

455 Appendix

456 A Background on Imitation Learning

457 A.1 Behavior Cloning

458 Given a dataset of expert rollouts for a desired task in the form of observation and action pairs
459 $\mathcal{D} \equiv \{(o, a)\} \subset \mathcal{O} \times \mathcal{A}$, behavior cloning (BC) aims to learn a policy $\pi: \mathcal{O} \rightarrow \mathcal{A}$ that models
460 this data without any online interactions with the environment nor a reward function. Often, such
461 policies are chosen from a hypothesis class parameterized by a parameter set θ . Following this
462 convention, the objective of BC is to find the value θ that maximizes the probability of the observed
463 data.

$$\theta^* = \operatorname{argmax}_{\theta} \prod_t \mathbb{P}(a_t | o_t; \theta) \quad (1)$$

464 When constrained to unimodal isotropic Gaussians, this maximum likelihood estimation problem
465 leads to minimizing the Mean Squared Error (MSE), $\sum_t \|a_t - \pi(o_t; \theta)\|^2$.

466 A.2 Inverse Reinforcement Learning

467 In this work, we employ FISH [21] and TAVI [66] to learn visual and visuotactile policies respec-
468 tively. For both of these methods, the first phase involves obtaining a non-parametric base-policy
469 $\pi^b: \mathcal{Z} \rightarrow \mathcal{A}$ with encoded representations $z \in \mathcal{Z}$ and actions $a \in \mathcal{A}$. Then a residual policy
470 $\pi^r: \mathcal{Z} \times \mathcal{A} \rightarrow \mathcal{A}$ is learned atop the base policy π^b such that an action sampled from the final
471 policy π is the sum of the base action $a^b \sim \pi^b(z)$ and the residual offset $a^r \sim \pi^r(z, a^b)$. The re-
472 ward for learning the residual policy through inverse RL is obtained through optimal transport based
473 trajectory matching [20, 68].

474 B Framework details

475 B.1 Structure of the framework

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

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

- 480 1. **Detector:** Receives the hand keypoints from the Meta Quest 3 and publishes them to ZMQ
481 sockets.
- 482 2. **Keypoint Transformer:** Subscribes the keypoints published by the detector and maps
483 them to the robot pose.
- 484 3. **Operator:** Receives the robot pose from the keypoint transformer and the current robot
485 state from the controller. The operator computes the robot’s actions which are published to
486 a ZMQ socket.
- 487 4. **Controller:** Subscribes an action from the operator and takes an action in the real or simu-
488 lated environment. After taking the action, the controller publishes the current states of the
489 environment for use by the operator.
- 490 5. **Visualizer:** Subscribes the RGB images from the camera process (or the environment in
491 case of simulations) and puts it on the screen inside the VR app for visualization during
492 teleoperation.

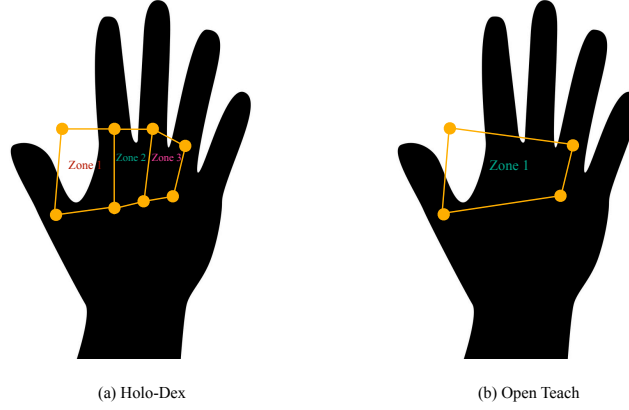


Figure 4: Thumb retargeting difference

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.

B.2 Thumb Retargeting for Robot Hand

Section 3.3 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. 4(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. 4(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].

C Baseline Comparisons

Table 5 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.

