arXiv:2603.22264v1 [cs.RO] 23 Mar 2026

UniDex: A Robot Foundation Suite for Universal Dexterous Hand Control from Egocentric Human Videos

Gu Zhang[1,2,*†], Qicheng Xu[1,*], Haozhe Zhang[1,*], Jianhan Ma[2,*], Long He[1,*], Yiming Bao[1,*], Zeyu Ping[3], Zhecheng Yuan[1,2], Chenhao Lu[1], Chengbo Yuan[1], Tianhai Liang[1], Xiaoyu Tian[1], Maanping Shao[1], Feihong Zhang[1], Mingyu Ding[4], Yang Gao[1,2], Hao Zhao[1], Hang Zhao[1,2], Huazhe Xu[1,2]

[1] Tsinghua University [2] Shanghai Qizhi Institute
[3] Sun Yat-sen University [4] The University of North Carolina at Chapel Hill
*Core Contributors †Project Lead

https://unidex-ai.github.io/

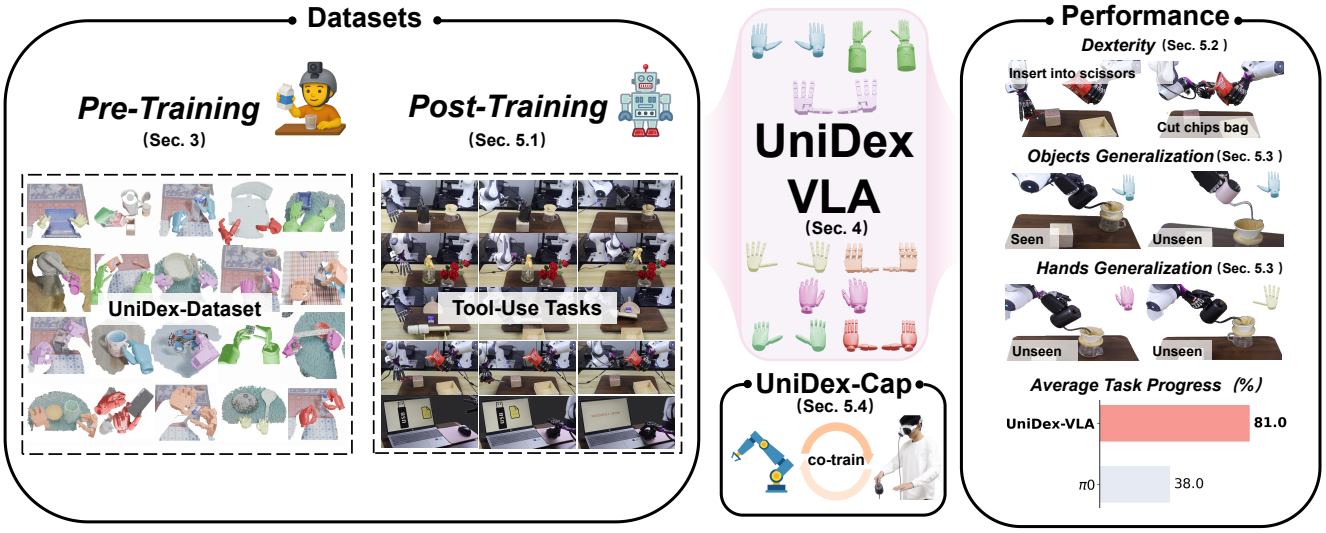


Figure 1. We introduce **UniDex**, a robot foundation suite for heterogeneous dexterous hand embodiments. We first curate UniDex-Dataset from egocentric human videos to obtain a diverse, robot-centric dataset for large-scale pretraining. Building on this, we train UniDex-VLA, a unified 3D VLA model that is finetuned with task demonstrations and evaluated on challenging real-world tool-use tasks. The policy exhibits strong dexterous capabilities, zero-shot object and cross-hand generalization and significantly outperforming existing VLA baselines. In addition, we design a practical setup, UniDex-Cap to support human–robot data co-training, further reducing the data cost.

Abstract

Dexterous manipulation remains challenging due to the cost of collecting real-robot teleoperation data, the heterogeneity of hand embodiments, and the high dimensionality of control. We present UniDex, a robot foundation suite that couples a large-scale robot-centric dataset with a unified vision–language–action (VLA) policy and a practical human-data capture setup for universal dexterous hand control. ***First****, we construct UniDex-Dataset, a robot-centric dataset over 50K trajectories across eight dexterous hands (6–24 DoFs), derived from egocentric human video datasets. To transform human data into robot-executable trajectories, we employ a human-in-the-loop retargeting procedure to align fingertip trajectories while preserving plausible hand–object contacts, and we operate on explicit 3D pointclouds with human hands masked to narrow kinematic and visual gaps.* ***Second****, we introduce the Function–Actuator–Aligned Space (FAAS), a unified action space that maps functionally similar actuators to shared coordinates, enabling cross-hand transfer. Leveraging FAAS as the action parameterization, we train UniDex-VLA, a 3D VLA policy pretrained on UniDex-Dataset and finetuned with task demonstrations.* ***In addition****, we build UniDex-*

* Core Contributors: see contribution list here.

Cap, a simple portable capture setup that records synchronized RGB-D streams and human hand poses and converts them into robot-executable trajectories to enable human–robot data co-training that reduces reliance on costly robot demonstrations. On challenging tool-use tasks across two different hands, UniDex-VLA achieves 81% average task progress and outperforms prior VLA baselines by a large margin, while exhibiting strong spatial, object, and zero-shot cross-hand generalization. Together, UniDex-Dataset, UniDex-VLA, and UniDex-Cap provide a scalable foundation suite for universal dexterous manipulation.

1. Introduction

In recent years, learning from demonstrations [6, 7, 13, 27, 58, 67, 74] has become the de facto paradigm for visuomotor control, enabling robots to acquire complex skills and motion patterns. However, achieving general, human-level manipulation under supervised learning remains challenging. Collecting real-robot demonstrations is labor-intensive and scales poorly, creating a persistent data bottleneck. Moreover, most robot foundation policies focus on parallel-jaw grippers, while foundation models for dexterous hands remain scarce—even though everyday tool-use often requires dexterous hands and many tasks (e.g., using scissors or spray bottles) are infeasible with grippers.

Simply porting gripper-based VLA designs to dexterous hands is insufficient. Building foundation models for dexterous hands is substantially more challenging than for grippers. The key difficulties are: (i) dexterous hand data are harder to collect than gripper data, and large, broadly usable pretraining datasets remain limited; (ii) dexterous hands vary widely in DoFs, morphology, kinematics, and appearance, leading to poor transfer of data and policies across hands; and (iii) dexterous hand control is inherently high-dimensional, demanding expressive action spaces and effective learning algorithms.

To address pretraining data scarcity, we leverage the fact that dexterous robot hands are designed to mimic human hands and often share similar action patterns, while humans naturally generate abundant manipulation data in daily life. Egocentric human videos are cheaper, more diverse than robot teleoperation data and easier to scale. We therefore transform human videos into robot-executable trajectories to build a robot-centric dataset from human activity. However, there are substantial *kinematic* and *visual* gaps between human and robot hands. To close these gaps, we (i) introduce a human-in-the-loop retargeting procedure that combines fingertip-based inverse kinematics with interactive adjustment to align robot fingertip trajectories with human trajectories, ensuring physically plausible hand–object contacts; and (ii) mask the human hand in the visual stream and attach the retargeted robot hand into scene pointclouds to reduce visual mismatch.

Following this human-to-robot transformation pipeline, we construct **UniDex-Dataset** by building on open-source egocentric RGB-D manipulation videos [4, 28, 35, 36]. **UniDex-Dataset** is a unified foundation dataset comprising 9M paired image–pointcloud–action frames and over 50K trajectories across eight dexterous hand platforms, covering active DoFs from 6 to 24. To our knowledge, UniDex-Dataset is the first dataset to span such a broad spectrum of dexterous hand morphologies at this scale. We also provide protocols that allow researchers to contribute new hands or human datasets with minimal effort, continually scaling UniDex-Dataset and accelerating progress on dexterous manipulation.

