

DEXPORT: A Hand-Held Skill Interface for Dexterous Hand Data Collection

Amber Xie^{1,2}, Jiawei Fu^{1,3}, Xiaofeng Guo¹, Carmelo Sferrazza^{†1}, Guanya Shi^{†1}, and Haozhi Qi^{†1}

¹Amazon FAR (Frontier AI & Robotics), ²Stanford University, ³UC San Diego

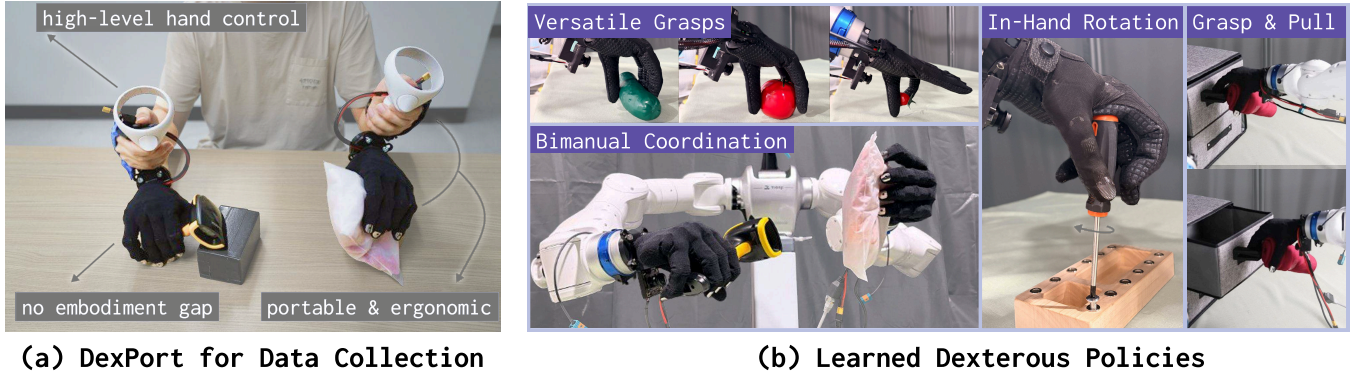


Fig. 1. **Portable dexterous data collection with DEXPORT.** (a) The operator moves the hand-held device to control wrist motion and uses a small set of skills to control the fingers. The portable interface uses the same robot hand used at deployment and records commanded joint targets and measured joint states. (b) Visuomotor policies trained on these demonstrations predict wrist and hand commands directly, supporting varied grasps, bimanual coordination, in-hand manipulation, and opening a drawer while holding an object.

Abstract—Collecting high-quality demonstrations remains a central bottleneck for learning dexterous manipulation. Teleoperation records executable robot actions, but it requires a complete robot setup and places the burden of coordinating many finger joints on the operator. Human demonstrations are easier to collect, but their motions must be estimated and retargeted to the robot, and they do not directly provide robot actions. In this paper, we introduce DEXPORT, a portable data-collection interface for high-DoF robot hands. It uses the same hand hardware for collection and deployment, avoiding a hand embodiment gap, and records both commanded joint targets and measured joint states. Our key observation is that a small set of skills can support a wide range of dexterous tasks. During collection, the operator moves the hand-held interface to specify the robot’s wrist motion and selects or adjusts skills that coordinate the finger motion. This removes the need to control each joint independently while retaining behaviors such as precise grasping and in-hand manipulation. We collect over 2,500 demonstrations across several real-world tasks in laboratory and in-the-wild settings without a robot arm. We then use behavior cloning to train visuomotor policies directly on the recorded wrist motions and hand commands, without skill labels. We evaluate on four tasks: pick-and-place with diverse objects, screwing with in-hand tool use, drawer stowing that requires simultaneous grasping and pulling, and bimanual barcode scanning. Using recorded robot commands instead of measured joint states improves average final-task success from 11% to 64% across the three tasks evaluated by success rate. Co-training with in-the-wild data improves average pick-and-place success from 36% to 78% across the three evaluation conditions. Videos are available at <https://amberxie88.github.io/dexport/>.

I. INTRODUCTION

Multi-fingered hands allow robots to use different grasps, manipulate tools, and reorient objects in hand. Learning these

behaviors from demonstrations requires data that capture detailed finger motion and contact-rich interactions. Teleoperation records executable robot actions, but it requires a complete robot setup. Collecting in-hand manipulation demonstrations through teleoperation is particularly challenging because the operator must coordinate many finger joints [1], [2], [3], [4], [5]. On the other hand, human hand demonstrations are easier to collect, but their motions must be estimated and retargeted to the robot, and they do not directly provide robot actions [6], [7], [8]. We therefore need a method that scales beyond a fixed robot setup while preserving executable hand actions.

To address this issue for parallel-jaw grippers, the Universal Manipulation Interface (UMI) [9] uses a portable, hand-held interface to collect demonstrations naturally and at scale. However, extending this paradigm to multi-fingered hands has not been trivial. Early efforts, including Dex-UMI [10] and DEXOP [11], can collect dexterous and contact-rich behaviors using wearable hand exoskeletons. However, designing a high-DoF exoskeleton involves a difficult trade-off between dexterity and wearability: supporting more independent finger motions increases both mechanical complexity and the operator’s control burden. In addition, these systems record exoskeleton configurations rather than the commands and resulting states of the deployed robot hand. RealDexUMI [12] preserves this information by collecting executable commands with the physical robot hand, but demonstrates the approach on a hand with only six active DoFs. It remains unclear how to preserve these benefits for a hand with many independently controlled joints without making the interface cumbersome.

Inspired by these works, we address three requirements

[†]Amazon FAR team co-lead.

System	DoF	Weight	Same Emb.	Action Data
UMI [9]	1	780 g	✓	✗
DexViTac [13]	6	<600 g	✗	✗
DEXOP [11]	12/9/7	>1000 g	✓	✗
DexEXO [14]	6	>1000 g	✓	✗
DEX-Mouse [15]	6	>1000 g	✓	✓
Exo-ViHa [16]	6	1100 g	✗	✓
DexUMI [10]	12/6	850 g	✓	✗
RealDexUMI [12]	6	680 g	✓	✓
DEXPORT	20	955 g	✓	✓

TABLE I: **Comparison of dexterous demonstration interfaces.** Systems are compared by controllable hand degrees of freedom (DoF), device weight, use of the deployed robot’s hand embodiment (Same Emb.), and recording of commanded joint targets (Action Data). DEXPORT controls a 20-DoF hand and records the commands sent to its joints.

for a portable interface for dexterous hands. Table I compares existing systems against these requirements. First, the interface should use the same end effector as the deployed robot (**Same Emb.**), so that hand motions need no retargeting and no embodiment gap is introduced. Second, it should record the commands sent to the hand (**Action Data**) rather than relying only on actions inferred from future measured states. This distinction matters during contact, when the commanded target and measured joint state may differ. Third, it should expose enough controllable degrees of freedom (**DoF**) to support tasks that require multi-fingered dexterity; otherwise, a simpler gripper would be sufficient. Reusing the deployment hand satisfies the first two requirements, but creating an intuitive and comfortable interface for high-DoF control is challenging, especially when the operator must control each joint directly.

