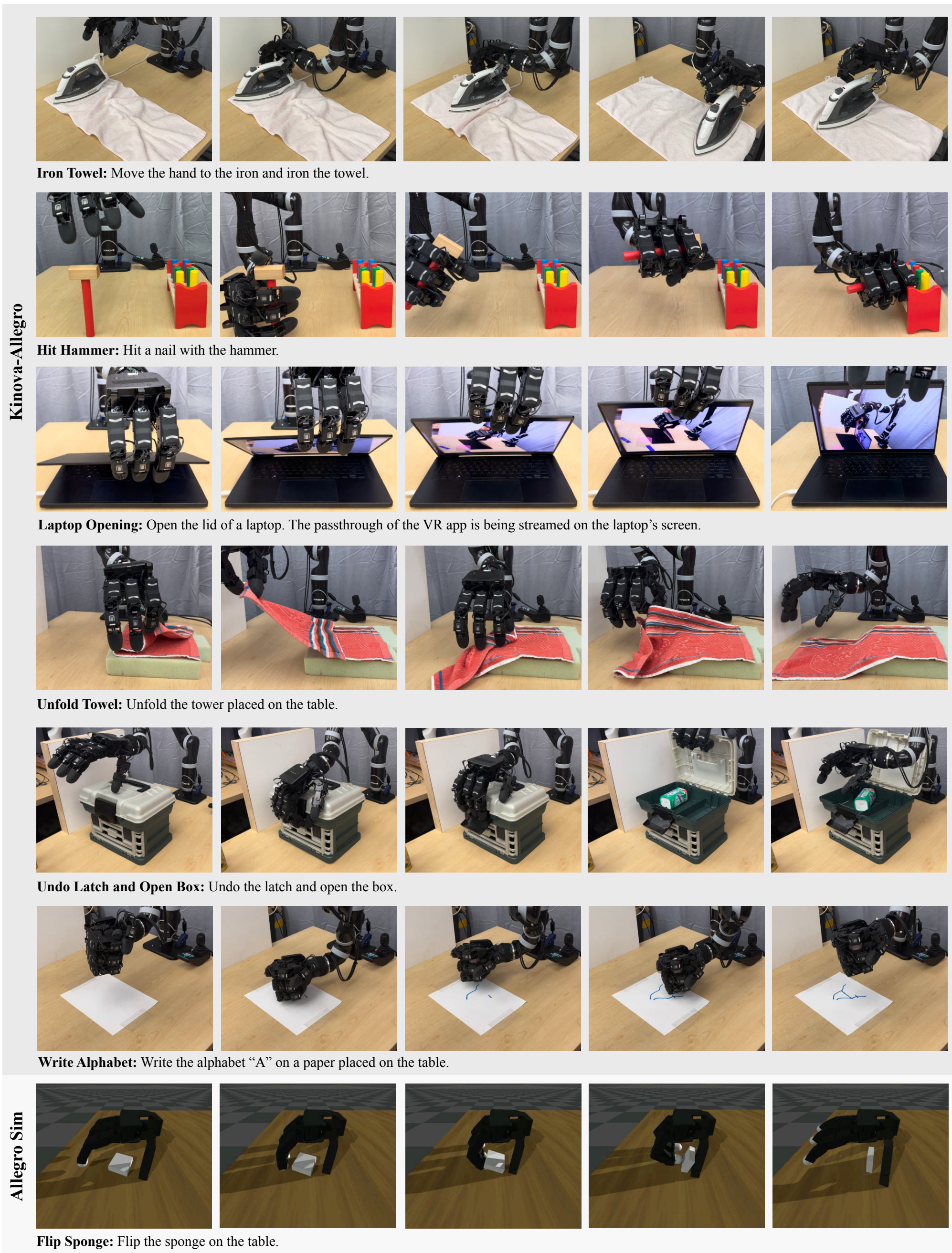


Figure 7: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

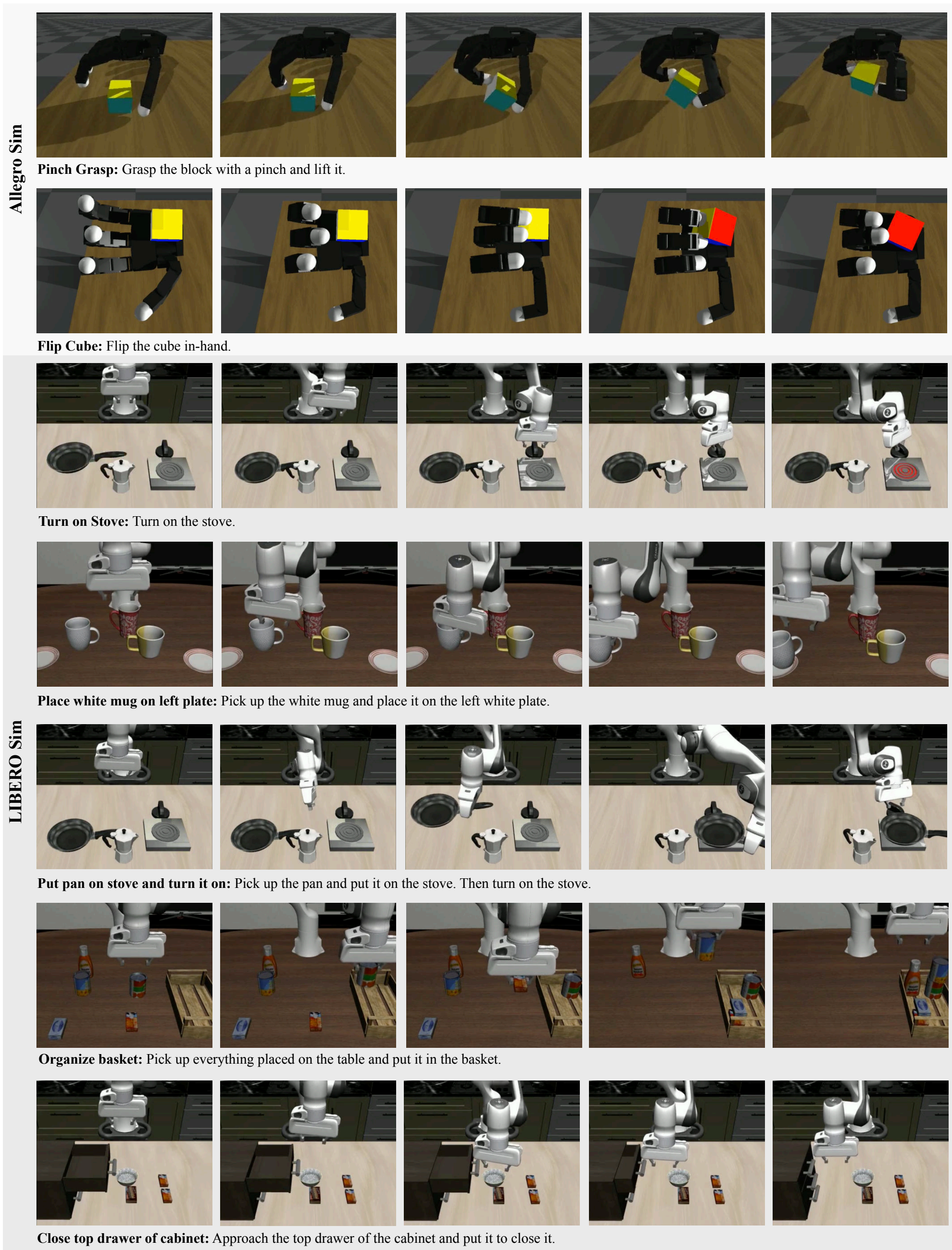


Figure 8: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

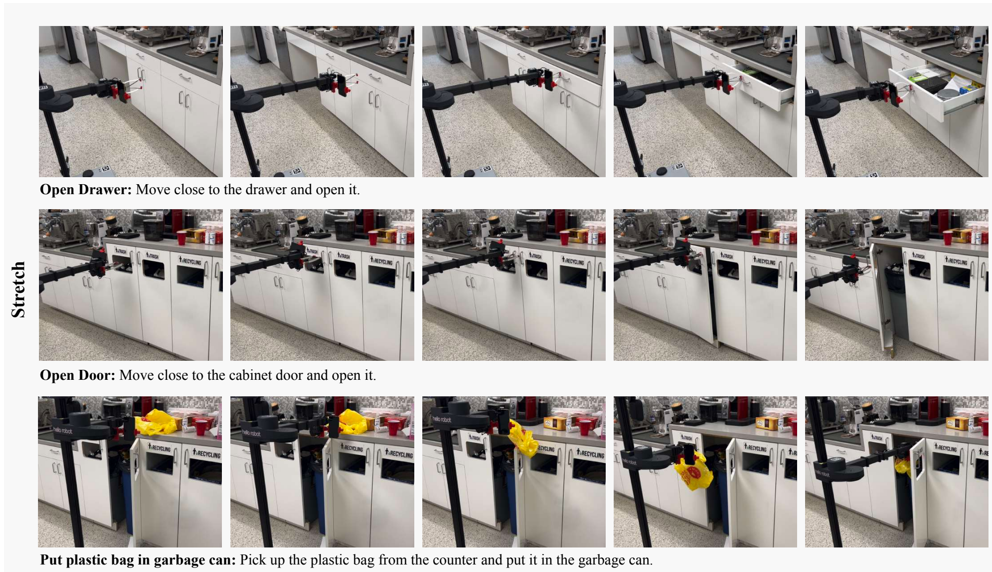


Figure 9: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

Table 8: Performance of policies learned on data collected through OPEN TEACH. FISH and BC were used to train policies for Allegro Sim and Libero Sim respectively. We report the mean and standard deviation for 25 evaluation trials across 3 seeds for each task.