To tackle heterogeneous embodiments and high-dimensional control, we further define a unified action space, the **Function–Actuator–Aligned Space (FAAS)**, which maps functionally similar actuators to shared coordinates. FAAS provides a function-centric control interface and enables skill transfer across different hands. Building on FAAS, we train **UniDex-VLA**, a 3D vision–language–action policy pretrained on UniDex-Dataset and finetuned with task demonstrations, serving as a foundation model that supports diverse dexterous hands.

In addition, we design a portable human-data capture setup, **UniDex-Cap**, which records synchronized RGB-D streams and human hand poses and converts them into robot-centric trajectories via the same transformation pipeline. UniDex-Cap enables efficient co-training on transformed human data together with smaller amounts of robot data, reducing teleoperation cost while preserving performance.

We evaluate UniDex-VLA on five challenging real-world tool-use tasks across two different hands. Across these tasks, **UniDex-VLA** achieves strong performance, outperforming other VLA baselines by a large margin (e.g., 81% average task progress vs. π_0 [7] at 38%), and demonstrates strong spatial, object, and cross-hand generalization; with FAAS and pretraining, it transfers skills to unseen hands in a zero-shot manner. Leveraging UniDex-Cap, we also provide a quantitative study showing how transformed human data can reduce post-training costs via human–robot co-training.

Our contributions are summarized as follows:

- **UniDex-Dataset:** a unified, diverse dexterous hand dataset (9M paired frames, over 50K trajectories, 8 hands, 6–24 DoFs) that supports large-scale pretraining toward universal dexterous hand foundation models.
- **FAAS & UniDex-VLA:** a function–actuator–aligned unified action space and a pretrained 3D vision–language–action model that achieves state-of-the-art performance on real-robot benchmarks, with strong spatial, object, and cross-hand generalization.

- **Human–Robot Data Co-training with UniDex-Cap:** a simple portable capture setup and pipeline that support human–robot data co-training; we quantitatively study how transformed human data can partially substitute real-robot demonstrations during post-training, showing that egocentric human videos both scale pretraining and reduce real-robot data needs.

2. Related Work

2.1. Dexterous Manipulation

Early research on dexterous manipulation was grounded in analytic and classical control formulations [2, 3, 26, 40, 43], and has since progressed toward learning-based methods that enable in-hand reorientation, rotation, and grasping [1, 10, 17, 19, 22, 24, 31, 45, 50, 54, 62, 65, 70, 75, 76]. Despite these advances, most approaches are tailored to specific tasks (grasping) or hardware and struggle to generalize to everyday tool-use. In contrast, we present UniDex-VLA, a foundation model aimed at general-purpose dexterous hand control.

2.2. Robot Foundation Policies and Unified Action Space

Diffusion-based policies and their variants constitute strong imitation-learning baselines [13, 51, 57, 58, 67]. With the rise of LLMs and VLMs, vision–language–action (VLA) models [6–8, 23, 27, 33, 39, 71–73] further scale imitation learning, but most existing approaches are pretrained on large-scale gripper-centric datasets. Recent efforts toward dexterous VLAs [22, 75] leverage simulation or limited real-world data, typically focusing on grasping and relying on hand-specific representations. In contrast, UniDex-VLA is pretrained on UniDex-Dataset to serve as a unified foundation policy for more general dexterous manipulation.

Designing a unified action space for robot foundation policies to handle embodiment heterogeneity is crucial for cross-embodiment generalization. RDT-1B [33] preserves the semantic structure of control signals, while π_0 [7] adopts a left-aligned action representation, and other methods introduce latent action spaces [8, 71]. However, these approaches primarily target gripper-centric actions. EgoVLA [60] attempts to leverage human parameters as a dexterous representation, but requires inverse kinematics in the post-training stage, which introduces additional errors, particularly for high-DoF dexterous hands. In contrast, FAAS provides a function-centric unified action representation that is post-processing-free, enabling more reliable cross-hand skill transfer.

2.3. Learning from Human Videos

Learning from human videos mitigates the data cost bottleneck but introduces visual and kinematic domain gaps. Prior work uses human hand trajectories for planning or control [9, 29, 34, 46, 52, 55, 63]; others apply retargeting with sim-to-real pipelines [11, 30, 66] or human-in-the-loop corrections [53], and some co-train with robot data [25, 48, 64, 78] to bridge the gap. However, many such pipelines primarily target grippers or do not scale robustly. There are also approaches that pretrained on egocentric human videos without explicit supervision of hand motion [41, 42, 61, 68]. More recent methods pretrain foundation models on egocentric videos to predict human hand motion, followed by specialized post-training to align with robot actions [38, 60], however these additional alignment stages can be complex and brittle. Our approach instead generates robot-centric dexterous hand supervision for pretraining, removing the need for specialized alignment tricks during fine-tuning while maintaining cross-hand control.

3. UniDex-Dataset

3.1. Overview

UniDex-Dataset is derived from four RGB-D egocentric human-manipulation datasets—H2O [28], HOI4D [35], HOT3D [4], and TACO [36]. We annotate language instructions if needed, segment videos into trajectory clips aligned with those instructions, and filter out invalid segments.

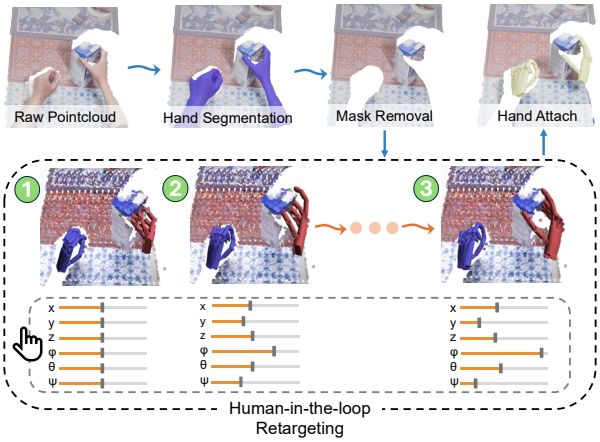


Figure 2. The figure illustrates the complete human–robot transformation pipeline. Starting from the raw scene pointcloud, we first mask out the human hands. We then perform human-in-the-loop retargeting through a user-friendly GUI in which the user only needs to **adjust slider bars** to modify the dummy base offset. ① shows the retargeted result without adjustment, whereas ③ shows the final configuration with improved, more plausible hand–object contact. Finally, after kinematic retargeting, we attach the retargeted robot dexterous hands to the scene.

The transformation from human data to robot-executable trajectories is illustrated in Fig. 2 and detailed in the next subsection. Applying this pipeline, we construct **UniDex-Dataset** comprising 9M paired image–pointcloud–action frames (recorded at 30 fps) and over 50k trajectories across

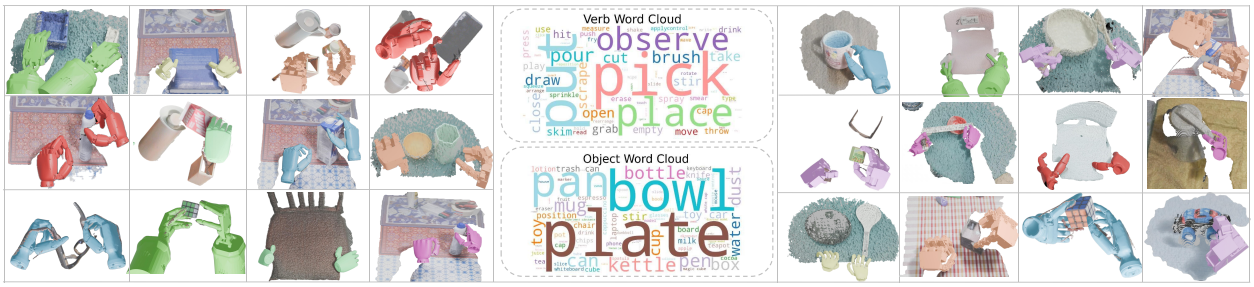


Figure 3. **UniDex-Dataset visualization.** We show a verb–object word cloud and a subset of UniDex-Dataset. **Colors** denote **different hands** (arbitrarily assigned; black corresponds to pretraining data). UniDex-Dataset spans diverse everyday tasks across a wide range of dexterous hand embodiments, including using a mobile phone, opening a milk carton, stir-frying with a spatula, lifting a chair, solving a Rubik's cube, and more.

eight dexterous hand platforms (Inspire, Leap, Shadow, Allegro, Ability, Oymotion, Xhand, and Wuji), covering active DoF from 6 to 24. Figure 3 visualizes the verb–object word cloud for the dataset and a subset of the data, spanning diverse daily manipulation tasks such as using a mobile phone, opening a milk carton, and stir-frying with a spatula. Table 1 compares UniDex-Dataset with released collected dexterous manipulation datasets [20, 37, 56] along the axes of trajectory count, hand variety and scene diversity, and supported perception modalities, highlighting the advantages of UniDex-Dataset. Owing to its diversity and robot-centric formulation—i.e., with minimal embodiment gap to the post-training stage—UniDex-Dataset serves as a strong foundation for pretraining dexterous manipulation models.