In this paper, we introduce DEXPORT, a portable data-collection interface built around these requirements. Our key design principle is to collect demonstrations with the same hand hardware used at deployment while controlling it through a small set of skills. Because the collection hand shares the deployment hand’s hardware and control interface, each demonstration records native commanded joint targets and resulting states. The operator moves the device to specify the robot’s wrist motion and selects or adjusts skills to control the hand. Parameterized skills produce power, pinch, and two-finger grasps, while a learned closed-loop skill generates the finger gaiting needed for in-hand rotation. This skill-level interface leaves task decisions to the operator while the robot hand produces the coordinated finger motion.

We evaluate DEXPORT on four tasks that require different forms of dexterity: selecting grasps for objects of different shapes, opening a drawer while holding an object in the same hand, rotating a tool in hand, and coordinating two hands to scan a barcode. We collect demonstrations both in the lab and in unstructured environments, without a robot arm or external tracking infrastructure. We then use behavior cloning to train visuomotor policies directly on the recorded wrist motions and hand commands, without skill labels. Our policies complete all four tasks and outperform a state-action baseline trained without commanded hand actions on each task. Across the three tasks scored by success rate,

final-task success averages 64% versus 11%. Co-training with in-the-wild demonstrations further improves pick-and-place in every condition, including unseen objects and scenes with distractors. DEXPORT also collects demonstrations at roughly $1.4\times$ the rate of glove teleoperation.

To summarize, our contributions are as follows:

- A portable and ergonomic data-collection interface that uses the same hand hardware and control space as the deployed robot. It avoids a hand embodiment gap and records native commanded actions and measured states.
- A skill-level control scheme that makes a high-DoF hand practical to operate during portable data collection. A compact set of parameterized and learned skills replaces joint-by-joint control while preserving the ability to perform dexterous tasks.
- An evaluation across four real-world tasks, including data collection in unstructured environments and visuomotor policies trained directly on the recorded wrist motions and hand commands, without skill labels.

II. RELATED WORK

Hand-Held Grippers for Robot Learning. Hand-held interfaces offer a middle ground between robot teleoperation and unconstrained human video: they can be carried into diverse environments while retaining an end-effector-centered observation and action space. Early systems used instrumented reacher-grabbers or other portable tools and recovered their motion using structure from motion, visual odometry, or smartphone odometry [17], [18], [19]. UMI [9] makes this approach more accurate and broadly useful through a hand-held parallel-jaw gripper, a wide field-of-view camera, visual-inertial pose tracking, and a policy interface that accounts for deployment latency. Matching the collection and deployment grippers further reduces the embodiment gap. UMI demonstrates that portable collection can support precise and long-horizon behaviors, but its gripper has only one actuated degree of freedom. Extending these benefits to a multi-fingered hand requires a practical way to control its higher-dimensional action space.

Learning Dexterous Manipulation. Dexterous manipulation skills are commonly learned from simulation, human demonstrations, or robot teleoperation. Simulation can generate robot actions at scale and has enabled complex in-hand behaviors, but transferring policies that rely on contact dynamics and tactile observations to real hardware remains difficult [20], [21], [22]. Human demonstrations collected from videos, hand tracking, or motion-capture systems provide natural and diverse behaviors [23], [24], [3], [6], [7], [8]. However, hand motion must be estimated and retargeted, and the resulting data do not directly contain robot actions. Teleoperation instead records executable actions by mapping human-hand motion to a robot hand [1], [2], [3], [4], [5], [25]. Immersive visual feedback and haptic devices can improve control [26], [27], [28], but these systems generally require a complete robot setup and ask the operator to coordinate many finger joints. Lower-dimensional interfaces

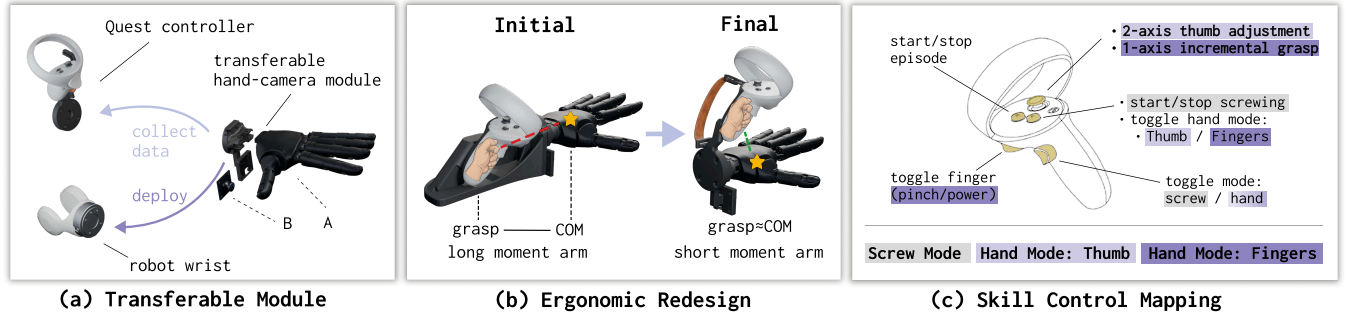


Fig. 2. **Transferable Hardware Interface and Control Mapping.** (a) *Transferable Module*: The Wuji hand (A) and fisheye camera (B) form a transferable hand-camera module that mounts to the Quest 2 controller for data collection and to the robot wrist for deployment. Reusing the same module preserves hand embodiment and camera-to-hand extrinsics. (b) *Ergonomic Redesign*: The final hardware interface places the operator’s grasp near the assembly’s center of mass and adds a support strap, shortening the moment arm and reducing wrist torque. (c) *Skill Control Mapping*: Quest 2 inputs start and stop demonstration episodes, select hand or screw mode, and modulate thumb, grasp, and screwing skills. The joystick adjusts thumb flexion and abduction in Thumb Mode and incrementally tightens or loosens the grasp in Finger Mode.

reduce this burden. HATO maps a small set of controller inputs to coordinated finger motions [29], while DexScrew uses a simulation-trained rotation policy as a skill during teleoperation [21]. These results motivate skill-level control, but neither system addresses portable collection with a high-DoF hand.

Portable Data Collection for Dexterous Hands. Recent systems enable dexterous data collection without a fixed robot setup. DexUMI, DEXOP, and DexEXO use wearable exoskeletons or passive hand mechanisms to constrain and measure the operator’s motion [10], [11], [14]. These systems efficiently collect demonstrations of contact-rich behaviors and reduce the human-to-robot kinematic gap. However, these passive collection mechanisms record hand configurations rather than both the commands and resulting states of the actuated hand used at deployment. Related systems improve kinematic alignment or tactile sensing through mechanically isomorphic and modular exoskeleton designs [30], [16]. Another line of work uses robot hands that operators carry or wear, reducing the contact-geometry gap and allowing robot-aligned data collection [31], [15]. RealDexUMI [12] follows the UMI principle of using the same end effector for collection and deployment. It also records robot commands, rather than inferring actions from measured states, but demonstrates this approach on a hand with six active DoFs. In contrast, DEXPORT targets a 20-DoF hand, for which joint-by-joint operation would be cumbersome. We use a compact set of skills to make the hand practical to operate while still recording the raw hand commands used to train the policy.

