北京大學
PEKING UNIVERSITY

Cross-Embodiment Dexterous Grasping with Reinforcement Learning

Haoqi Yuan, Bohan Zhou, Yuhui Fu, Zongqing Lu

ICLR 2025

Project page: https://sites.google.com/view/crossdex

Code: https://github.com/PKU-RL/CrossDex/

About me: https://yhqpkueecs.github.io/

Background

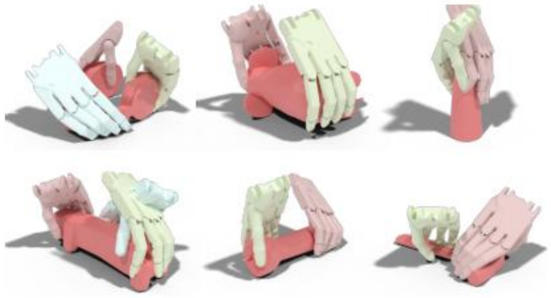
- Dexterous grasping
  - Grasping is the **fundamental skill** for robotic manipulation
    - Object rearrangement, tool use, in-hand manipulation, …

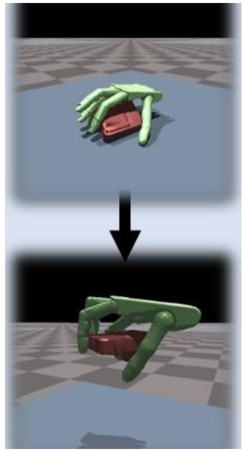
- Challenges in **universal dexterous grasping**:
  - High degrees-of-freedom, high-dimensional action space
  - Diverse object properties: geometry, friction, rigidity…
  - Complex real-world scenarios: clustered scene, tabletop settings, object orientation, …

Background

- Object-level generalization: grasping datasets + large-scale reinforcement learning (RL)

Human data / synthetic data in simulation

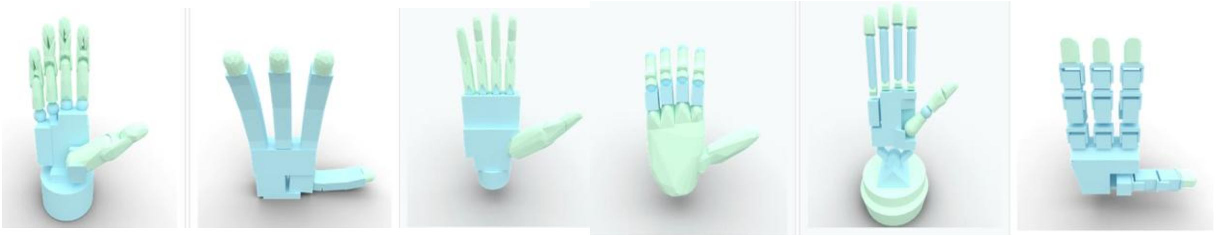




(UniDexGrasp)

Xu et al., 2023; Wan et al., 2023; Zhang et al., 2024

- Embodiment-level generalization: ?



Allegro hand: 4 fingers,16 DoF, 1.6x scale

Shadow hand: 5 fingers, >20 DoF, 1x scale

Background

- **Cross-embodiment learning**: learn a unified model for various embodiments
  - Establish embodiment-agnostic understanding on physical tasks and interactions
  - Enhance generalization

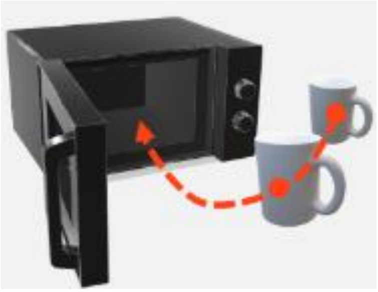
- Robot arms

Bousmalis et al., 2023
Wang et al., 2024
Chen et al., 2024

- Embodiment variants

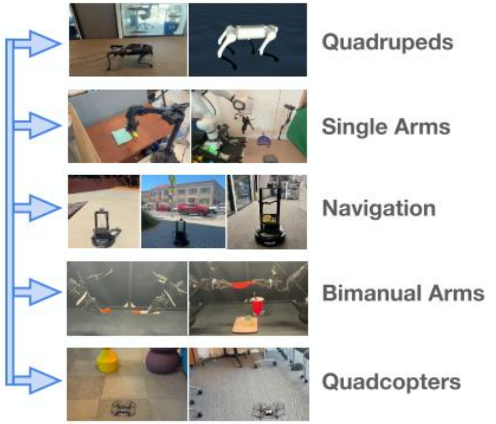
Hejna et al., 2021; Yu et al., 2022;
Liu et al., 2022; Patel et al., 2024