Code Snippet 1: Robot control using VR

```
# Initialize robot and VR
vr = VR()
robot = Robot()
while(True):
    # Step 1: Get hand pose from VR
    hPose = vr.getHandPose()
    # Step 2: Retarget to robot pose
    rPose = robot.retargetH2R(hPose)
    # Step 3: Move robot to target pose
    robot.move(rPose)
```

Table 4: 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

Table 5: 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 [47]	✓	✓	✗	✗	✗	✓
Telekinesis [39]	✓	✓	✓	✗	✗	✗
Qin et al. [69]	✓	✓	✗	✓	✗	✓
MVP-Real [28]	✗	✓	✓	✗	✗	✗
Transteleop [54]	✗	✓	✓	✗	✗	✗
Mosbach et al. [57]	✗	✓	✓	✗	✗	✓
AnyTeleop [35]	✓	✓	✓	✗	✗	✗
ALOHA [2]	✓	✗	✓	✓	✗	✓
Mobile ALOHA [53]	✓	✗	✓	✓	✓	✓
GELLO [32]	✓	✗	✓	✓	✗	✓
AirExo [33]	✓	✗	✓	✓	✗	✓
Dobb-E [4]	✓	✗	✓	✗	✓	✓
OPEN TEACH	✓	✓	✓	✓	✓	✓

522 **D Task Details**

523 **D.1 Demo Collections times**

524 Table 4 provides the average times required to collect a demonstration for 16 tasks across 3 real-
525 world setups (Franka-Allegro, Kinova-Allegro, Bimanual) and 2 simulated environments (Allegro
526 sim, LIBERO sim).

527 **D.2 Task Descriptions**

528 Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, and Fig. 10 provide rollouts of all the tasks performed both in
529 the real world and in simulated environments. Each task rollout is labeled with the name of the task
530 and a task description.

531 **E User Study**

532 Following up from Section 4.6, we provide the success rate and average completion times for all 15
533 users for each task performed in Table 6 and Table 7 respectively. Each user roughly performed 3
534 tasks on average, with 5 trials for each task. As mentioned in Section 4.6, since the Holo-Dex [13]
535 and AnyTeleop [35] baselines lack open-source code for arm retargeting, we were unable to evaluate
536 them on tasks involving arm movements. We observe a wide range of differences in success rates
537 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	-	-