III. METHOD

In Sec. III-A, we introduce the hardware design of our portable data collection interface. Next, in Sec. III-B we describe our library of skills, designed to be intuitive and useful for learning dexterous policies. Finally, Sec. III-C explains how this data can be used for imitation learning.

A. Hardware Design

Transferable Hand-Camera Module. Our portable data-collection interface comprises a transferable hand-camera

module and a collection handle built around a Meta Quest 2 controller, as shown in Fig. 2(a). The module combines a five-fingered, 20-DoF Wuji robot hand with a wide-angle fisheye camera through a rigid 3D-printed flange. During collection, it attaches to the collection handle; during deployment, the same physical module mounts to the robot wrist. Reusing the module preserves the hand embodiment and camera-to-hand extrinsics across the two settings, so recorded hand targets transfer to the robot without retargeting. This satisfies the **Same Emb.** requirement. The mechanical design can be adapted to other hand-camera pairs by replacing the flange. We design flanges for both the Wuji 1 and Wuji 2.

Recorded Signals. During data collection, the Quest 2 controller provides the wrist’s 6-DoF trajectory, from which we construct end-effector motion targets. The fisheye camera captures the hand-object interaction and surrounding task context, while the hand controller records both the commanded 20-DoF joint targets and the resulting joint states. Each demonstration therefore carries the actions actually sent to the hand, rather than actions inferred from measured states, satisfying the **Action Data** requirement. The rigid connections keep the controller-to-hand and camera-to-hand transforms fixed, so the tracked motion and visual observations remain expressed relative to the same hand frame.

Ergonomic Design. The complete hand-held assembly weighs 0.955 kg, of which the non-hand hardware contributes 0.220 kg. To support this load while leaving the thumb and index finger free to operate the controls, we extend the Quest 2 battery cover into a collection handle that connects the controller to the transferable module. A strap across the back of the operator’s hand carries part of the load, reducing the sustained grip force.

Our initial design placed the controller behind the robot hand, offsetting the operator’s grip from the assembly’s center of mass and creating a long moment arm, as shown in Fig. 2(b). The final design positions the controller above the hand so that the grip lies near the center of mass. This shortens the moment arm and reduces the wrist torque that the operator must counteract.

B. Dexterous Control via Parameterized Skills

Control via skills is an effective way to operate high-dimensional robotic systems. For example, a humanoid can be commanded through a desired velocity rather than by specifying the motion of every joint. We apply the same principle to dexterous hands by separating the operator’s intent from the finger motions required to realize it. To this end, we design a compact set of skills that we find effective across a wide range of dexterous tasks. The skill interface is *parameterized* instead of discrete, allowing more wholistic control. For instance, the user controls the strength of the grasp by using the joystick to incrementally close or open the grasp, instead of snapping to a single fixed grasp.

Together, the skills keep all 20 joints of the hand available during collection without asking the operator to command each one, satisfying the **DoF** requirement that separates a dexterous interface from a simpler gripper. We implement this interface using a Meta Quest 2 controller, which provides two buttons, two continuous triggers, and a two-axis joystick. As illustrated in Fig. 2(c) and detailed below, these inputs allow the operator to select and modulate the skills.

- **Thumb Control.** The two joystick axes control thumb flexion and extension along one axis and abduction and adduction along the other, following HATO [29]. This allows the operator to position the thumb independently for different grasp geometries.
- **Power Grasp.** One joystick axis controls the coordinated closure of the four non-thumb fingers. Moving the joystick incrementally loosens or tightens the grasp, while releasing it holds the current target. Unlike continuous trigger control [29], this persistent command does not require the operator to maintain pressure throughout a grasp. It is primarily used to wrap the hand around larger objects.
- **Pinch Grasp.** The same incremental command controls only the index finger, which closes against the independently positioned thumb. This skill supports small objects and grasps that require precise fingertip contact.
- **Two-Finger Grasp.** The index and middle fingers open or close together while the remaining fingers retain their targets. This skill supports multi-stage behaviors such as multi-object grasping and two-finger grasp-and-pull motions, without requiring the operator to coordinate the two fingers individually.
- **In-Hand Rotation.** A button invokes a learned closed-loop policy that generates the repeated finger gaiting required to rotate a grasped tool in hand. The policy controls the fingers while the operator continues to control the global motion of the hand.

Learning the In-Hand Rotation Skill. We learn the in-hand rotation skill in simulation. This relieves the operator of difficult low-level, in-hand manipulation, especially without finger-level haptic feedback. Similar to [21], we train a closed-loop screwing policy in simulation with PPO [32] in Isaac Lab [33].

We model the task as rotating a cylindrical handle on a revolute joint. In the real-world, the operator can make wrist

adjustments, but we simplify the simulation environment so hand base is fixed, with the policy commanding fingers only. The policy learns delta actions over 12 actuated joints, with the ring and pinky finger fixed. The policy observes a three-step history, including measured joint positions, commanded targets, and the previous action. We learn a privileged critic which has access to the handle’s rotation angle.

We follow rewards and domain randomization similar to [21], which includes a handle rotation, fingertip proximity, bounded action smoothness, initial grasp proximity, joint torque, and mechanical work rewards. We randomize object mass, center of mass, friction, restitution, finger PD gains, initial grasp, and random disturbance forces. We train for 1.18×10^8 environment steps across 4096 environments. During collection, the actor, a three-layer MLP, runs at 20 Hz on the hand’s joint encoders while the operator retains control of the wrist.

C. Policy Learning

Raw-Action Supervision. The skills are used only to assist the operator during data collection. The learned policy directly predicts wrist poses and 20-DoF hand joint targets. At each time step, we record the raw 20-DoF target produced by the active skill and sent to the hand controller, together with the resulting measured joint state. The policy receives no skill identity, skill parameter, or task-phase label. Instead, it is trained to predict the commanded joint targets directly. Thus, the skills structure how demonstrations are collected without constraining the representation learned by the policy. Switching the action label from the commanded target to the measured joint state is the single change that defines our state-as-action baseline in Sec. IV; every other part of the recipe below is shared.

Observation and Action Spaces. We train a diffusion policy [34] with a convolutional U-Net denoiser and a fine-tuned DINOv2 visual encoder. Images are augmented with a random crop and resize, color jitter, random grayscale, and Gaussian blur. The vision encoder is optimized with a reduced learning rate, one-tenth that of the remaining network. The policy observes the two most recent time steps, each consisting of a 224×224 RGB image, the end-effector position represented relative to the current chunk start [9], and the measured 20-DoF hand state.

The policy predicts a 16-step action chunk containing end-effector poses relative to the start of the chunk and absolute 20-DoF hand targets. Each action step is 29-dimensional: 9 for the wrist (3 translation and a 6D rotation) and 20 for the hand. For bimanual tasks, we concatenate the observations and actions of the two hands without otherwise changing the architecture. We train 50 DDIM steps and sample with 10 steps. Following HATO [29], policy inference runs asynchronously to maintain smooth control.