- Object-centric manipulation



Salhotra et al., 2023
Xu et al., 2024

- Unify all embodiments



Doshi et al., 2024
Yang et al., 2024

- We study how to train **cross-embodiment dexterous grasping** models

Problem formulation

- Tabletop dexterous grasping

POMDP $M_{h,\omega} = \langle \mathcal{O}, \mathcal{S}, \mathcal{A}, \mathcal{T}, R, \mathcal{U} \rangle$

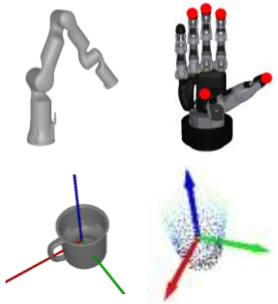
Dexterous hand embodiment $h \in \mathcal{H}$

Object $\omega \in \Omega$

- **Goal**: learn a cross-embodiment policy $\pi(a_t|o_t)$, maximizing expected return on all embodiments and objects

$$\sum_{h \in \mathcal{H}, \omega \in \Omega} \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right]$$

Problem formulation



- Observation space
  - Proprioception: joint positions, 3D keypoints
  - Object perception: 6D pose (in simulation) / point cloud (real world)

- Action space
  - Target joint positions for PD controller

- Challenges in cross-embodiment learning:
  - Different dexterous hands vary in DoFs and structures. Joint positions in **observations and actions** cannot be aligned and unified.
  - The policy should adapt to variation in hand sizes, shapes, …

Motivation

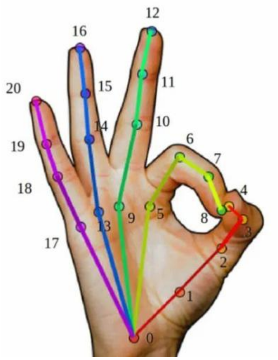
- ➢ Teleoperation: **humans' policy for cross-embodiment grasping**
  - A unified embodiment-agnostic policy: human can make decisions **based on visual feedback**, without prior knowledge on the robot embodiment.
  - **Retargeting**: mapping human hand poses to target joint positions for dexterous hands.

CrossDex: hand pose retargeting

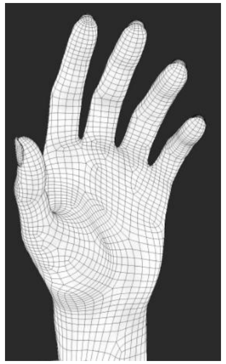
➢ The **MANO** hand model

16 axis angles of human hand joints $\theta \in \mathbb{R}^{48}$
Eigenvector of hand shape $\beta \in \mathbb{R}^{10}$