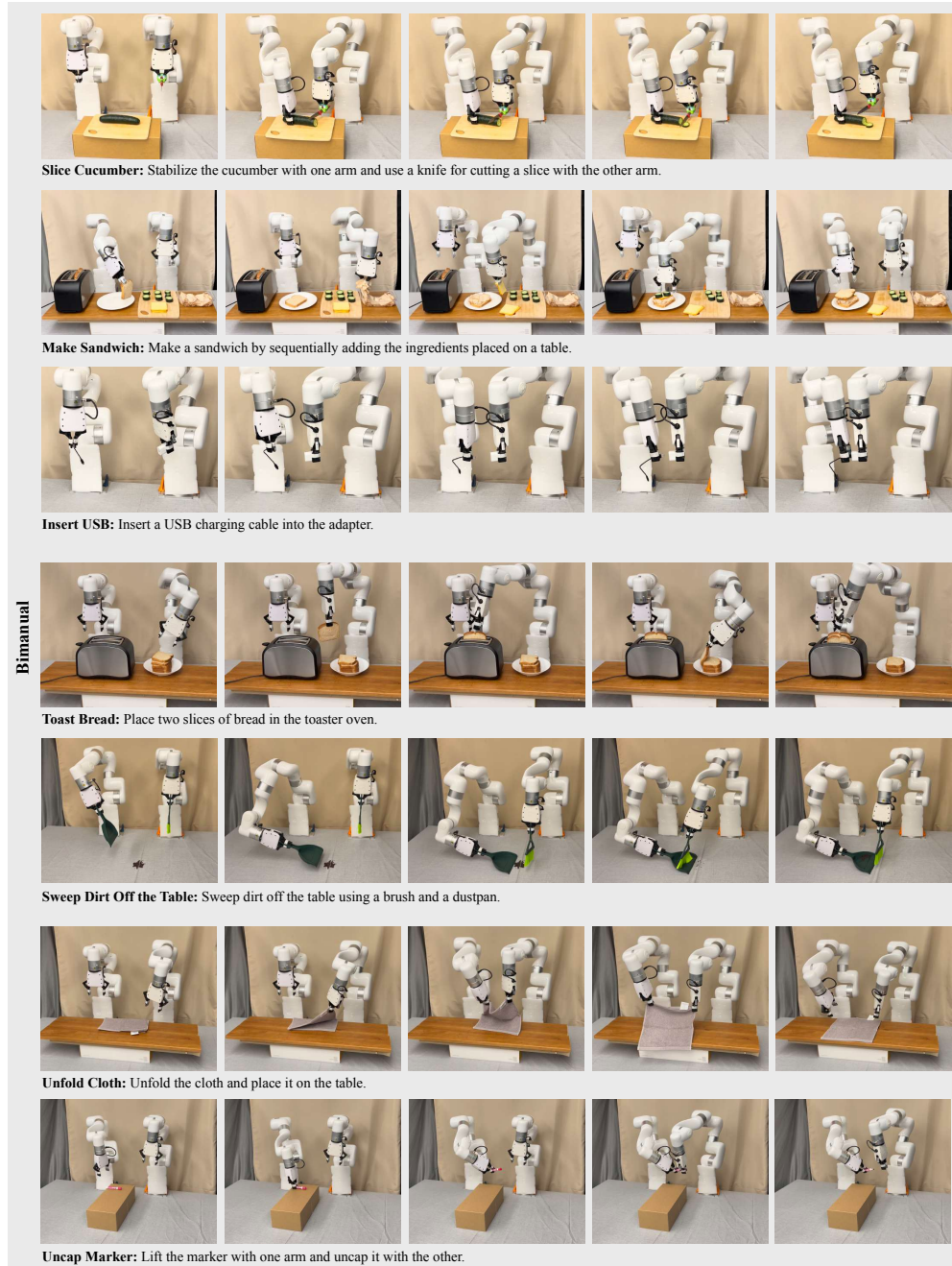


Figure 5: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