Deployment Calibration. The collected end-effector actions are expressed in the Quest controller frame, whereas deployment actions must be expressed in the robot end-effector frame. We estimate the fixed transform between these frames

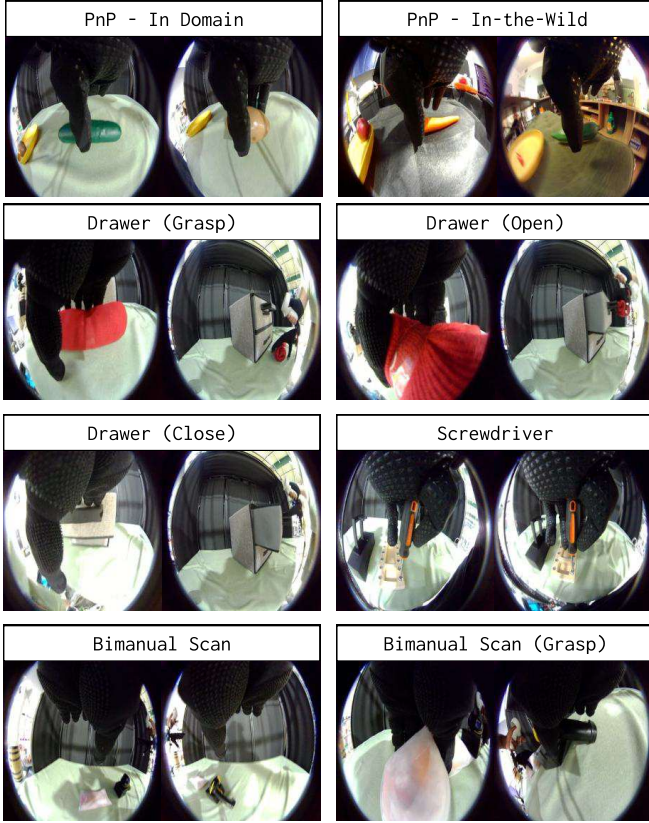


Fig. 3. **Data Visualization.** We show fisheye camera views recorded during data collection. For our pick-and-place task (PnP), we collect in-domain and in-the-wild data. For drawer stowing, we add a third-person camera because the grasped object occludes much of the scene. We show both camera views at three stages of the trajectory. For bimanual scanning and drawer stowing, the operator is occasionally visible, but we do not post-process the data to remove the operator.

from paired controller and end-effector poses collected with the device mounted on the robot. Concretely, we bolt the module to the robot flange and record N paired samples of the flange pose R_i , and the controller pose Q_i , from the headset. Each sample satisfies $Q_i = X R_i M$, where X is the unknown robot base to headset transform and M is the controller to flange transform, which is the mount transform we want. Any two samples eliminate X ,

$$R_i^{-1} R_j = M (Q_i^{-1} Q_j) M^{-1}, \quad (1)$$

so N poses give up to $N(N-1)/2$ pairwise constraints on exactly the conjugation that deployment applies, $A^{\text{ee}} = M A^{\text{quest}} M^{-1}$. We fit the full 6-DoF M by least squares on (1), independently for each of the two arms. We apply this transform to the predicted end-effector motions at deployment; the 20-DoF hand targets require no conversion.

IV. EXPERIMENTS

We evaluate DEXPORT through three research questions:

- **Q1:** Do the hand commands recorded by DEXPORT enable policies that succeed on diverse dexterous tasks?
- **Q2:** Does co-training with demonstrations collected in unstructured environments improve robustness to unseen objects and cluttered scenes?

- **Q3:** How does the collection throughput of DEXPORT compare with teleoperation and human-hand demos?

A. Experimental Setup

Robot Platform. We deploy policies on Tianji Marvin, a bimanual robot with two 7-DoF arms. The hand-camera modules used for collection (Sec. III-A) mount directly to its wrists, so the hand embodiment and camera-to-hand extrinsics are identical across collection and deployment. Hand targets therefore need no retargeting, and the only correction at deployment is the fixed controller-to-end-effector transform (Sec. III-C).

We showcase the modularity of our system by using different hands: the Wuji 1 hand for Pick-and-Place and Screwing, and the Wuji 2 hand for Drawer Stowing and Bimanual Barcode Scanning.

Tasks. We evaluate four tasks that exercise complementary forms of dexterity. Fig. 4 shows a successful rollout of each, annotated with the phases we use to score partial success and with the grasp types that emerge:

- **Vegetable Pick-and-Place.** The robot sequentially picks and places two vegetables into a plate. Variations in object size and shape require the policy to select between pinch and power grasps without explicit grasp labels.
- **Screwing.** The robot maintains a stable grasp while rotating a screwdriver in hand, testing sustained contact and repeated finger gaiting.
- **Drawer Stowing.** Without releasing a grasped object, the robot opens a drawer with the same hand, places the object inside, and closes the drawer. This task couples object retention with articulated-object manipulation.
- **Bimanual Barcode Scanning.** One hand grasps and presents an object while the other positions and operates a barcode scanner. Success requires coordinated bimanual positioning of the object and scanner.

Observations. Policies observe the fisheye view from the hand-camera module, shown for every task in Fig. 3. For drawer stowing we add a third-person camera, because the grasped object occludes much of the scene while the drawer is pulled open. The operator is occasionally visible in the recorded frames, and we apply no post-processing.

Datasets. We collect 1,605 laboratory demonstrations across the four tasks: 700 for vegetable pick-and-place, 200 for screwdriver manipulation, 380 for drawer stowing, and 325 for bimanual barcode scanning. We additionally collect 1,000 pick-and-place demonstrations in unstructured apartment and office environments. The co-trained policies use both the laboratory and in-the-wild pick-and-place datasets.

Training Variants and Baselines. To answer Q1, we compare policies trained on the commanded hand targets recorded by DEXPORT against a state-as-action baseline that instead uses measured hand states as action labels. To answer Q2, we compare the laboratory-only pick-and-place policy with two co-trained variants. To answer Q3, we compare the number of trajectories collected per 30 minutes using DEXPORT, teleoperation, and human-hand demonstrations.