3.2. Human-Robot Transformation

Transforming human data into robot trajectories requires overcoming two core gaps: *kinematic* and *visual*. We outline our methods below.

3.2.1. Kinematic Retargeting

Fingertips are the primary contact points in human–object interaction. Our goal is to align human fingertip trajectories with those of the robot hand in 3D, while allowing a global hand-base adjustment to better ensure physically plausible contact.

Given a human hand pose, we extract m fingertip targets

$$X^\star = \left[x_1^\star, \dots, x_m^\star\right] \in \mathbb{R}^{3\times m}, \tag{1}$$

where m equals the number of robot fingers. The global human hand transform in the world frame is T_{hand}.

To precisely apply fingertip-based IK while permitting a base adjustment, we introduce a *6-DoF alignment offset*, implemented as a **dummy base** inserted before the real robot base. Let T_{offset} be the rigid transform from the dummy base to the real base, and let $T_{\text{world}}^{\text{dummy}}$ be the dummy-base pose in the world frame. The forward kinematics of fingertip i is

$$x_i(q; T_{\text{offset}}) \;=\; \mathrm{Trans}\Big(T_{\text{world}}^{\text{dummy}}\, T_{\text{offset}}\, T_i(q)\Big) \in \mathbb{R}^3, \tag{2}$$

where $T_i(q)$ is the homogeneous transform from the robot base to fingertip i, and $\mathrm{Trans}(\cdot)$ extracts the translation. We set $T_{\text{world}}^{\text{dummy}} = T_{\text{hand}}$ and keep it fixed during optimization. Stacking fingertip residuals yields the IK error:

$$e(q, T_{\text{offset}}) = \begin{bmatrix} x_1(q; T_{\text{offset}}) - x_1^\star \\ \vdots \\ x_m(q; T_{\text{offset}}) - x_m^\star \end{bmatrix} \in \mathbb{R}^{3m}. \tag{3}$$

For robot hands containing *mimic joint structures* (e.g., Inspire, Oymotion, Agility), we handle dependent joints through an iterative correction process. After solving the primary IK problem, each mimic joint j_s is updated from its master joint j_m as

$$q_{j_s} = k\, q_{j_m} + c \tag{4}$$

consistent with the kinematic model specification, where k and c denote the mimic constraints. This correction is repeated for N iterations, re-evaluating fingertip error each time until convergence.

For implementation, we provide a user-friendly and rapid process. The whole pipeline is a two-stage, human-in-the-loop retargeting procedure.

1. **Automatic stage.** Given an initial T_{offset}, we solve Eq. 3 via PyBullet [15]'s multi-end-effector IK solver to obtain a joint configuration q that minimizes fingertip error while satisfying joint limits and damping.
2. **Interactive stage.** A lightweight GUI exposes the six degrees of freedom of T_{offset} (three translations and three rotations, as shown in Fig. 2) and other configuration for IK solver. The user visually inspects alignment and manually adjusts T_{offset}; after each adjustment, we re-solve the IK problem. This process typically converges within a few manual tweaks, producing robust fingertip alignment across diverse poses. ① and ③ in Fig. 2 shows the comparison between and after the interactive stage.

Dataset	# of Trajectories	# of Hands	Language Annotations	Varied Scenes	RGB	Depth	Pointcloud
UniDex-Dataset	52K	8	✓	✓	✓	✓	✓
ActionNet [20]	30K	2	✓	✗	✓	✓	✗
RoboMind [56]	19K	1	✓	✗	✓	✓	✗
RealDex [37]	2K	2	✓	✗	✓	✓	✓

Table 1. **Comparison between UniDex-Dataset and other dexterous manipulation datasets.** UniDex-Dataset advances in total trajectories, variety across hands/actions/scenes, and supports for all perception modalities. ✗denotes the pointcloud in ActionNet [20] is very low-quality.

For each human dataset and each dexterous hand, we perform a basic interactive calibration to select dummy base offsets to handle systematic differences across datasets (e.g., coordinate frames/ hand-pose estimation bias) and hand morphology differences. We then adjust a small subset of frames, focusing on contact-rich segments to improve contact plausibility. In practice, we find the basic calibration suffices to cover the vast majority of trajectories, enabling our transformation pipeline to scale to large egocentric datasets with modest human effort.

3.2.2. Visual Alignment

We compute pointclouds from RGB-D frames. Then to reduce the visual gap, we mask human hands (using WiLoR [44] together with SAM2 [49]) and remove the corresponding points. We then place the retargeted robot-hand mesh into the scene and render its geometry into the pointcloud. Finally, we reproject the fused pointcloud back to the RGB-D frame via a pinhole camera model [21] to avoid occlusions caused by incorrect depth ordering, matching the single-view setting used during real-world fine-tuning.

4. UniDex-VLA

4.1. Unified Action Space: FAAS

We pretrain our robot foundation model on UniDex-Dataset, which spans diverse dexterous-hand embodiments. A unified action space that enables transfer across hands is therefore critical. To this end, we introduce a simple yet effective action representation, the **Function–Actuator–Aligned Space** (**FAAS**). For any dexterous hand with n actuated DoFs in its kinematic model, each *actuator* is mapped to the FAAS *index* corresponding to its functional role. Here we use "actuator" broadly to denote any controllable DoF/channel derived from the robot URDF, including mimic joints when present.

Conceptually, FAAS exposes a function-centric control interface shared across embodiments rather than a URDF-specific joint space. Although dexterous hands differ in link lengths, couplings, and layouts, they all implement a small set of functional primitives—such as thumb–index pinch, finger curling around handles, or lateral ab-/adduction for stabilization. FAAS groups actuators by these functional roles and maps them into a common coordinate system, discarding embodiment-specific nuisance factors while preserving task-relevant control semantics. Fig. 4 illustrates, for the thumb and ring fingers of different hands, how individual joints are mapped to FAAS indices.

FAAS is an 82-dimensional action vector. The first 18 dimensions encode wrist poses (9 per hand), where each 9d pose consists of a 6d continuous rotation representation (two 3d vectors for the local x- and y-axes) followed by a 3d translation. and the remaining 64 dimensions encode joint commands, with 32 slots for each hand. Among these slots, we reserve 21 *base* actuator slots that are shared across all hands, and use the remaining slots for hand-specific DoFs (e.g., additional wrist joints on the Shadow Hand) and for future hands. The details of joint mapping for different hands are shown in Sec. C and Fig. 15 in Appendix.

Figure 4. **Function–Actuator–Aligned Space (FAAS).** We show the thumb and ring fingers of Oymotion (11 actuators), Allegro (16), Inspire (12), and Wuji (20), with colors denoting individual joints, curves indicating rotation directions, and dotted lines indicating rotation axes. Indices {0,1,3,5,6} are aligned across all four hands because the corresponding joints share similar functional roles.