MANO



21 3D keypoints



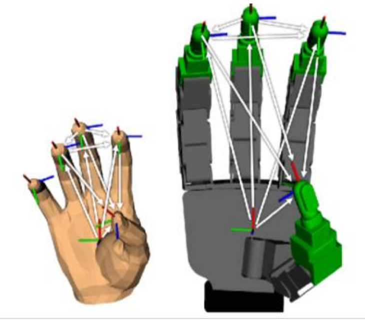
mesh

➢ **Hand pose retargeting**

Human hand pose:
21 keypoint positions

Retargeting

Robot hand pose: target joint positions



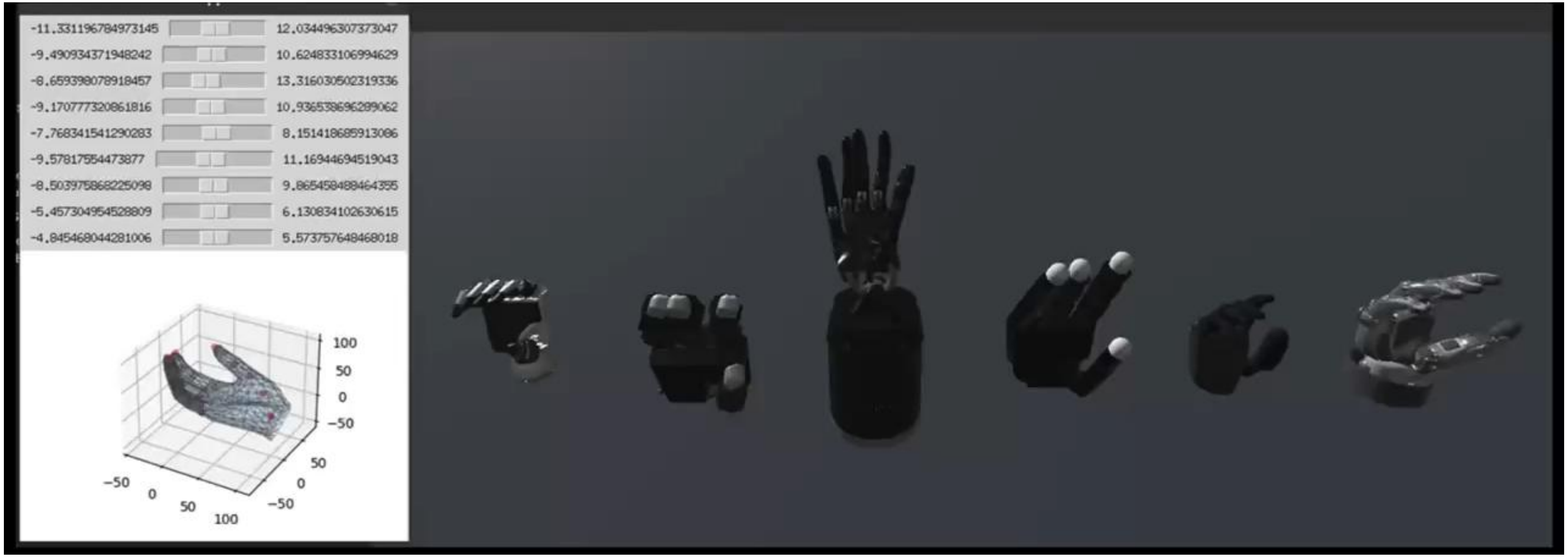
DexPilot: solve optimization problem
Maximize similarities of relative positions between keypoints

$$\min_{\boldsymbol{J}_t^h} S\left(f^h(\boldsymbol{J}_t^h), \boldsymbol{x}_t^M\right) + \|\boldsymbol{J}_t^h - \boldsymbol{J}_{t-1}^h\|^2$$

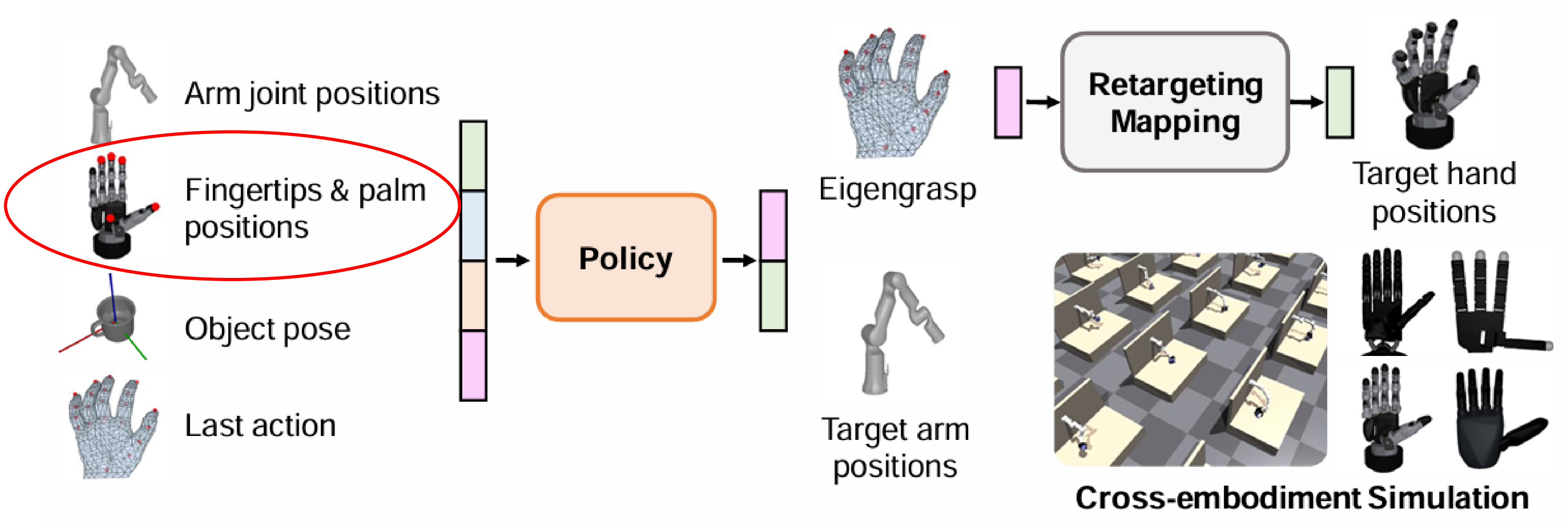
$$\text{s.t.} \quad \boldsymbol{J}_{lower}^h \leq \boldsymbol{J}_t^h \leq \boldsymbol{J}_{upper}^h,$$