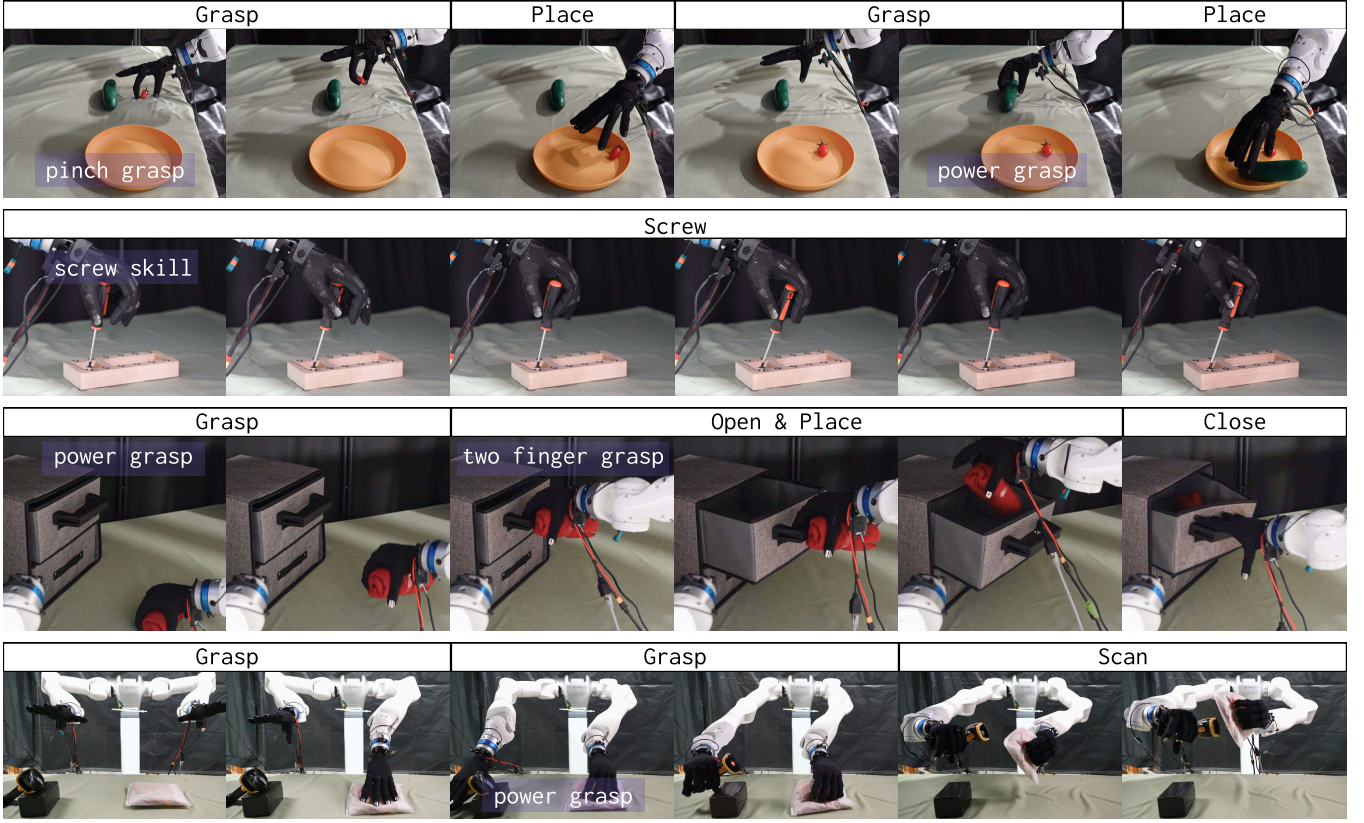


Fig. 4. **Task Rollouts.** We show successful rollouts annotated with the phases of each task. We use these phases to score partial success. We also annotate grasp types that emerge in the rollouts, such as a pinch grasp for the cherry tomato and a power grasp for the cucumber. These annotations describe observed behavior. The policy predicts raw joint targets without skill labels.

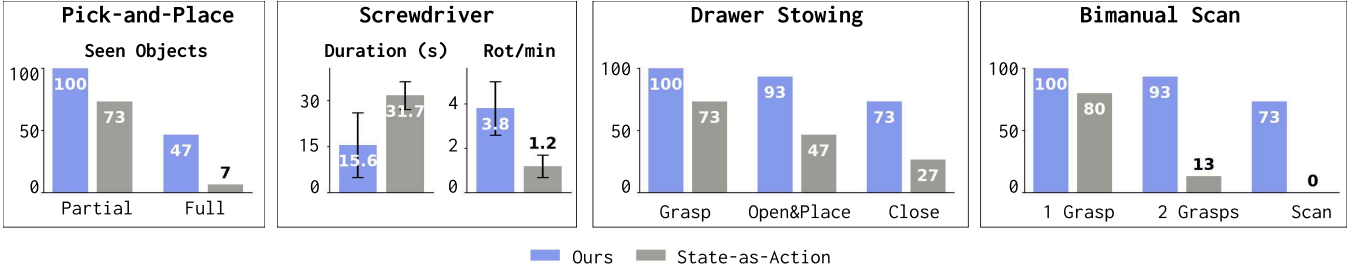


Fig. 5. **Policy Learning Results.** We show imitation learning performance across four tasks. Policies trained on recorded hand commands achieve higher success rates and faster screwdriver rotation than the state-as-action baseline. Partial success for pick-and-place denotes grasping and placing one object. For drawer stowing and bimanual scanning, we report success rates for the task phases shown in Fig. 4. A phase counts as successful only if all preceding phases also succeed. Each policy is evaluated for 15 episodes per task.

Evaluation Protocol. We evaluate each policy for 15 episodes per task or evaluation condition. For vegetable pick-and-place, we test seen objects, unseen objects, and scenes with distractors; partial success denotes placing one of the two objects, while full success denotes placing both. For drawer stowing and bimanual scanning, we report completion of the task phases visualized in Fig. 4. For screwdriver manipulation, we report the average hold duration in seconds and rotations per minute.

B. Policy Learning Across Dexterous Tasks (Q1)

Fig. 5 reports imitation learning results across the four tasks. Policies trained on DEXPORT demonstrations acquire all of the target behaviors: they select between pinch and power grasps without grasp labels, hold an object while

pulling open a drawer with the same hand, gait the fingers to rotate a screwdriver in hand, and coordinate two arms to present an object to a scanner.

The state-as-action baseline, which uses measured joint states as action labels, is worse on every task. For instance, we find that the state-as-action baseline is unable to sufficiently grasp the object. For the scanning task, grasping the package is nearly impossible for the baseline, whereas with robot commands, we exert the necessary force to hold the item tightly. For the drawer task, the laundry often falls during the drawer opening segment. We visualize this in Fig. 7. Because a stage cannot succeed once an earlier stage has failed, the deficit compounds rather than appearing as a constant offset: on bimanual scanning the baseline grasps

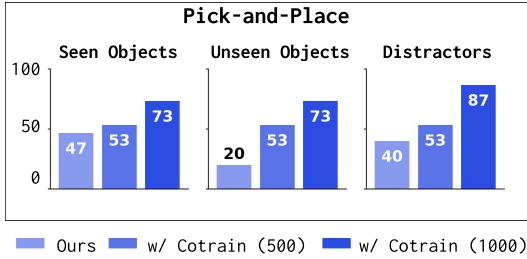


Fig. 6. **In-the-Wild Co-Training.** One benefit of DEXPORT’s design is its portability. For our pick-and-place task, we collect 1,000 in-the-wild demonstrations (visualized in Fig. 3) and co-train with the 700 laboratory demonstrations. We find that increasing the number of cotrained trajectories increases the success rate of the policy in every condition. In the distractor setting, full success rises from 40% to 87%.

	DEXPORT	Skill-Teleop	Glove-Teleop	Human hand
Traj. / 30 min	122	68	86	275
Portable	✓	✗	✗	✓
Same hand emb.	✓	✓	✓	✗
Action data	✓	✓	✓	✗

TABLE II: **Data Collection Throughput.** Trajectories collected during 30 minutes of continuous operation on the unimanual pick-and-place task. For Glove Teleoperation, we use the Wuji Glove for hand tracking and the VR controller for end-effector control. For Skill Teleoperation, we use the same skill interface as DEXPORT to command the hand and VR tracking to control the end effector. DEXPORT is the only method compared here that combines portable collection with the deployed hand’s embodiment and its commanded actions.

the first object in 80% of episodes, holds both in 13%, and never completes a scan. The screwdriver task exposes the same weakness through a different metric. While the baseline holds the tool more than twice as long as our policy (31.7 s vs. 15.6 s), it rotates it at a third of the rate and is much less effective. These results highlight the importance of **Action Data**, since access to robot commands for the deployment hand is crucial for a higher success rate. This is a major limitation for data collection interfaces based on exoskeletons or hardware without action data.