4.2. VLA Policy

UniDex-VLA aims to be a 3D, language-conditioned foundation model for dexterous control. Unlike prior VLAs that pair 2D encoders with low-dimensional gripper actions, our setting is inherently volumetric and high-DoF: tool-use requires reasoning about fine 3D geometry and contact affordances, especially in the egocentric single-view observation. By coupling 3D visual inputs with the unified FAAS action space, UniDex-VLA aligns geometric perception and

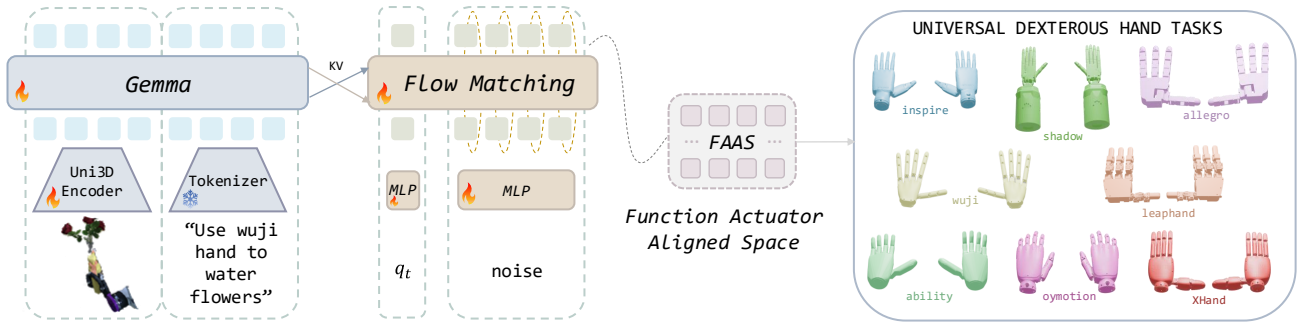


Figure 5. **Overview of UniDex-VLA.** At time t, the model consumes a single-view colored pointcloud P_t, a language instruction ℓ_t, and proprioception q_t, and predicts an H-step action chunk $A_t = [a_t, \ldots, a_{t+H-1}]$ expressed in the unified action space FAAS. Uni3D [77] encodes the colored pointcloud; features are fused with text and proprioception in the backbone and decoded into FAAS actions. The policy is pretrained on UniDex-Dataset and optimized with a conditional flow-matching objective.

control in a shared representation, supporting spatial, object, and cross-hand generalization.

4.2.1. Observations and Action Outputs

As shown in Fig. 5, the observation at time t is $o_t = [P_t, \ell_t, q_t]$, where P_t is a single-view colored pointcloud derived from an RGB-D image and then cropped and downsampled, ℓ_t is a natural-language instruction, and q_t is a vector of robot proprioceptive states. We model $p(A_t \mid o_t)$, where $A_t = [a_t, \ldots, a_{t+H-1}]$ denotes an H-step action chunk [74]. Both q_t and each a_t are represented in FAAS. For the wrist in q_t, we use an *absolute* pose; for action outputs, we adopt a *relative* wrist pose with respect to the first frame of the action chunk, following UMI [14]. For dexterous-hand joints, we likewise use abstracted representations in both q_t and a_t.

4.2.2. Model Architecture

The UniDex-VLA architecture largely follows π_0 [7], with modifications for pointcloud inputs. Specifically, we replace the SigLIP [69] 2D vision encoder in PaliGemma [5] with Uni3D [77], a strong 3D pointcloud encoder. Uni3D adopts a vanilla ViT [18] design and is initialized from a 2D pretrained ViT, aligning pointcloud features with image–text–aligned features. We train the policy with a conditional flow-matching objective and generate denoised action chunks at inference time via forward–Euler integration [32]. More details of UniDex-VLA training are shown in Sec. A in Appendix.

5. Experiments

5.1. Experimental Setup

Hardware Platform. Our real-world experiments use a 7-DoF Franka robotic arm equipped with three dexterous end-effectors: an Inspire Hand (6 active, 12 full DoFs), a Wuji Hand (20 active DoFs), and an Oymotion Hand (6 active, 11 full DoFs), all mounted at the end-effector. An Intel RealSense L515 provides egocentric RGB-D observations for all experiments. The complete workstation is shown in Fig. 6.

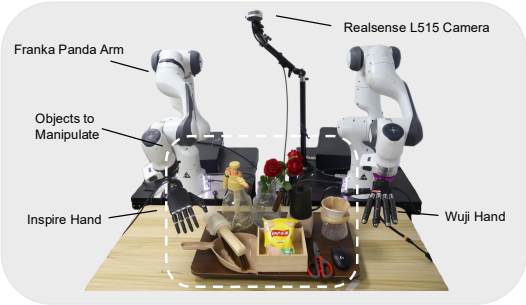


Figure 6. Real-world experiments setup overview

Task Description. Everyday manipulation commonly involves many tools designed for human hands—e.g., scissors, spray bottles, and sweepers—which impose stringent requirements on finger coordination and in-hand reconfiguration. To better assess the dexterity and generality of our approach, we evaluate five challenging tool-use tasks, with visualization of different stages in Fig. 7: (i) **Make Coffee** (Inspire Hand): Grasp the kettle and lift it to the dripper to pour water to make pour-over coffee. Task decomposed into kettle grasping (**Grasp**) and water pouring (**Pour**). (ii) **Sweep Objects** (Inspire Hand): Grasp a sweeper and sweep tabletop objects into a dustpan. Task decomposed into sweeper grasping (**Grasp**) and sweeping (**Sweep**). (iii) **Water Flowers** (Wuji Hand): Grasp a spray bottle, lift it, and press the trigger with the thumb to water flowers. Task decomposed into bottle grasping (**Grasp**) and pressing trigger to water (**Press**). (iv) **Cut Bags** (Wuji Hand): Insert thumb, middle and ring fingers into scissors and grasp them in a human-like manner to cut bags. Task decomposed into scissors grasping (**Grasp**) and cutting (**Cut**). (v) **Use Mouse** (Wuji Hand): Place fingers on a computer mouse and use it to drag a file into a USB folder in the desktop interface and click the mouse to finish. We report the mean success rate across all task stages as the **average task progress**, which

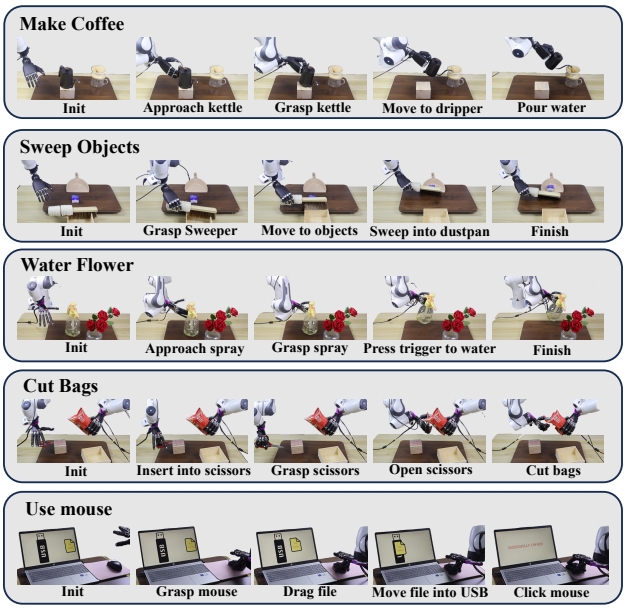


Figure 7. Our real-robot benchmark comprises 5 challenging tool-use tasks. We visualize the key stages of each task, illustrating the precise dexterous control required to successfully complete them.

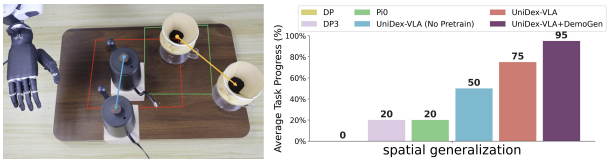


Figure 8. **Spatial generalization.** Left: the kettle and dripper are placed at *out-of-distribution (OOD)* positions relative to training demonstrations. Red and green lines circling regions denote the training placement ranges for the kettle and dripper, respectively. Right: average task progress for different methods (10 trials each).

serves as our primary metric for comparing methods.