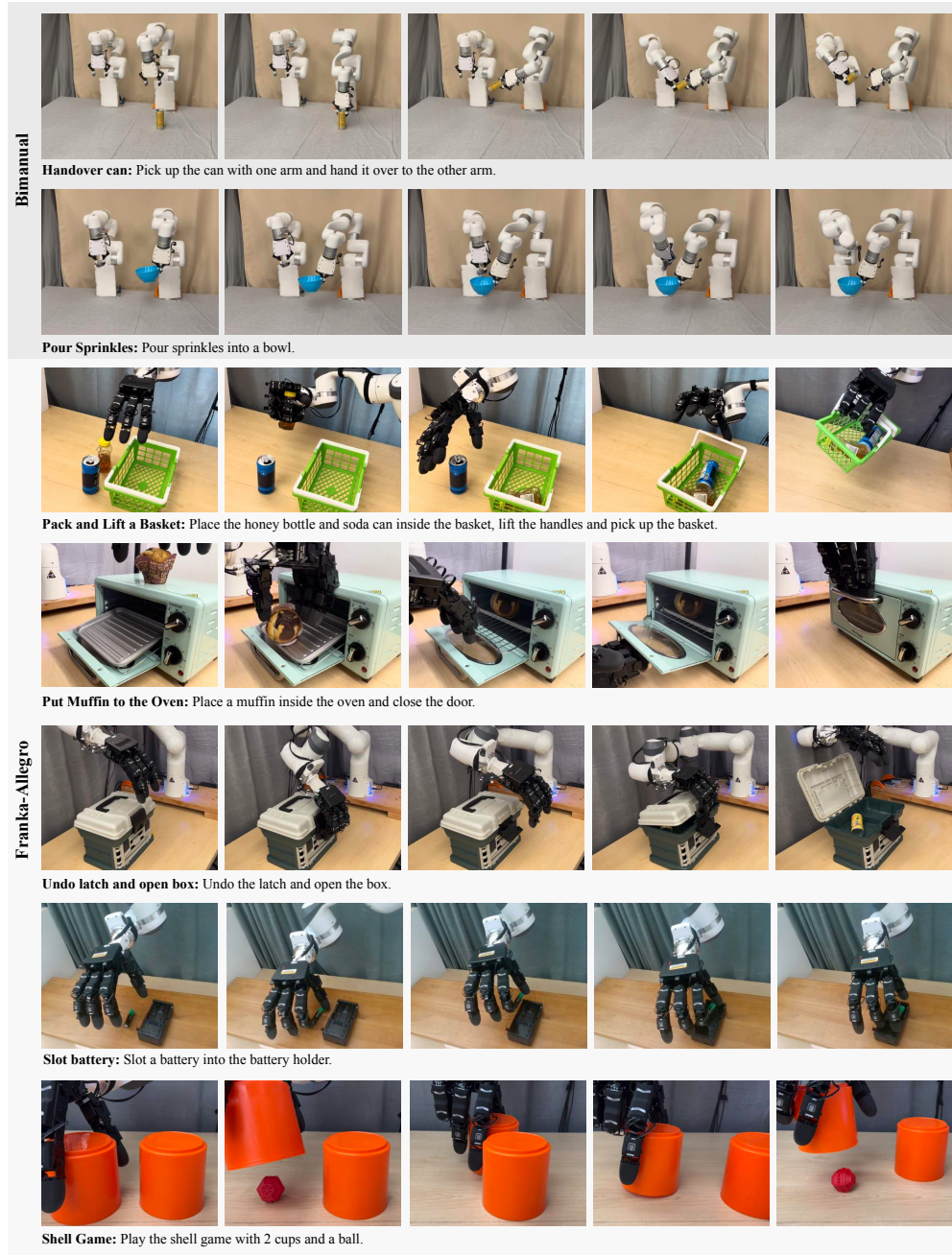


Figure 6: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.