C. Co-Training with In-the-Wild Data (Q2)

We collect a large in-the-wild pick-and-place dataset across several sites, including an apartment room and a range of office tables. DEXPORT requires neither a robot arm nor external tracking infrastructure, thus enabling this portable data collection. We use the same training objects, with the difference being diversity in scene, background, and lighting rather than in object identity, as Fig. 3 illustrates.

Fig. 6 shows that full success improves in every condition and grows with the amount of in-the-wild data. With 1,000 in-the-wild demonstrations, full success rises from 40% to 87% under distractors and from 20% to 73% on unseen objects, so the benefit transfers to new objects. It also improves on seen objects, from 47% to 73%. App. A reports a lower data regime.

D. Data Collection Throughput (Q3)

We measure throughput as the number of trajectories collected during 30 minutes of continuous operation on the unimanual pick-and-place task, comparing DEXPORT against



Fig. 7. **State-as-Action Failure.** Policies trained with measured joint states as action labels produce weak grasps and fail to hold onto objects. *Top:* the baseline drops the package when presenting it to the scanner. *Bottom:* the baseline cannot retain the laundry once it begins to pull the drawer open.

bare-hand human demonstration and teleoperation. For Glove Teleoperation, the operator uses the Wuji Glove for hand tracking and the VR controller for the end-effector. For Skill Teleoperation, we use the VR controller for end-effector control and the same hand skills as DEXPORT. The operator is expert-level for each style of data collection.

Table II shows that DEXPORT collects roughly $1.4\times$ as many trajectories as Glove Teleoperation and $1.8\times$ as many trajectories as Skill Teleoperation. While Skill Teleoperation is a simpler interface, reducing the operator’s cognitive burden, Glove Teleoperation still remains faster because the hand tracking is more responsive and flexible. Generally, both are much slower than DEXPORT. When teleoperating, the operator is out of the scene, removing the direct visual and haptic cues. This makes approaching an object and performing precise tasks, such as pinch grasping a small object, much slower and more difficult. Human hand data collection remains about $2.3\times$ faster than DEXPORT, but we regard it as an upper bound on speed rather than a competing source of training data, as human hand data provides neither the deployed hand’s embodiment nor its commanded actions.

V. CONCLUSIONS

We presented DEXPORT, a portable interface for collecting demonstrations with high-DoF dexterous hands. The system uses the same hand hardware for collection and deployment, preserving its contact geometry while recording its commanded targets and resulting joint states. A compact set of skills makes the hand practical to operate by separating task-level intent from detailed finger coordination. These skills assist only the demonstrator; the learned policy directly predicts the raw hand commands without skill or task-phase labels. In our experiments, DEXPORT collects 122 trajectories per 30 minutes, roughly $1.4\times$ the rate of teleoperation. Policies trained on the resulting demonstrations succeed on all four tasks and consistently outperform a state-as-action baseline, which never completes a bimanual barcode scan. With 700 laboratory demonstrations, co-training with in-the-wild data further improves pick-and-place success in every evaluation condition. Together, these design choices provide a practical and scalable solution for collecting portable, executable demonstrations for dexterous robot learning.

Limitations. Despite its ergonomic design, the weight of the device limits uninterrupted use and rotation-heavy motions.

As the weight of the hand is the primary bottleneck, we expect that future hardware developments for lighter and smaller hands can address this. Our skills also cover only part of the space of dexterous behaviors, and future work may explore extending them with additional skills learned in simulation, such as other in-hand manipulation skills, learned grasps, and more.

ACKNOWLEDGMENT

The authors would like to thank Keyshawn King and Volodymyr Behma for data collection and support, and Andrew Wright for initial hardware designs. We also thank Tingfan Wu, Anas Houssaini, Jensen Gao, and the FAR team for feedback and support.