Demonstration Collection. We build our teleoperation system on OpenTeleVision [12] and dex-retargeting [47] with Apple Vision Pro. We only collect *50* demonstrations per task for fine-tuning.

Baselines. We compare **UniDex-VLA** with representative imitation learning and VLA methods: Diffusion Policy (DP) [13], 3D Diffusion Policy (DP3) [67], and the strong VLA baseline π_0 [7] pretrained on gripper action datasets. To directly assess the effect of pretraining, we include UniDex-VLA (No Pretrain). We adopt FAAS for UniDex-VLA (No Pretrain) and π_0, and retain low-dimensional outputs for DP and DP3.

5.2. Performance

We report results on five real-world manipulation tasks across two dexterous hands at Fig. 11. The results show that, with only 50 demonstrations per task, **UniDex-VLA** attains high success rates on these challenging, long-horizon tool-use tasks and surpasses all baselines by a large margin, including on the especially difficult *Use Scissors to Cut Bags* task. The performance gap between UniDex-VLA (No-Pretrain) and UniDex-VLA further provides a clear ablation of the benefit of pretraining on UniDex-Dataset. Computing relative improvement over the best competing method (Fig. 11), UniDex-VLA achieves the largest gain on the hardest setting, *Use Scissors to Cut Bags*, with an **84.6%** increase in average task progress. Overall, these results indicate that pretraining endows UniDex-VLA with strong motion priors for dexterous hand control, particularly on highly dexterous tool-use tasks, enabling more efficient adaptation to new and challenging behaviors.

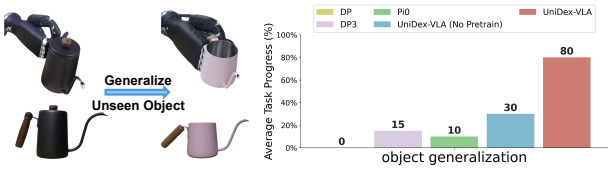


Figure 9. **Object generalization.** Left: we replace the original black kettle with a smaller purple kettle that differs in color, size, and functional parts (handle & spout). Right: average task progress for different methods (10 trials each).

Hand Type	π_0	UniDex-VLA (No Pretrain)	UniDex-VLA
Wuji	0%	0%	**40%**
Oymotion	10%	5%	**60%**

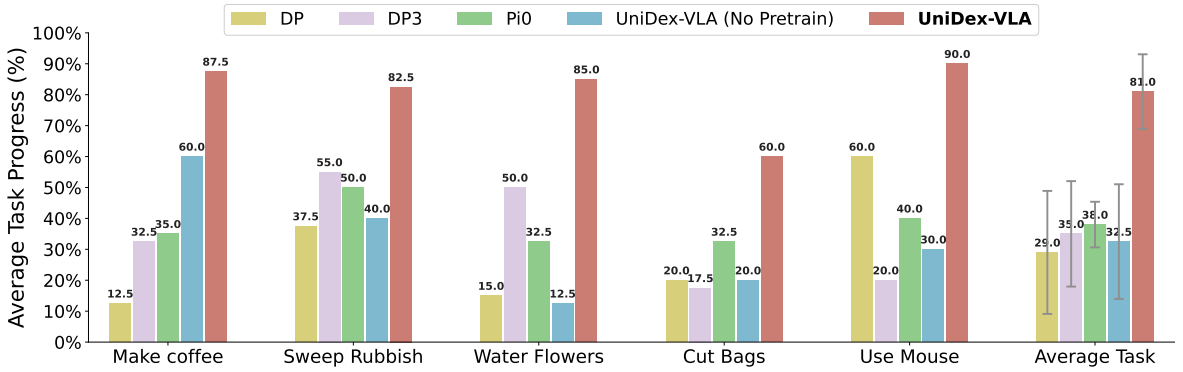
Figure 10. **Hand generalization (zero-shot skill transfer).** We transfer a policy trained on the Inspire Hand to Wuji and Oymotion. Table reports **average task progress (%)** under zero-shot deployment (10 trials each).

5.3. Generalization

Beyond outperforming performance, UniDex-VLA demonstrates strong spatial, object, and hand generalization.

Spatial Generalization. UniDex-VLA benefits from 3D perception, and pointclouds further enable simple, automatic data augmentation via geometric editing. In the *Make Coffee* experiment, we segment the pointclouds of the kettle and the dripper, and translate them along the table's x/y axes to sweep across the workspace and generate out-of-distribution (o.o.d.) placements. After editing the pointclouds, the corresponding robot states are aligned to the new scenes using Task and Motion Planning (TAMP) [16]. DemoGen [59] provides an automated pipeline for this procedure. As shown in Fig. 8, UniDex-VLA generalizes well across spatial configurations; with DemoGen [59] augmentation, it approaches very high success rate over full workspace.

Object Generalization. As in Fig. 9, we replace the black



Model	DP	DP3	π_0	UniDex-VLA (No Pretrain)	UniDex-VLA
Average Task Progress (avg)	$29.0 \pm 19.9\%$	$35.0 \pm 17.1\%$	$38.0 \pm 7.4\%$	$32.5 \pm 18.5\%$	$\mathbf{81.0 \pm 12.1}\%$
Final Success Rate (avg)	$22.0 \pm 22.5\%$	$30.0 \pm 18.7\%$	$35.0 \pm 10.0\%$	$23.0 \pm 12.0\%$	$\mathbf{76.0 \pm 17.8}\%$

Figure 11. Average task progress across five real-world tasks (top), with aggregate averages of average task progress and final success rate (bottom) over 5 tasks. Each task/algorithm uses **20** trials.

kettle with a smaller purple kettle that differs in color, size, and functional parts (handle & spout). **UniDex-VLA** maintains strong performance on this unseen object, indicating generalizable tool understanding capacity crucial for robust and general tool-use.

Hand Generalization (Skill Transfer). We evaluate cross-hand transfer by taking a policy trained to *Make Coffee* on the Inspire Hand (6 active DoF) and deploying it *zero-shot* on Wuji (20 active DoFs) and Oymotion (6 active DoFs with different kinematics). As shown in Fig. 10, **UniDex-VLA** achieves **60%** success on Oymotion and **40%** on Wuji without any fine-tuning, whereas baselines are near zero. These results highlight that pretraining across diverse dexterous hands—together with FAAS—indeed enables zero-shot cross-hand skill transfer.

5.4. UniDex-Cap for Human-Robot Data Co-train

We introduce **UniDex-Cap**, a practical data-capture setup that records synchronized RGB-D streams and hand/head poses. The system combines an Apple Vision Pro for hand and head pose estimation, an Intel RealSense L515 for high-quality RGB-D, and a custom 3D-printed mount to physically couples the two sensors with a fixed rigid transform. This transform is calibrated to ensure the RGB-D stream and the hand/head poses are time-synchronized and expressed in the shared coordinate frame. As illustrated in Fig. 12, we then apply the human-to-robot transformation pipeline (Sec. 3.2) to convert captured human data into robot-executable trajectories. In addition, we perform a viewpoint transformation to align human and robot perspectives and downsample the human motion to match typical teleoperation speeds.

Leveraging UniDex-Cap, we collect human demonstrations, transform them, and *co-train* with real-robot data

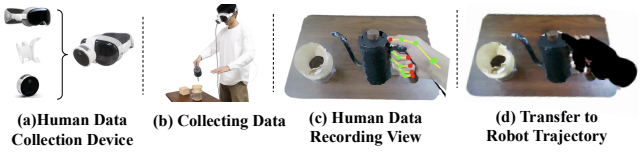


Figure 12. (a,b) show the components of UniDex-Cap. (c,d) shows the example captured data and converted robot-executable trajectories.

on *Make Coffee* task to quantitatively explore the effect of human demos during the finetuning stage. Figure 13 reports average task progress versus the numbers of co-trained transformed human demos (h) and robot demos. We observe: (i) **Retargeted human data helps, but robot data is indispensable.** Although for a fixed r, increasing h consistently improves average task progress within our evaluated range but success always remains near zero without any robot data. (ii) **Human–robot exchange rate $\approx$ 2:1.** From Fig. 13, the boundary separating the "high-performance" region (comparable to the r=50 robot-only result green area) has slope $\approx$ 2, suggesting roughly *two human demos can substitute for one robot demo*. (iii) **Cost efficiency.** On *Make Coffee* task, human demos are $\sim$5.2$\times$ faster to collect than real robot demos; considering the $\approx$2:1 exchange rate, co-training with human demos can substantially reduce data collection cost.