CrossDex: eigengrasps

- If we use human hand axis angles θ as actions: high-dimensional action space, hardcoded joint limits, coupling between joints are not considered, …
- **Eigengrasps:** apply PCA, use eigenvectors to represent common hand poses $\{\boldsymbol{e}_i\}_{i=1}^{k}$
- Linearly combine the eigengrasps to generate diverse hand poses: $\theta = \sum_{i=1}^{k} w_i \boldsymbol{e}_i$
- CrossDex action space: use weights for first-k eigengrasps as actions. Use DexPilot to map the human hand poses to dexterous hand poses.

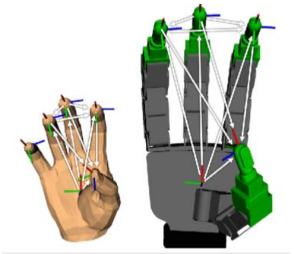


CrossDex: training

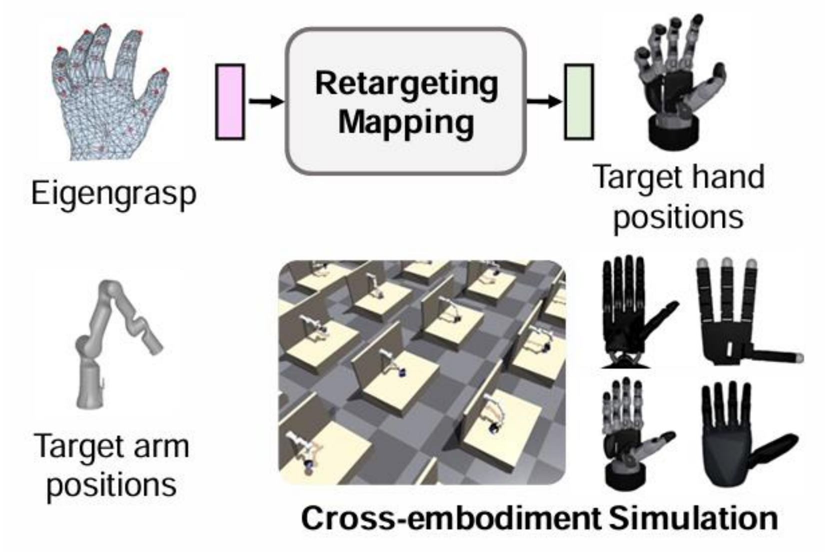


- Build parallel simulation in IsaacGym
- Teacher-student learning
  - RL: Train a **state-based** policy for each object. Grasp each object given object pose observations.
  - DAgger: distill all state-based policies into a **vision-based** policy. Grasp any object given point cloud observations.