Robot Setup	**Task**	**Number of Demos**	**Success Rate (25 trials)**
Allegro Sim	Flip Cube	6	0.97 ± 0.03
	Flip Sponge	6	0.79 ± 0.05
	Pinch Grasp	6	0.75 ± 0.07
Libero Sim	Close Top Drawer of Cabinet	10	0.96 ± 0.03
	Turn on Stove	10	0.95 ± 0.04
	Pick and Place Soup into Basket	50	0.77 ± 0.02

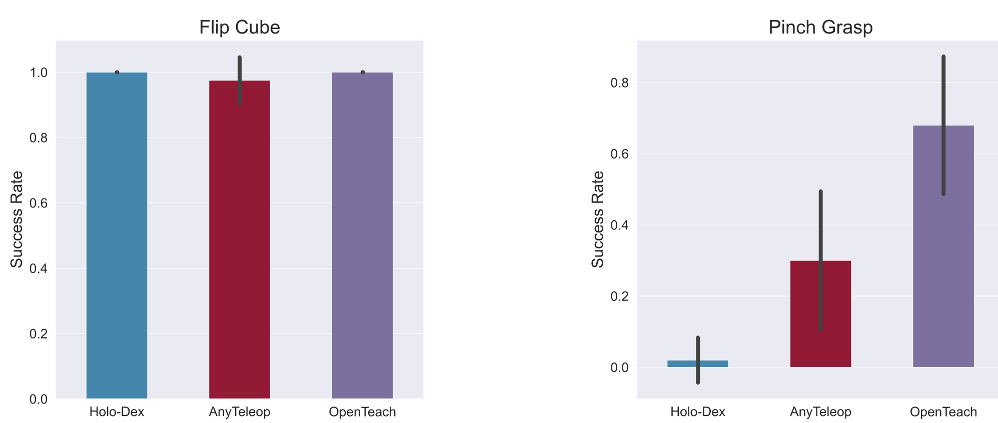


Figure 10: Success rates for the user study conducted across 15 individuals on 2 tasks - Flip Cube and Pinch Grasp. We report the mean and standard deviation for 3 methods - Holo-Dex, AnyTeleop, and Open Teach (Ours).

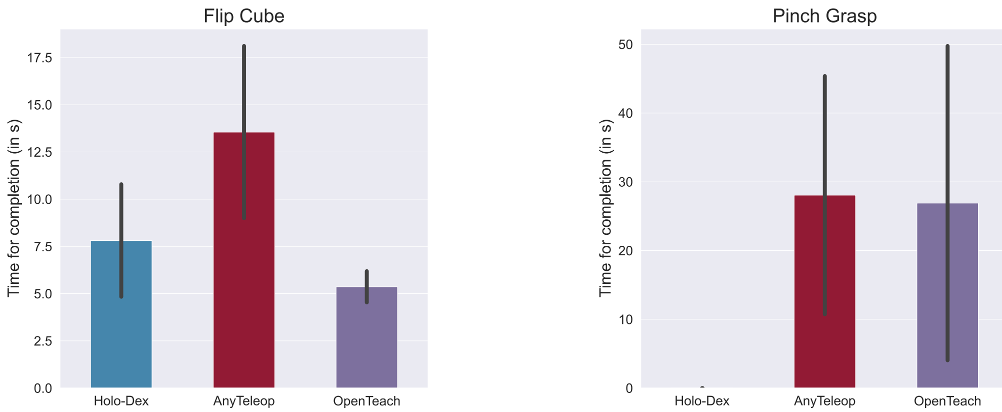
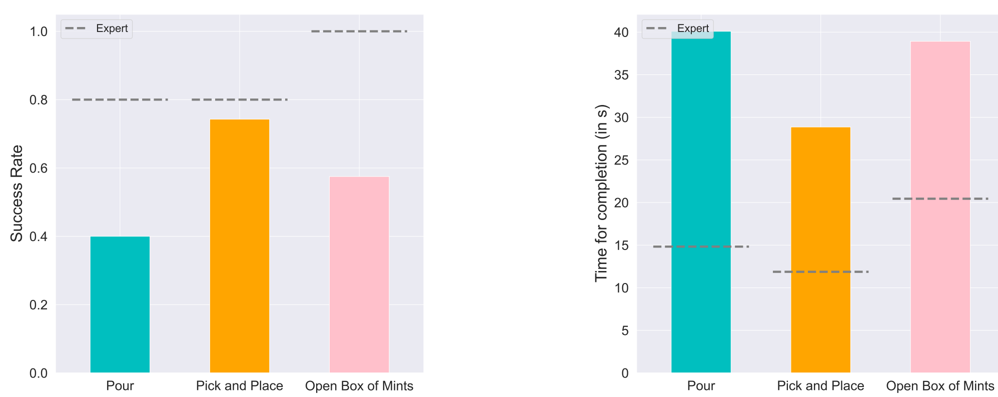


Figure 11: Average completion times for successful trials for the user study conducted across 15 individuals for 2 tasks - Flip Cube and Pinch Grasp. We report the mean and standard deviation for 3 methods - Holo-Dex, AnyTeleop, and Open Teach (Ours).



(a) Success Rate

(b) Time for completion (in seconds)

Figure 12: We compare the (a) success rate, and (b) average completion time (in seconds) for using OPEN TEACH between an expert and 15 individuals participating in a user study. We report this comparison for 3 tasks - Pour, Pick and Place, and Open Box of Mints.