6. Conclusion and Limitation

We presented UniDex, a robot foundation suite built from egocentric human videos, comprising UniDex-Dataset, UniDex-VLA, and UniDex-Cap. We believe UniDex can serve as a practical foundation platform for the community, accelerating progress toward general, scalable, and transferable dexterous manipulation. A limitation of our cur-

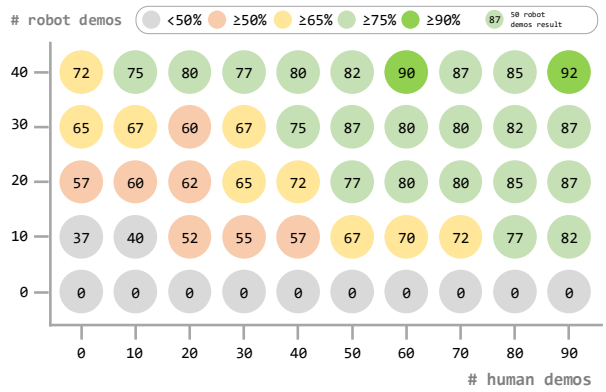


Figure 13. **Human-Robot co-training.** Average task progress versus the numbers of transformed human demos (h) and robot demos (r). Colors indicate different performance bands (green: comparable to the r=50 robot-only result). Each point averages over **20** trials.

rent work is that we do not yet leverage large *action-free* (or weakly labeled) egocentric activity datasets; extending UniDex to incorporate such data is a promising direction for further scaling dexterous pretraining.

7. Acknowledgment

We would like to give special thanks to Wuji Technology Inc. for providing the hardware support, and Hojin Bae, Haoxu Huang, Shaoting Zhu for their technical support. Tsinghua University Dushi Program supports this project.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 3

[2] Suguru Arimoto. Intelligent control of multi-fingered hands. *Annual Reviews in Control*, 28(1):75–85, 2004. 3

[3] Yunfei Bai and C Karen Liu. Dexterous manipulation using both palm and fingers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1560–1565. IEEE, 2014. 3

[4] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. *CVPR*, 2025. 2, 3

[5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 6

[6] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 2, 3

[7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2, 3, 6, 7, 1

[8] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025. 3

[9] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27661–27672, 2025. 3

[10] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023. 3

[11] Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024. 3

[12] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024. 7

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 2, 3, 7, 1

[14] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024. 6

[15] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016. 4

[16] Murtaza Dalal, Ajay Mandlekar, Caelan Garrett, Ankur Handa, Ruslan Salakhutdinov, and Dieter Fox. Imitating task and motion planning with visuomotor transformers. *arXiv preprint arXiv:2305.16309*, 2023. 7

[17] Kairui Ding, Boyuan Chen, Ruihai Wu, Yuyang Li, Zongzheng Zhang, Huan-ang Gao, Siqi Li, Guyue Zhou, Yixin Zhu, Hao Dong, et al. Preafford: Universal affordance-based pre-grasping for diverse objects and environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7278–7285. IEEE, 2024. 3

[18] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[19] Hao-Shu Fang, Hengxu Yan, Zhenyu Tang, Hongjie Fang, Chenxi Wang, and Cewu Lu. Anydexgrasp: General dexterous grasping for different hands with human-level learning efficiency. *arXiv preprint arXiv:2502.16420*, 2025. 3

[20] Yao Mu Fourier ActionNet Team. Actionnet: A dataset for dexterous bimanual manipulation. 2025. 4, 5

[21] Richard Hartley. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[22] Jiawei He, Danshi Li, Xinqiang Yu, Zekun Qi, Wenyao Zhang, Jiayi Chen, Zhaoxiang Zhang, Zhizheng Zhang, Li Yi, and He Wang. Dexvlg: Dexterous vision-language-grasp model at scale. *arXiv preprint arXiv:2507.02747*, 2025. 3

[23] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1724–1734, 2025. 3

[24] Juntao Jian, Xiuping Liu, Zixuan Chen, Manyi Li, Jian Liu, and Ruizhen Hu. G-dexgrasp: Generalizable dexterous grasping synthesis via part-aware prior retrieval and prior-assisted generation. *arXiv preprint arXiv:2503.19457*, 2025. 3

[25] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025. 3

[26] Jeffrey Kerr and Bernard Roth. Analysis of multifingered hands. *The International Journal of Robotics Research*, 4 (4):3–17, 1986. 3

[27] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 2, 3

[28] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10138–10148, 2021. 2, 3

[29] Gen Li, Nikolaos Tsagkas, Jifei Song, Ruaridh Mon-Williams, Sethu Vijayakumar, Kun Shao, and Laura Sevilla-Lara. Learning precise affordances from egocentric videos for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10581–10591, 2025. 3

[30] Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6991–7003, 2025. 3

[31] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5637–5643. IEEE, 2025. 3

[32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 6

[33] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. 3

[34] Vincent Liu, Ademi Adeniji, Haotian Zhan, Siddhant Haldar, Raunaq Bhirangi, Pieter Abbeel, and Lerrel Pinto. Egozero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025. 3

[35] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 2, 3

[36] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21740–21751, 2024. 2, 3

[37] Yumeng Liu, Yaxun Yang, Youzhuo Wang, Xiaofei Wu, Jiamin Wang, Yichen Yao, Sören Schwertfeger, Sibei Yang, Wenping Wang, Jingyi Yu, et al. Realdex: Towards human-like grasping for robotic dexterous hand. *arXiv preprint arXiv:2402.13853*, 2024. 4, 5

[38] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: Vision-language-action pre-training from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025. 3

[39] Cui Miao, Tao Chang, Meihan Wu, Hongbin Xu, Chun Li, Ming Li, and Xiaodong Wang. Fedvla: Federated vision-language-action learning with dual gating mixture-of-experts for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6904–6913, 2025. 3

[40] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012. 3

[41] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 3

[42] Dantong Niu, Yuvan Sharma, Haoru Xue, Giscard Biamby, Junyi Zhang, Ziteng Ji, Trevor Darrell, and Roei Herzig. Pre-training auto-regressive robotic models with 4d representations. *arXiv preprint arXiv:2502.13142*, 2025. 3

[43] Jean Ponce, Steve Sullivan, Attawith Sudsang, Jean-Daniel Boissonnat, and Jean-Pierre Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16(1): 11–35, 1997. 3

[44] Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand

localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12242–12254, 2025. 5

[45] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023. 3

[46] Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022. 3

[47] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems*, 2023. 7

[48] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025. 3

[49] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

[50] Zilin Si, Gu Zhang, Qingwei Ben, Branden Romero, Zhou Xian, Chao Liu, and Chuang Gan. Difftactile: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. *arXiv preprint arXiv:2403.08716*, 2024. 3

[51] Jingyi Tian, Le Wang, Sanping Zhou, Sen Wang, Jiayi Li, Haowen Sun, and Wei Tang. Pdfactor: Learning tri-perspective view policy diffusion field for multi-task robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15757–15767, 2025. 3

[52] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023. 3

[53] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024. 3

[54] Youzhuo Wang, Jiayi Ye, Chuyang Xiao, Yiming Zhong, Heng Tao, Hang Yu, Yumeng Liu, Jingyi Yu, and Yuexin Ma. Dexh2r: A benchmark for dynamic dexterous grasping in human-to-robot handover. *arXiv preprint arXiv:2506.23152*, 2025. 3