CrossDex: training



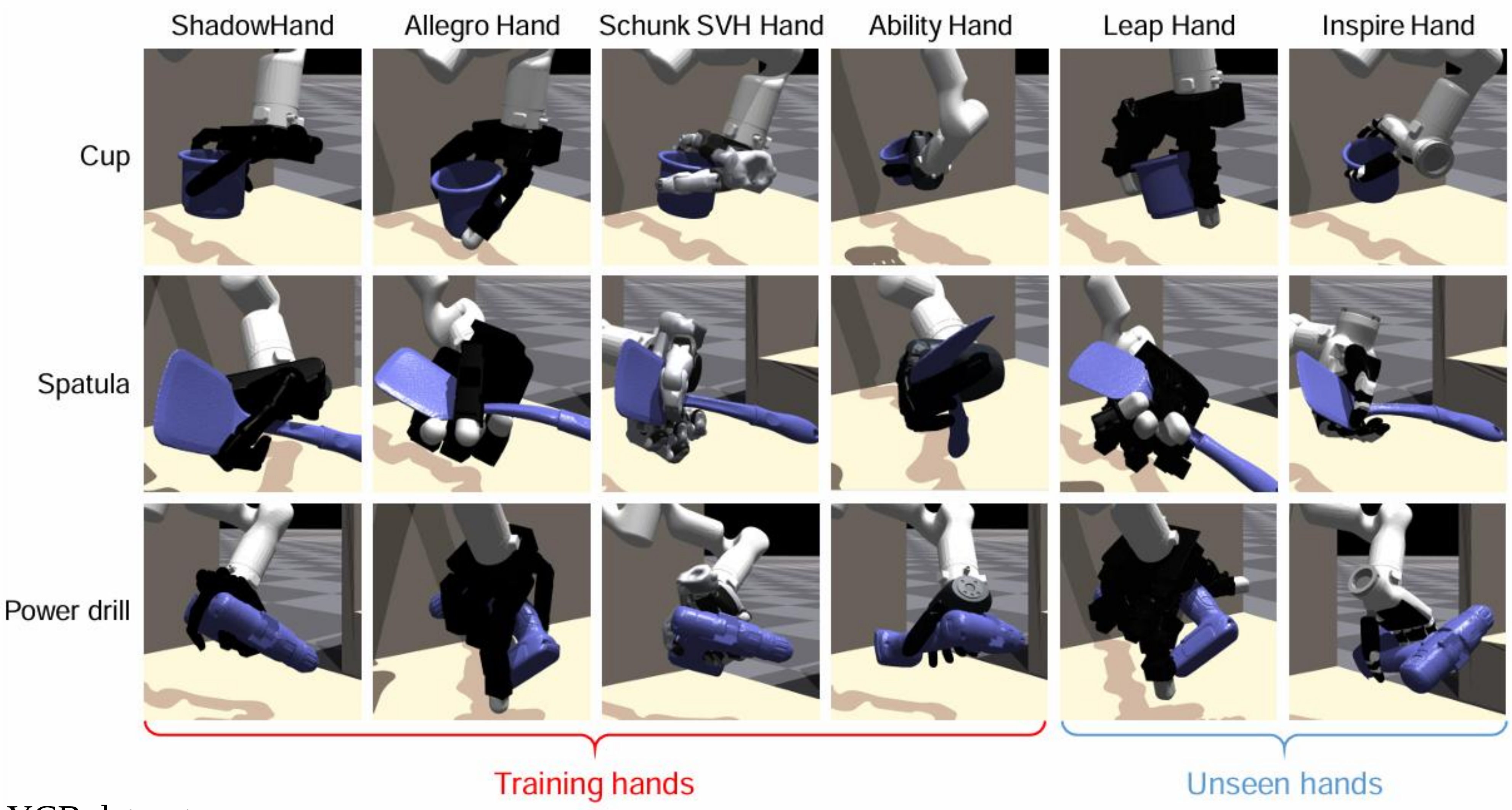
➢ Optimization-based retargeting: 300 FPS



➢ Train neural networks to fit the retargeting mappings on the hand pose dataset.