REFERENCES

- [1] 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,” 2019. [Online]. Available: <https://arxiv.org/abs/1910.03135>
- [2] 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,” 2023. [Online]. Available: <https://arxiv.org/abs/2307.04577>
- [3] A. Sivakumar, K. Shaw, and D. Pathak, “Robotic telekinesis: Learning a robotic hand imitator by watching humans on YouTube,” 2022. [Online]. Available: <https://arxiv.org/abs/2202.10448>
- [4] C. Wang, Y. Feng, H. Fang, S. Xia, L. Yang, C. Wen, and C. Lu, “AnyDexRT: Calibration-free dexterous hand retargeting with few-shot human guidance,” 2026. [Online]. Available: <https://arxiv.org/abs/2607.08341>
- [5] S. Yang, M. Liu, Y. Qin, R. Ding, J. Li, X. Cheng, R. Yang, S. Yi, and X. Wang, “ACE: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.11805>
- [6] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, “DexCap: Scalable and portable mocap data collection system for dexterous manipulation,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.07788>
- [7] T. Tao, M. K. Srirama, J. J. Liu, K. Shaw, and D. Pathak, “DexWild: Dexterous human interactions for in-the-wild robot policies,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.07813>
- [8] I. Guzey, H. Qi, J. Urain, C. Wang, J. Yin, K. Bodduluri, M. Lambeta, L. Pinto, A. Rai, J. Malik, T. Wu, A. Sharma, and H. Bharadhwaj, “Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations,” 2025. [Online]. Available: <https://arxiv.org/abs/2511.16661>
- [9] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, “Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.10329>
- [10] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song, “DexUMI: Using human hand as the universal manipulation interface for dexterous manipulation,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.21864>
- [11] H.-S. Fang, B. Romero, Y. Xie, A. Hu, B.-R. Huang, J. Alvarez, M. Kim, G. Margolis, K. Anbarasu, M. Tomizuka, E. Adelson, and P. Agrawal, “DEXOP: A device for robotic transfer of dexterous human manipulation,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.04441>
- [12] C. Xu, Y. Jiang, J. Huan, Y. Fu, H. Zhou, W. Yuan, J. Yu, W. Zhang, H. Yuan, and Z. Lu, “RealDexUMI: A wearable universal manipulation interface for dexterous robot learning,” 2026. [Online]. Available: <https://arxiv.org/abs/2606.06033>
- [13] X. Chen, Y. Pan, M. Li, and X. Ding, “DexViTac: Collecting human visuo-tactile-kinematic demonstrations for contact-rich dexterous manipulation,” 2026. [Online]. Available: <https://arxiv.org/abs/2603.17851>
- [14] A. Zhu, M. Zhu, B. J. Kim, J. V. S. H. Ramos, Y. Shi, Y. Wu, R. Dhar, F. Yang, R. Hou, H. Fang, Q. Wang, Y. Cui, and D. W. Hong, “DexEXO: A wearability-first dexterous exoskeleton for operator-agnostic demonstration and learning,” 2026. [Online]. Available: <https://arxiv.org/abs/2603.17323>
- [15] J. Koh, H. Jung, N. Kim, W. Ko, and C. Nam, “DEX-Mouse: A low-cost portable and universal interface with force feedback for data collection of dexterous robotic hands,” 2026. [Online]. Available: <https://arxiv.org/abs/2604.15013>
- [16] X. Chao, S. Mu, Y. Liu, S. Li, C. Lyu, X.-P. Zhang, and W. Ding, “Exo-ViHa: A cross-platform exoskeleton system with visual and haptic feedback for efficient dexterous skill learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.01543>
- [17] S. Song, A. Zeng, J. Lee, and T. Funkhouser, “Grasping in the wild: Learning 6DoF closed-loop grasping from low-cost demonstrations,” 2019. [Online]. Available: <https://arxiv.org/abs/1912.04344>
- [18] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto, “Visual imitation made easy,” 2020. [Online]. Available: <https://arxiv.org/abs/2008.04899>
- [19] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, “On bringing robots home,” 2023. [Online]. Available: <https://arxiv.org/abs/2311.16098>
- [20] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, “In-hand object rotation via rapid motor adaptation,” 2022. [Online]. Available: <https://arxiv.org/abs/2210.04887>
- [21] E. Hsieh, W.-H. Hsieh, Y.-J. Wang, T. Lin, J. Malik, K. Sreenath, and H. Qi, “Learning dexterous manipulation skills from imperfect simulations,” 2025. [Online]. Available: <https://arxiv.org/abs/2512.02011>
- [22] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg, “Crossing the human-robot embodiment gap with sim-to-real RL using one human demonstration,” 2025. [Online]. Available: <https://arxiv.org/abs/2504.12609>
- [23] K. Shaw, S. Bahl, and D. Pathak, “VideoDex: Learning dexterity from internet videos,” 2022. [Online]. Available: <https://arxiv.org/abs/2212.04498>
- [24] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, “DexMV: Imitation learning for dexterous manipulation from human videos,” 2021. [Online]. Available: <https://arxiv.org/abs/2108.05877>
- [25] H. Jiang, Y. Wu, Y. Wang, G. S. Sukhatme, and D. Seita, “Concurrent prehensile and nonprehensile manipulation: A practical approach to multi-stage dexterous tasks,” 2026. [Online]. Available: <https://arxiv.org/abs/2603.11655>
- [26] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, “Open-TeleVision: Teleoperation with immersive active visual feedback,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.01512>
- [27] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang, “Bunny-VisionPro: Real-time bimanual dexterous teleoperation for imitation learning,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.03162>
- [28] H. Zhang, S. Hu, Z. Yuan, and H. Xu, “DOGlove: Dexterous manipulation with a low-cost open-source haptic force feedback glove,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.07730>
- [29] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik, “Learning visuotactile skills with two multifingered hands,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.16823>
- [30] J. Du, J. Ren, Q. Yu, N. Zhang, Y. Deng, X. Wei, Y. Liu, G. Gu, and X. Zhu, “MILE: A mechanically isomorphic hand exoskeleton and visuotactile robotic hand for data collection in dexterous manipulation,” 2025. [Online]. Available: <https://arxiv.org/abs/2512.00324>
- [31] D. Wei and H. Xu, “A wearable robotic hand for hand-over-hand imitation learning,” 2023. [Online]. Available: <https://arxiv.org/abs/2309.14860>
- [32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [33] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Muñoz, X. Yao, R. Zurbrugg, N. Rudin, L. Wawrzyniak, M. Rakhsha, A. Denzler, E. Heiden, A. Borovicka, O. Ahmed, I. Akinola, A. Anwar, M. T. Carlson, J. Y. Feng, A. Garg, R. Gasoto, L. Gulich, Y. Guo, M. Gussert, A. Hansen, M. Kulkarni, C. Li, W. Liu, V. Makovychuk, G. Malczyk, H. Mazhar, M. Moghani, A. Murali, M. Noseworthy, A. Poddubny, N. Ratliff, W. Rehberg, C. Schwärke, R. Singh, J. L. Smith, B. Tang, R. Thaker, M. Trepte, K. Van Wyk,

- F. Yu, A. Millane, V. Ramasamy, R. Steiner, S. Subramanian, C. Volk, C. Chen, N. Jawale, A. V. Kuruttukulam, M. A. Lin, A. Mandlekar, K. Patzwaldt, J. Welsh, H. Zhao, F. Anes, J.-F. Lafleche, N. Moëgne-Loccoz, S. Park, R. Stepinski, D. Van Gelder, C. Amevor, J. Carius, J. Chang, A. H. Chen, P. de Heras Ciecchowski, G. Daviet, M. Mohajerani, J. von Mural, V. Reutsky, M. Sauter, S. Schirm, E. L. Shi, P. Terdiman, K. Vilella, T. Widmer, G. Yeoman, T. Chen, S. Grizan, C. Li, L. Li, C. Smith, R. Wiltz, K. Alexis, Y. Chang, D. Chu, L. J. Fan, F. Farshidian, A. Handa, S. Huang, M. Hutter, Y. Narang, S. Pouya, S. Sheng, Y. Zhu, M. Macklin, A. Moravanszky, P. Reist, Y. Guo, D. Hoeller, and G. State, “Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2511.04831>
- [34] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.04137>
- [35] C. Schwarke, M. Mittal, N. Rudin, D. Hoeller, and M. Hutter, “RSL-RL: A learning library for robotics research,” 2025. [Online]. Available: <https://arxiv.org/abs/2509.10771>
- [36] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, “Asymmetric actor critic for image-based robot learning,” 2017. [Online]. Available: <https://arxiv.org/abs/1710.06542>
- [37] M. Oquab, T. Darce, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, “DINOv2: Learning robust visual features without supervision,” 2023. [Online]. Available: <https://arxiv.org/abs/2304.07193>
- [38] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” 2017. [Online]. Available: <https://arxiv.org/abs/1709.07871>
- [39] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” 2020. [Online]. Available: <https://arxiv.org/abs/2010.02502>
- [40] C. M. Kim, B. Yi, H. Choi, Y. Ma, K. Goldberg, and A. Kanazawa, “PyRoki: A modular toolkit for robot kinematic optimization,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.03728>

APPENDIX

A. Scaling In-the-Wild Data

Fig. 6 shows the benefit of scaling in-the-wild data with 700 laboratory demonstrations. We also evaluate pick-and-place policies trained on 500 laboratory demonstrations with 0, 500, or 1,000 in-the-wild demonstrations.

From Table III, co-training with in-the-wild data improves full success on seen objects and with distractors for both the 500 in-scene and 700 in-scene demonstration settings. However, for 500 demos, in-the-wild data does not improve performance in the unseen object evaluation setting. Because the in-the-wild demonstrations use the same objects as the laboratory data, they diversify scenes rather than object geometry, and we hypothesize that transferring this robustness to new objects requires a sufficiently large laboratory dataset.

Training Details. For the policies, we cotrain with 50% of the batch drawn from in-the-wild data. We train every policy for the same number of epochs over its laboratory demonstrations (≈ 250 epochs).