[55] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 3

[56] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. 4, 5

[57] Huilin Xu, Jian Ding, Jiakun Xu, Ruixiang Wang, Jun Chen, Jinjie Mai, Yanwei Fu, Bernard Ghanem, Feng Xu, and Mohamed Elhoseiny. Diffusion-based imaginative coordination for bimanual manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11469–11479, 2025. 3

[58] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025. 2, 3

[59] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025. 7, 1

[60] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025. 3

[61] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 3

[62] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023. 3

[63] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 3

[64] Chengbo Yuan, Rui Zhou, Mengzhen Liu, Yingdong Hu, Shengjie Wang, Li Yi, Chuan Wen, Shanghang Zhang, and Yang Gao. Motiontrans: Human vr data enable motion-level learning for robotic manipulation policies. *arXiv preprint arXiv:2509.17759*, 2025. 3

[65] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024. 3

[66] Zhecheng Yuan, Tianming Wei, Langzhe Gu, Pu Hua, Tianhai Liang, Yuanpei Chen, and Huazhe Xu. Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation. *arXiv preprint arXiv:2508.20085*, 2025. 3

[67] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024. 2, 3, 7, 1

[68] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024. 3

[69] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training.

In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 6

[70] Gu Zhang, Hao-Shu Fang, Hongjie Fang, and Cewu Lu. Flexible handover with real-time robust dynamic grasp trajectory generation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3192–3199. IEEE, 2023. 3

[71] Yang Zhang, Chenwei Wang, Ouyang Lu, Yuan Zhao, Yunfei Ge, Zhenglong Sun, Xiu Li, Chi Zhang, Chenjia Bai, and Xuelong Li. Align-then-steer: Adapting the vision-language action models through unified latent guidance. *arXiv preprint arXiv:2509.02055*, 2025. 3

[72] Zongzheng Zhang, Haobo Xu, Zhuo Yang, Chenghao Yue, Zehao Lin, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Elucidating the design space of torque-aware vision-language-action models. In *9th Annual Conference on Robot Learning*, 2025.

[73] Zongzheng Zhang, Chenghao Yue, Haobo Xu, Minwen Liao, Xianglin Qi, Huan-ang Gao, Ziwei Wang, and Hao Zhao. Robochemist: Long-horizon and safety-compliant robotic chemical experimentation. *arXiv preprint arXiv:2509.08820*, 2025. 3

[74] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 2, 6

[75] Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Zhang Chen, Tianrui Guan, Fanlian Zeng, Ka Num Lui, Yuyao Ye, Yitao Liang, et al. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. *arXiv preprint arXiv:2502.20900*, 2025. 3

[76] Yiming Zhong, Qi Jiang, Jingyi Yu, and Yuexin Ma. Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22584–22594, 2025. 3

[77] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023. 6

[78] Jiaming Zhou, Teli Ma, Kun-Yu Lin, Zifan Wang, Ronghe Qiu, and Junwei Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22551–22561, 2025. 3

Appendix

A. Training Details

A.1. UniDex-VLA Flow-Matching Loss

To train UniDex-VLA, we minimize a conditional flow-matching loss:

$$L^\tau(\theta) = \mathbb{E}_{p(A_t|o_t),\, q(A_t^\tau|A_t)} \left[\left\| v_\theta(A_t^\tau, o_t) - u(A_t^\tau \mid A_t) \right\| \right], \tag{5}$$

where $\tau \in [0,1]$ and $q(A_t^\tau \mid A_t) = \mathcal{N}\big(\tau A_t,\ (1-\tau)I\big)$ is a linear-Gaussian probability path. We sample $A_t^\tau = \tau A_t + (1-\tau)\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ and compute the target conditional vector field $u(A_t^\tau \mid A_t) = A_t - \epsilon$. The network is trained such that the predicted vector field $v_\theta(A_t^\tau, o_t)$ approximates $u(A_t^\tau \mid A_t)$.

At inference time, we integrate the learned vector field using a forward Euler scheme to generate a denoised action chunk:

$$A_t^{\tau+\delta} \;=\; A_t^\tau + \delta\, v_\theta(A_t^\tau,\, o_t)\,,$$

with step size $\delta = 0.1$ and initial condition $A_t^0 \sim \mathcal{N}(0, I)$.

A.2. UniDex-VLA Pretraining

During pre-training, we use 8 NVIDIA H800 GPUs with a total batch size of 128. The model used for subsequent post-training is trained for 3 epochs ($\sim$ 30k steps), which takes around 24 hours. We adopt the AdamW optimizer and a cosine learning-rate scheduler with an initial learning rate of 1e-4. The learning rate is decayed by a factor of 0.95 at the 2nd epoch. The weight decay is set to 1e-10, and we apply gradient clipping with a maximum norm of 1.0.

A.3. UniDex-VLA Post-training

During post-training, we use 2 NVIDIA H800 GPUs for each task, with a total batch size of 8. We use the AdamW optimizer without a learning rate scheduler and set the initial learning rate to 2.5e-5. For common data, we train the model for 50 epochs ($\sim$ 3k steps), which takes around 4 hours. For DemoGen [59] augmented data, we train the model for 2 epochs ($\sim$ 1.8k steps), which takes around 2.5 hours. The weight decay is set to 1e-10, and we again use gradient clipping with a maximum norm of 1.0.

A.4. Baselines

For all baselines (DP [13], DP3 [67], and π_0 [7]), we post-train the models until convergence on the validation set.

For DP [13] and DP3 [67], we use the AdamW optimizer with an initial learning rate of 1e-4. The state horizon is set to 4 and the action horizon to 32. We use a batch size of 32 and train for 400 epochs.

For π_0 [7], we use the AdamW optimizer with an initial learning rate of 2.5e-5. The batch size is set to 8 and the model is trained for 50 epochs. The number of diffusion steps is set to 10.

For our UniDex-VLA baseline without pretraining, we use the same training hyperparameters as UniDex-VLA with pretraining.

B. Human-in-the-loop Retargeting GUI

To minimum human efforts in our human-in-the-loop retargeting process, we develop a human-friendly web-based GUI, as shown in Fig. 14. Through this interface, users can adjust dummy base links, IK parameters, and other retargeting settings to obtain satisfactory robot trajectories.

C. FAAS Details

Here we show the details for the 32 dimensions encoding dexterous hand joints. Dimensions 0–4, 5–9, 10–14, 15–19, and 20–24 correspond to the thumb, index, middle, ring, and little fingers, respectively. Dimensions 25–26 are reserved for extra wrist joints of Shadow hands. Dimensions 27–31 are left unused for new hands. The detailed joint mappings of the robotic hands used in FAAS are shown in Fig. 15.