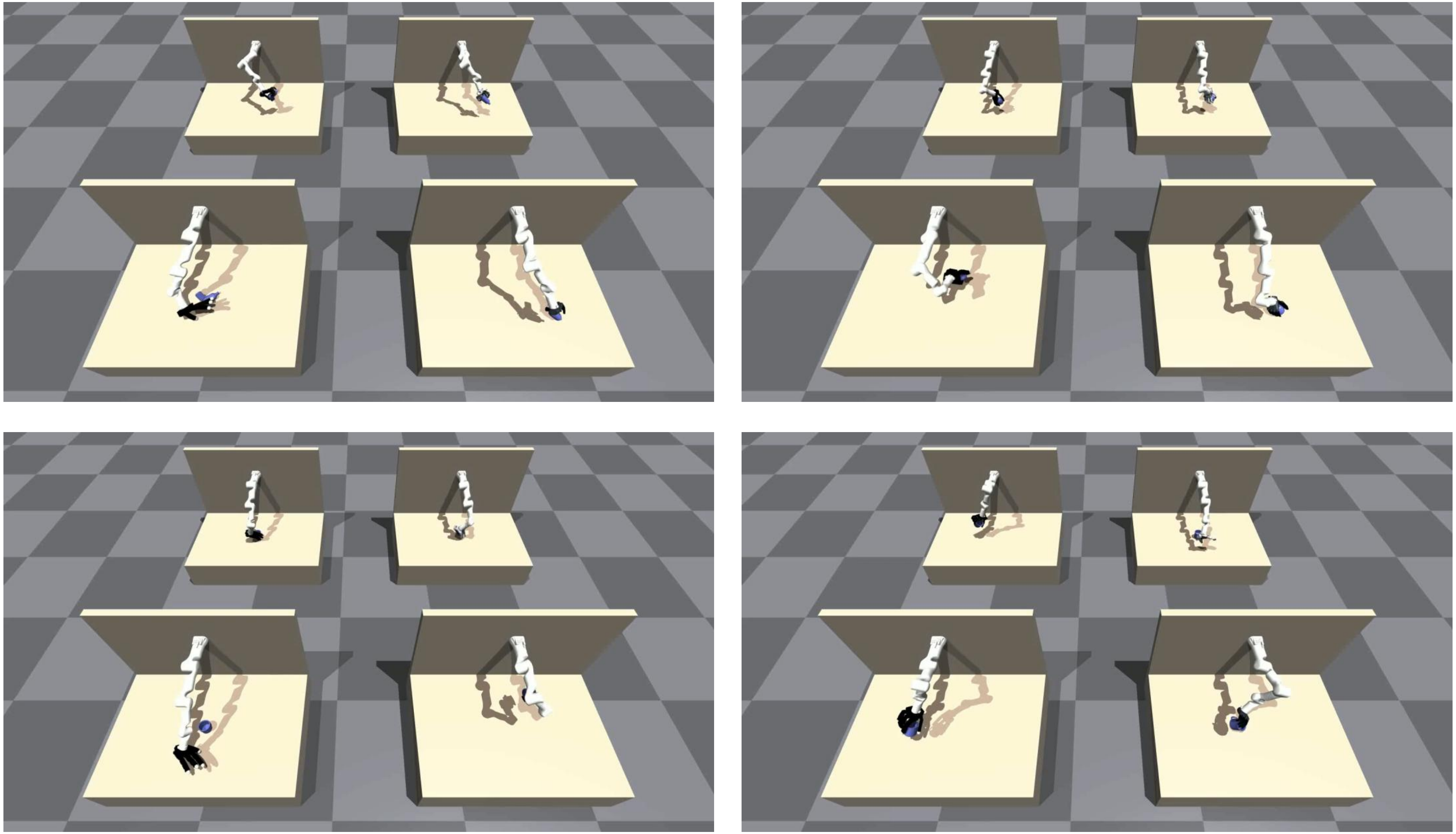
➢ IsaacGym simulation: 100000 FPS

CrossDex: experiments



YCB dataset

CrossDex: experiments



CrossDex: experiments

Zero-shot embodiment generalization

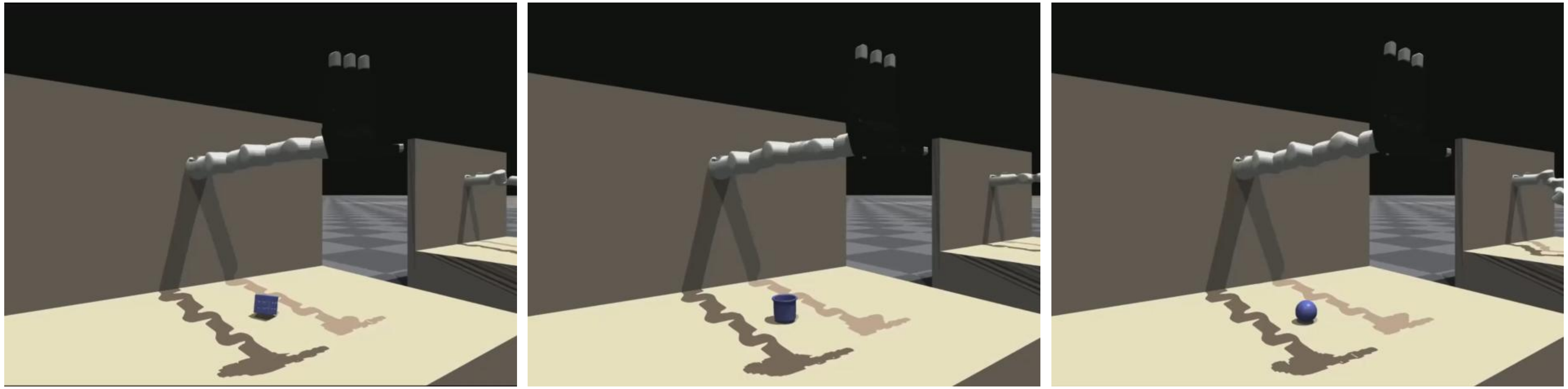
Method	Training hands (State)	Training hands (Vision)	Unseen hands (State)	Unseen hands (Vision)
MT-Raw-OA	**0.914**	0.782	0.054	0.162
MT-Raw-A	0.823	0.728	0.272	0.210
MT-Raw-O	0.884	0.779	0.046	0.145
CrossDex	0.885	**0.800**	**0.391**	**0.352**