B. Hardware and Data Collection

Hardware. The hand-camera module pairs a Wuji hand with a USB board camera with a 200° fisheye lens. On the Wuji 1, hand targets pass through a 5 Hz low-pass filter and are interpolated by the hand firmware at 1 kHz, with a per-joint torque limit of 0.5 N·m.

Recording and Synchronization. A single recording loop runs at 30 Hz. The camera, hand, and Quest controller are

# Demos	# In-the-Wild	Success (%)		
		Seen	Unseen	Distractors
500	0	0	20	0
500	500	33	13	53
500	1,000	47	7	40
700	0	47	20	40
700	500	53	53	53
700	1,000	73	73	87

TABLE III: **Scaling In-the-Wild Data.** Pick-and-place success rates (%) of policies trained on 500 or 700 in-scene laboratory demonstrations (# Demos) and 0, 500, or 1,000 in-the-wild demonstrations (# In-the-Wild). We evaluate on seen objects, unseen objects, and scenes containing distractors. In-the-wild data improves full success on seen objects and with distractors in both data regimes, but improves success on unseen objects only with 700 laboratory demonstrations.

Hyperparameter	Value
RL library	RSL-RL [35]
Parallel environments	4096
Rollout length	24 steps
Iterations	1200 (1.18×10^8 steps)
Learning rate	5×10^{-3} , adaptive (KL target 0.01)
Discount γ / GAE λ	0.99 / 0.95
Clip ratio	0.2
Value loss coefficient	1.0 (clipped)
Entropy coefficient	0
Epochs / minibatches	5 / 4
Max gradient norm	1.0
Initial action std.	1.0
Actor and critic MLPs	[512, 256, 128], ELU
Observation normalization	Running mean and std.
Control rate / episode	20 Hz / 40 s

TABLE IV: **PPO Hyperparameters.** Settings for training the in-hand rotation skill in Isaac Lab [33]. The simulation runs at 200 Hz with 10 physics steps per policy step.

each read in a background thread. Because the camera and the hand use same 30 Hz loop during collection and deployment, we expect their latencies to roughly match across the two settings. We therefore do not measure stream latencies or temporally align the streams in post-processing, as UMI [9] does. At deployment, the asynchronous inference in App. E discards actions whose time has already passed.

C. In-Hand Rotation Skill

Environment. We model the screwdriver as a 12-sided prism that rotates about its axis on a revolute joint. The hand base is fixed in simulation. An episode terminates early if the thumb, index, or middle fingertip moves more than 12 cm from the handle. Tables IV and V list the training hyperparameters and reward terms, and Sec. III-B lists the randomized parameters.

Observations and Actions. The actor observes a 147-D vector: the 20 measured joint positions and 20 joint targets over the last three steps, the previous action, and the fixed pose of the hand base. Following the asymmetric actor-critic approach [36], the critic observes a 127-D privileged state without history: joint positions, targets, and velocities, the hand base pose, fingertip positions, the handle’s position,

Term	Weight	Definition
Handle rotation	6.0	$\text{clip}(\dot{\theta}, -4, 4)$
Fingertip proximity	2.0	$\text{clip}(1 - \bar{d}/0.035, 0, 1)$
Action smoothness	1.0	$\exp(-(\overline{a_t - a_{t-1}})^2/0.5^2)$
Pose deviation	-6.0	$\ q - q_0\ ^2$
Over-speed	-0.3	$\max(\dot{\theta} - 10, 0)$
Joint torque	-0.1	$\ \tau\ ^2$
Mechanical work	-0.01	$(\sum_j \tau_j \dot{q}_j)^2$

TABLE V: **Rotation Skill Rewards.** $\dot{\theta}$ is the handle’s angular velocity in rad/s, $\bar{d}$ is the mean horizontal distance in meters from the thumb, index, and middle fingertips to the handle axis, q and q_0 are the current and initial joint position vectors, and τ is the joint torque vector. a_t is the policy action at step t , $\dot{q}_j$ is the velocity of joint j , and the overline in the action smoothness term denotes the mean over joints.

Hyperparameter	Value
Vision encoder	DINOv2 ViT-S/14 [37], fine-tuned
Denoiser	1D conditional U-Net [34], FiLM [38]
U-Net channels per level	
Pick-and-place in-the-wild	(512, 1024, 2048)
Pick-and-place baseline	(256, 512, 1024)
Screwdriver	(512, 1024, 2048)
Drawer stowing	(512, 1024, 2048)
Bimanual scanning	(512, 1024, 2048)
Noise scheduler	DDIM [39], squared cosine
Prediction target	Noise (ϵ)
Train / inference steps	50 / 10
Optimizer	AdamW, $\beta = (0.9, 0.999)$
Weight decay	0.01
Learning rate	1×10^{-4} (vision encoder 1×10^{-5})
LR schedule	Cosine, 500 warmup steps
Batch size	64
Gradient steps	5×10^5
EMA decay	$\min(1 - (1 + t)^{-3/4}, 0.9999)$
Random crop	212 px, resized to 224
Color jitter	(0.3, 0.4, 0.5, 0.08)
Random grayscale	$p = 0.1$
Gaussian blur	$k = 5, \sigma \in [0.1, 1.5], p = 0.2$

TABLE VI: **Diffusion Policy Hyperparameters.** Settings are shared across tasks except for the task-specific U-Net channel sizes listed above. Color jitter lists the brightness, contrast, saturation, and hue ranges. At inference, images are center-cropped to 212 px and resized to 224 px.

rotation angle, and angular velocity, the applied joint torques, and the previous action. The policy outputs a 20-D action $a \in [-1, 1]^{20}$, and each joint target moves by $0.04a$ rad per step within the joint limits, which is a scaling factor inspired by [21]. The ring and pinky fingers are fixed, and the policy controls the remaining 12 joints.

Real-World Execution. During collection, the actor runs at 20 Hz on joint positions measured by the hand’s encoders and holds its targets between policy steps, while the operator controls the wrist through the collection handle.

D. Policy Learning

Architecture. Table VI lists the hyperparameters of the diffusion policy in Sec. III-C. The end-effector pose is represented relative to the current end-effector frame, with a 3-D translation and a 6-D rotation. Each predicted wrist

action is a relative pose with respect to the start of the chunk, and each hand action is an absolute 20-D target. Bimanual policies concatenate the observations and actions of both hands and additionally observe the relative pose between them. All inputs and targets are normalized using their 2nd and 98th percentiles.

E. Deployment

Mount Calibration. To estimate the controller-to-flange transform M in (1), we bolt the module to each robot flange and jog the arm through about 30 poses that tilt the flange axis by at least 30° , pausing 1.5 s at each pose before recording the paired flange and controller poses. We initialize M with a closed-form hand-eye solution and refine it by nonlinear least squares.

Control. The policy runs at 30 Hz, matching the collection rate. Following HATO [29], inference runs asynchronously: each new 16-step chunk is aligned to the control tick at which its observation was taken, the first 8 steps are kept, steps whose time has passed are dropped, and the hand and wrist commands at each tick average all chunks that cover it. We sample each chunk with 10 DDIM steps. Predicted wrist motions are converted to flange targets with the mount transform and solved for the 7-DoF arm with Levenberg-Marquardt inverse kinematics in PyRoki [40].