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

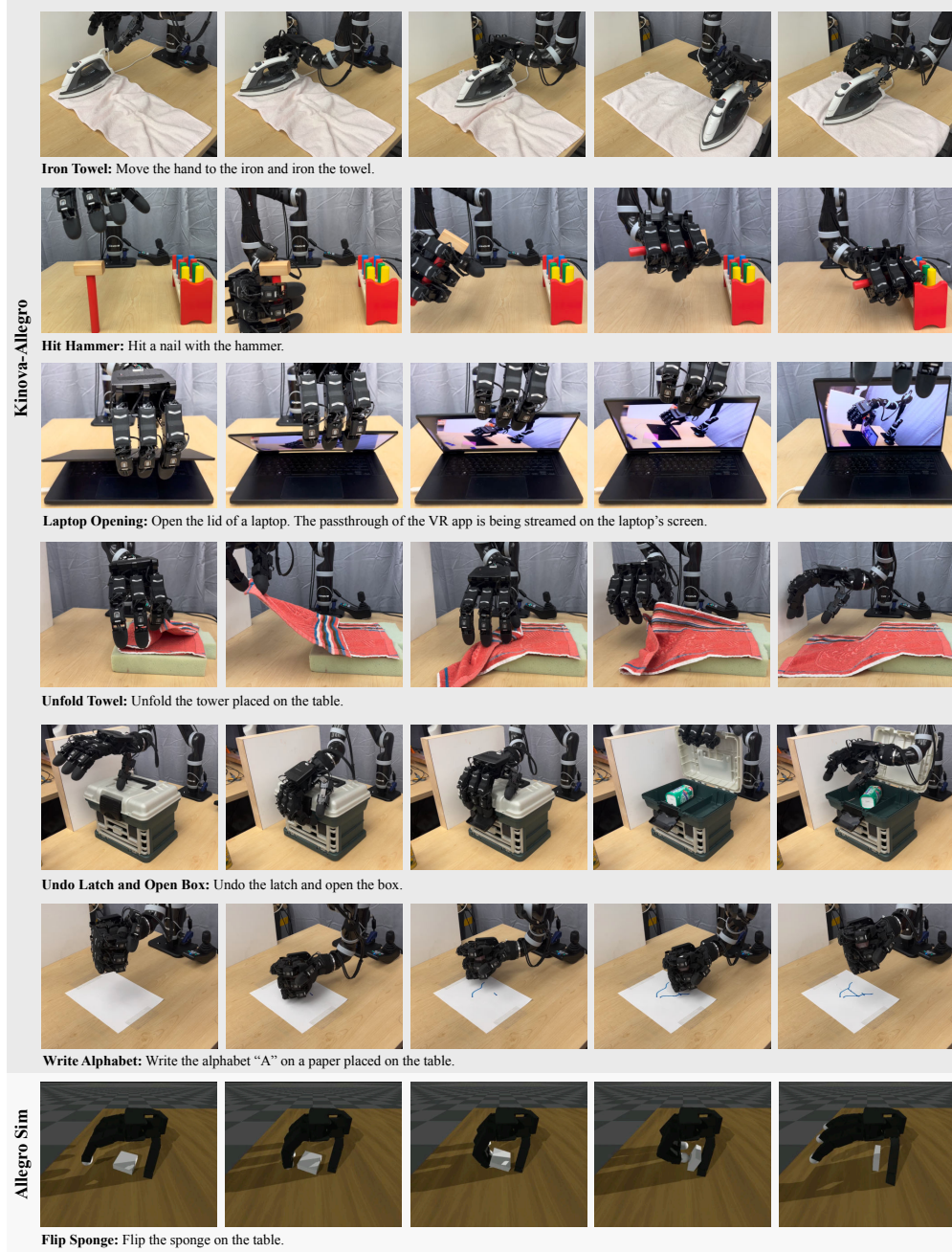


Figure 8: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

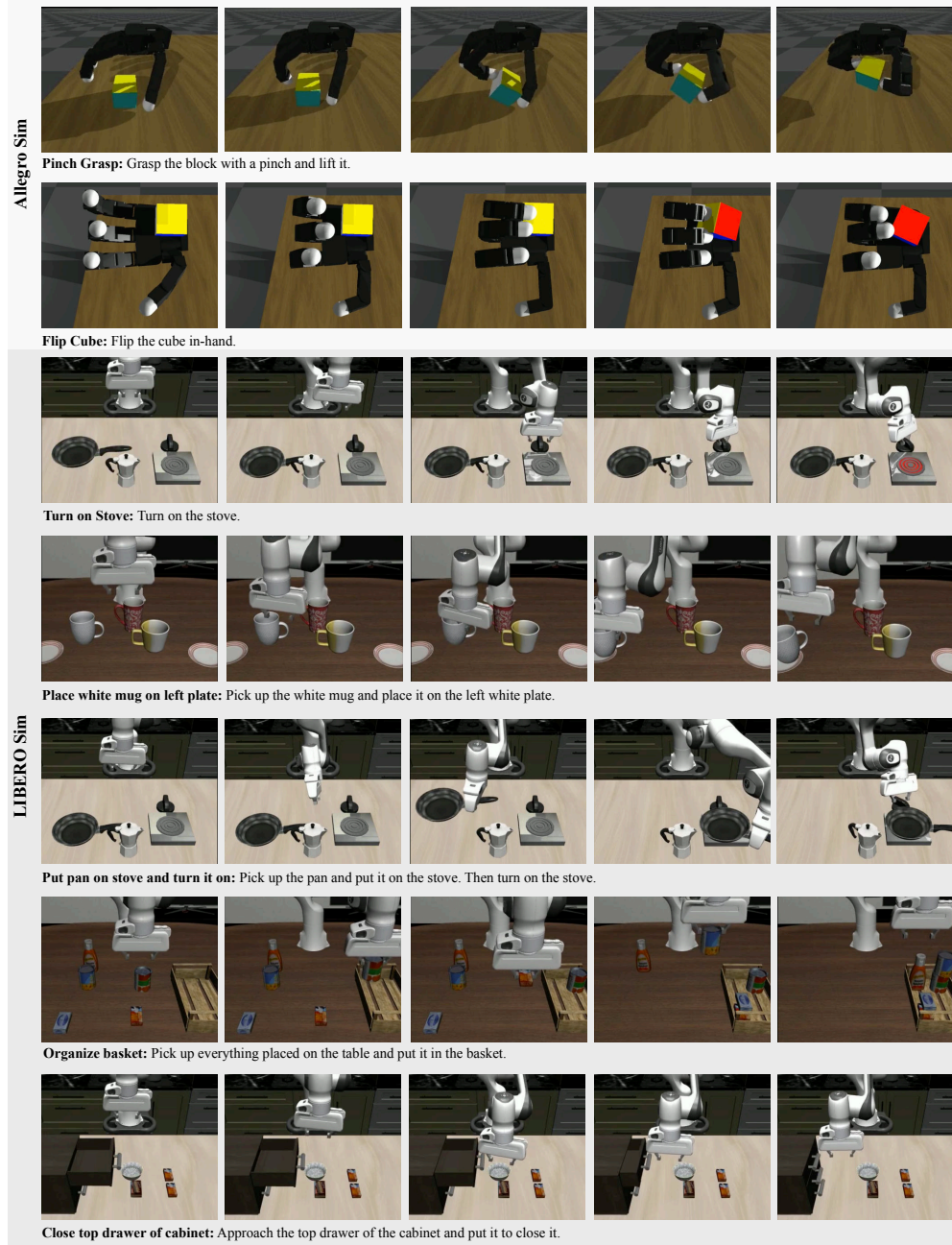