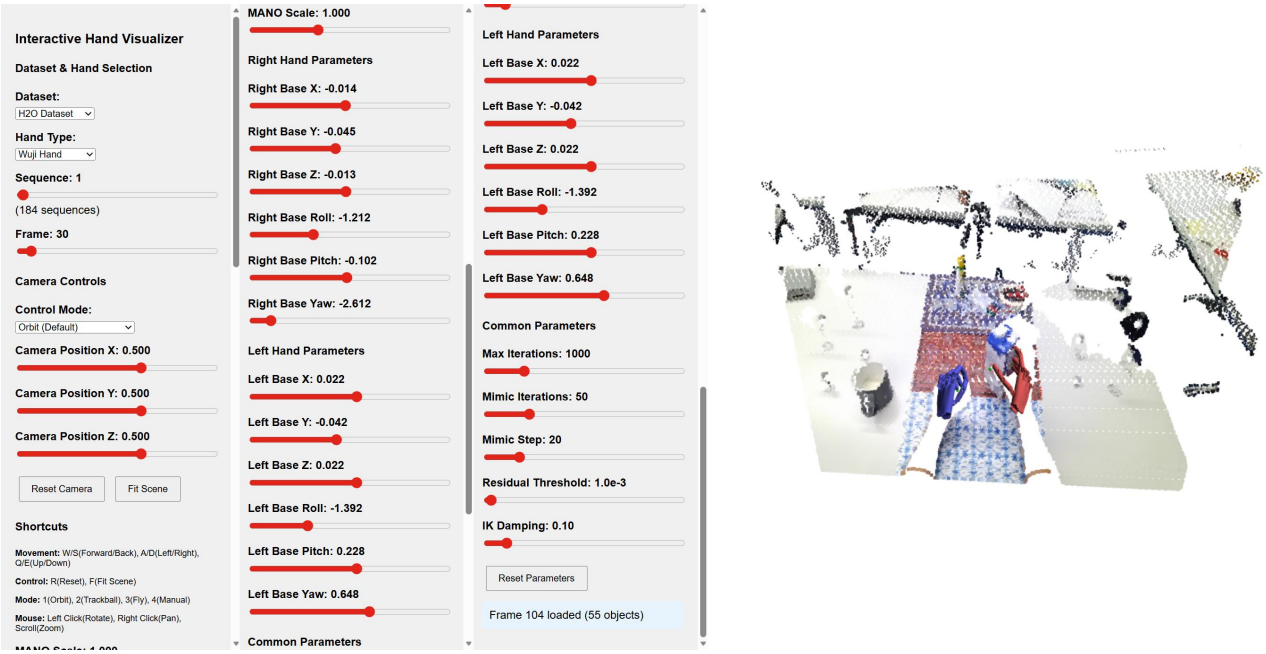


Figure 14. Human-friendly web-basedGUI for retargeting human demonstrations to robot executions. Users can adjust the IK parameters, dummy links, and other settings through the GUI to obtain satisfactory retargeted robot trajectories.

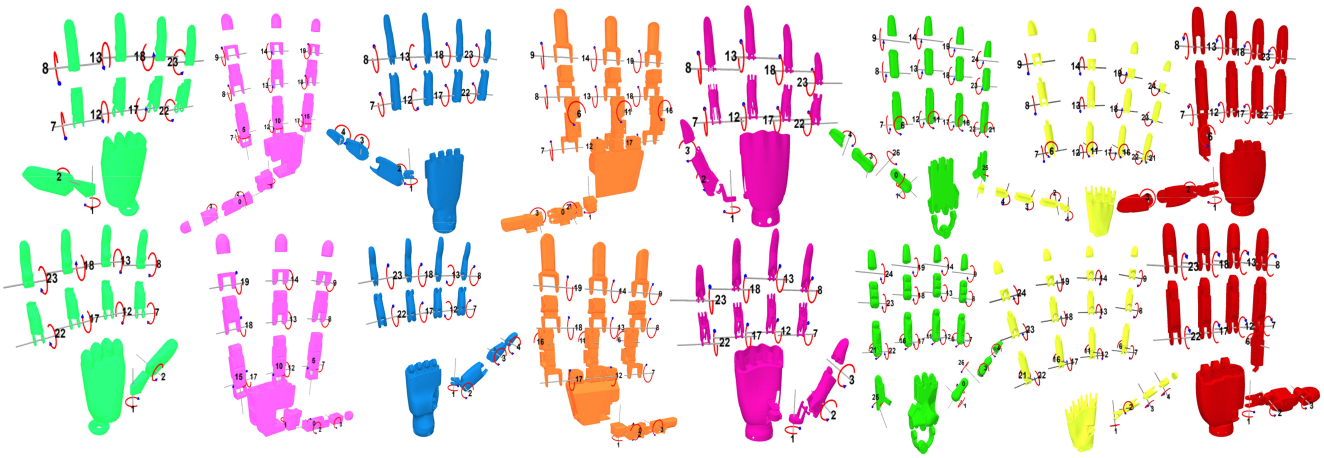


Figure 15. Joint mappings of different robotic hands used in FAAS. From left to right are Ability, Allegro, Inspire, Leap, Oymotion, Shadow, Wuji, and Xhand. The two rows show different views of the joint mappings on the right hand.

D. UniDex-Cap Setup Calibration

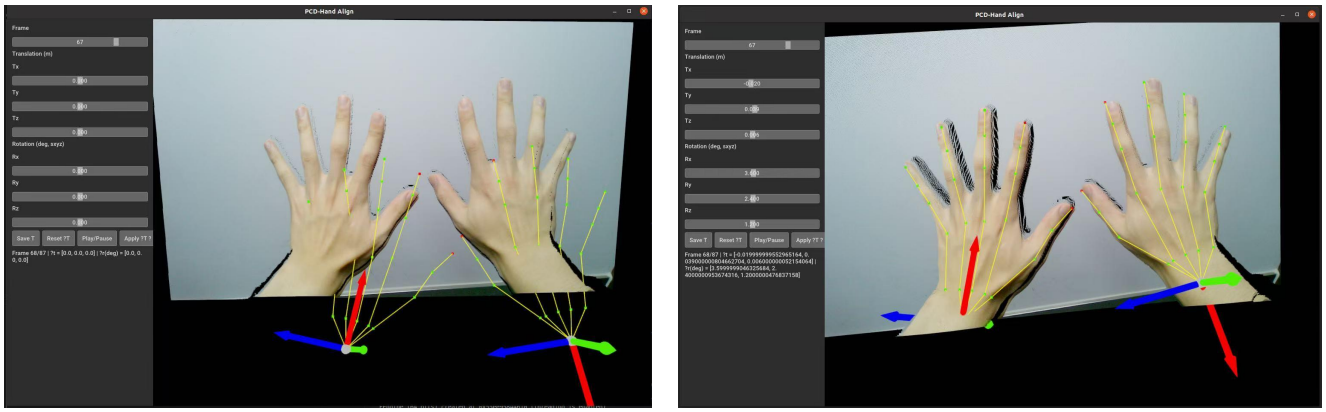
UniDex-Cap combines an Apple Vision Pro (for hand and head poses, denoted $\{P_{\mathrm{VP}}\}$) and an Intel RealSense L515 (for RGB-D). Because Vision Pro does not expose third-party RGB-D video recording, we physically couple the two sensors with a custom 3D-printed mount that rigidly fixes their relative pose. This mechanical constraint ensures that the extrinsic transform between the Vision Pro and the RealSense remains stable for a given user.

As shown in Fig. 16, we provide a lightweight GUI to estimate the remaining constant extrinsics with minimal manual effort. The user records a short calibration clip and then uses a slider-based interface to adjust the hand and wrist poses in the Vision Pro coordinate frame—visualized as a skeleton—until they align with the 3D hand point cloud captured by the RealSense camera. The slider values directly correspond to the transform $T_{\mathrm{RS}}^{\mathrm{VP}}$. Once this transform is determined, all Vision

Pro poses are converted into the RealSense camera frame, yielding temporally aligned hand and head trajectories:

$$P_{\mathrm{RS}} \;=\; T_{\mathrm{RS}}^{\mathrm{VP}}\, P_{\mathrm{VP}}, \tag{6}$$

where P_{VP} and P_{RS} are represented in homogeneous coordinates. This pipeline produces temporally synchronized, geometrically consistent annotations suitable for downstream retargeting and post-training.



(a) Before Calibration **(b) After Calibration**

Figure 16. GUI for UniDex-Cap calibration. (a) shows the initial state before calibration; (b) shows the calibrated result where the hand poses captured by Vision Pro align with the 3D point cloud captured by the RealSense L515 camera.

E. Core Contribution List

The main contributions of the core contributors are as follows:
Gu Zhang: Project lead. Developed the overall dataset construction pipeline, model architecture, and unified action space; built the robot system infrastructure; and wrote the paper.
Qicheng Xu: Led VLA model training; optimized the dataset construction and policy inference pipelines; and contributed to paper writing.
Haozhe Zhang: Led dataset processing; improved the robot system and the human–robot data capture pipeline; and contributed to paper writing.
Jianhan Ma: Implemented retargeting algorithms; and developed dataset visualizations and contributed to early-stage exploration.
Long He: Implemented DemoGen algorithm; and contributed to paper writing.
Yiming Bao: Collected robot data and human data; and contributed to DemoGen algorithm implementation.