Method	YCB (5 objects, state)	YCB (multi-task, vision)	GRAB (multi-task, vision)
No-Pretrain	0.758±0.122	0.436±0.159	0.313±0.373
MT-Raw-OA	0.701±0.002	0.417±0.007	0.651±0.007
MT-Raw-A	0.798±0.002	0.390±0.005	0.655±0.006
MT-Raw-O	0.708±0.014	0.385±0.003	0.616±0.018
CrossDex	**0.872**±0.013	**0.643**±0.009	**0.740**±0.009

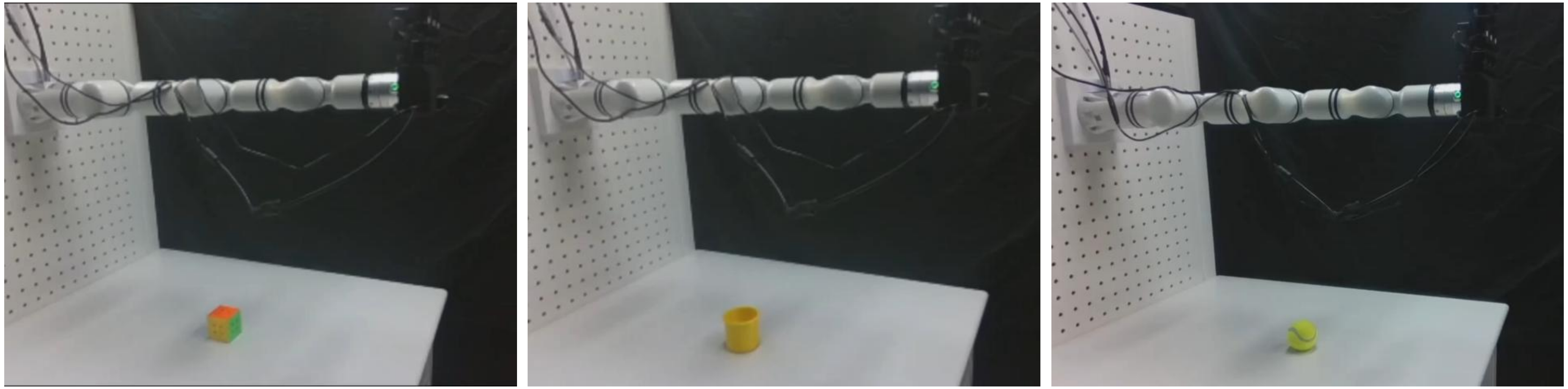
Finetuning on new embodiments

CrossDex: experiments

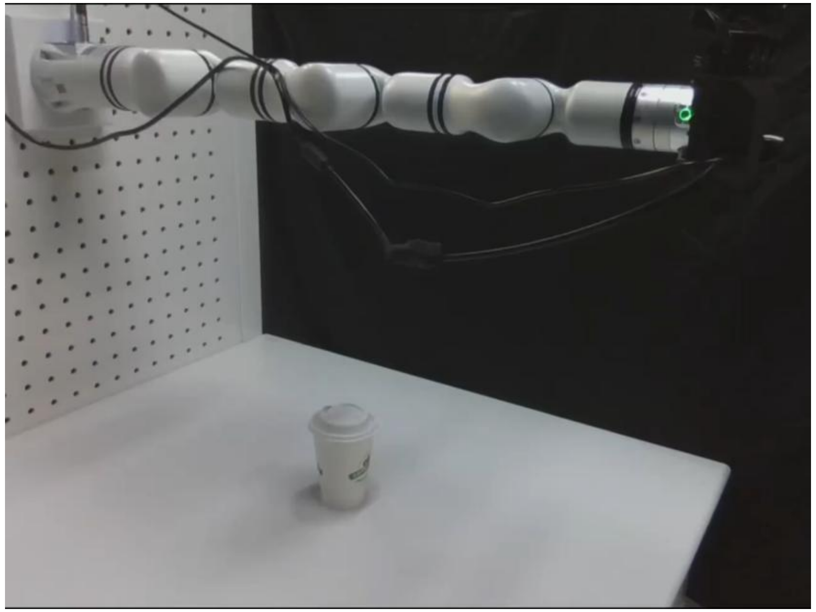
Simulation

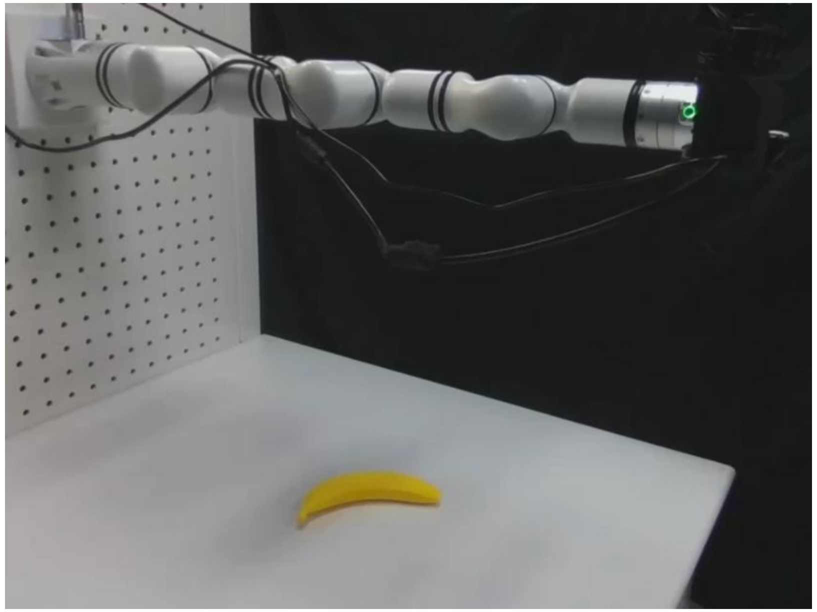


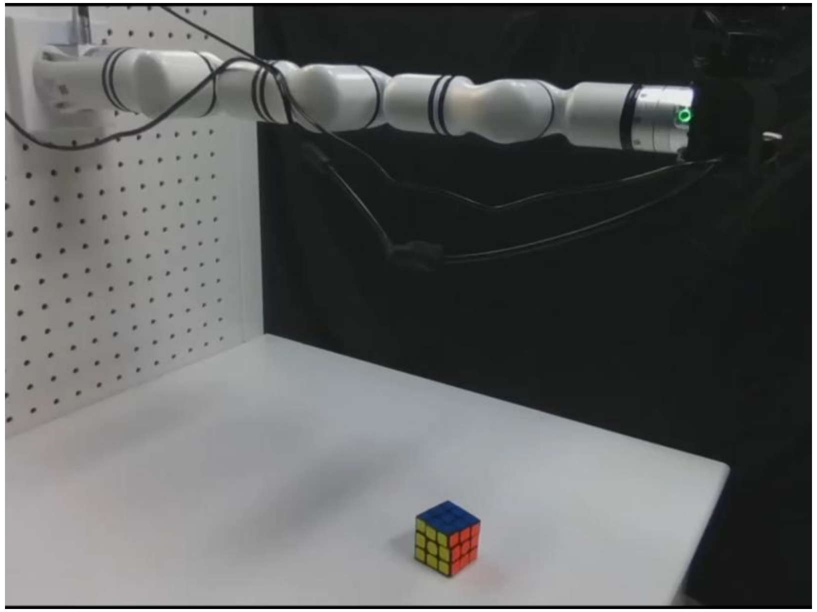
Real world



CrossDex: experiments







Conclusion and limitations

- CrossDex is an initial attempt towards a unified RL policy for various dexterous hands.
- We propose novel techniques -- eigengrasp action space, retargeting networks, unified observation space – to address the problem.
- Experiments on YCB demonstrates positive transfer in training and generalization to unseen hands.

- Performance of zero-shot generalization to unseen hands is still quite limited.
  - Include a wide range of dexterous hand models.
  - Context-based RL: learn from in-context trial-and-error.
- Extend to other manipulation tasks: object reorientation, functional grasping, …

Project page: https://sites.google.com/view/crossdex

Code: https://github.com/PKU-RL/CrossDex/

Thank you!