Figure 9: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.

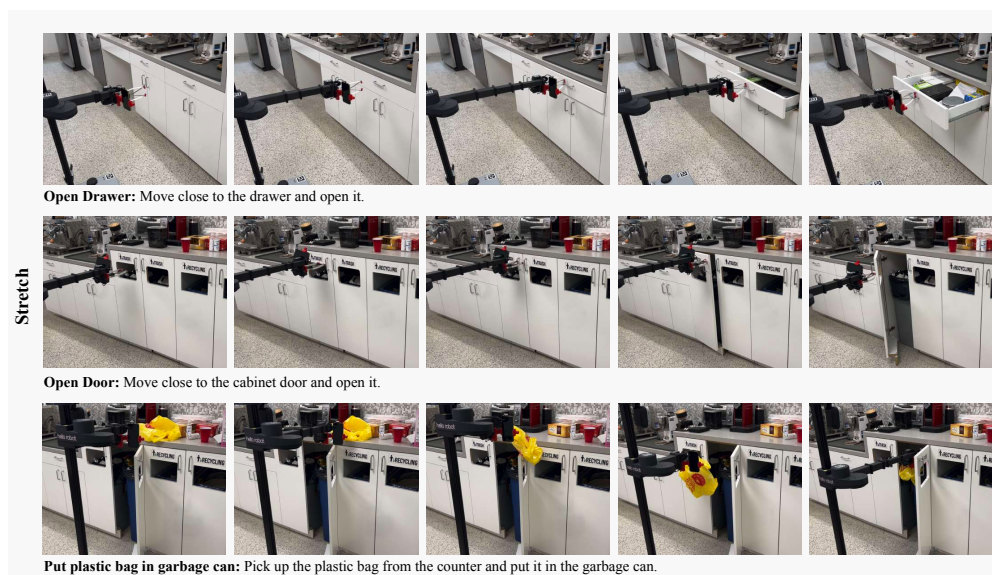


Figure 10: Real world task rollouts demonstrating the ability of OPEN TEACH to perform intricate, long-horizon tasks.