Learning Object-Centric Motion Priors from Human for Robotic Dexterous Manipulation

Zhengdong Hong, Guofeng Zhang*

State Key Lab of CAD&CG, Zhejiang University
12321092@zju.edu.cn, zhangguofeng@zju.edu.cn

Abstract

Manipulating diverse objects with multi-fingered dexterous hands is challenging due to the high dimensionality and complex dynamics. Human-Object Interaction (HOI) datasets provide rich knowledge about task information and embodied interactions. Instead of solely imitating the human demonstrations, our method learns to holistically predict future hand-object states by leveraging these datasets. The predicted future states of the object can serve as a general-purpose reward term for reinforcement learning, reducing reliance on task-specific reward engineering and enhancing generalization across tasks. We conduct extensive experiments on three manipulation tasks in simulation and the real world. Our approach outperforms existing SOTA methods in both success rate and generalizability on novel objects. Furthermore, we validate the cross-embodiment compatibility of our methods by successfully deploying the skills on different robot hands.

Introduction

Achieving human-level dexterous manipulation is a critical but challenging goal for robotics, primarily due to the high degrees of freedom in robot hands and complex contact dynamics. Previous methods, such as model-based trajectory optimization (Pang et al. 2023; Jin 2024), struggle with these dynamic complexities. Meanwhile, model-free reinforcement learning (RL) (Rajeswaran et al. 2017; Gupta et al. 2016; Akkaya et al. 2019; Chen et al. 2023; Wang et al. 2024b) is often inefficient, demanding extensive exploration and task-specific reward engineering. While imitation learning from robot data (Chi et al. 2023) presents a viable solution, the required large-scale data collection is frequently impractical for research labs (Wang et al. 2024a; Black et al. 2024). In contrast, the vast amount of available human data, which implicitly encodes rich manipulation experience, offers a promising and scalable alternative for learning dexterous manipulation.

What should we learn from the human demonstration? To answer this, let's look at how humans learn. Children learn to interact with the physical world using the following two-stage learning paradigm. The first stage involves forming an abstract task concept through watching others play. In the second stage, this conceptual knowledge is grounded and refined through physical experimentation by themselves, facilitating the development of operational competence. Similarly, prior literature (Qin, Wu, and Liu 2022; Chen et al. 2024b; Shi et al. 2025; Chen et al. 2024a; Li et al. 2025) investigates a similar two-stage pipeline for learning dexterous manipulation skills. Most of them either focus solely on imitating human motions or designing task-specific rewards, then testing on only one task, which makes it hard to draw a general conclusion from that. Some of these works overlook an important fact: while humans are manipulating objects, they are also subconsciously predicting the future state changes of the objects. This ability to predict the future state of objects not only helps humans understand tasks but also enables them to perform more general operations on different tasks. Inspired by that, our observation is that considering human-object as a whole and learning human manipulation priors from hand-object interactions provides more general and efficient task-level guidance for the second-stage low-level policy learning. Instead of using human-designed task-specific rewards like lifting rewards in previous work (Chen et al. 2024b; Shi et al. 2025), we believe that human-object interaction priors encourage robots to manipulate the object in the way described in the dataset, which is task-agnostic to different tasks.

Specifically, we propose a pipeline to auto-regressively predict future hand-object states holistically. And we use the predicted hand and object states to guide the low-level policy learning in different tasks without task-specific reward design in low-level reinforcement learning in a physical simulation. Then we perform zero-shot sim-to-real skill transfer to the real-world robots. Our contributions are as follows:

- We propose an efficient learning diagram to utilize human-object interaction priors for robot skill learning, without the need for real-robot data.
- Predicting hand-object priors holistically and properly utilizing the predicted object states can not only facilitate reinforcement learning by alleviating the heavy reliance on task-specific reward design, but also enable our pipeline to be more general to various task settings.
- We conducted extensive real-robot experiments on various tasks, embodiments, and object settings, outperforming existing SOTA methods in all three challenging tasks.

*Corresponding author.

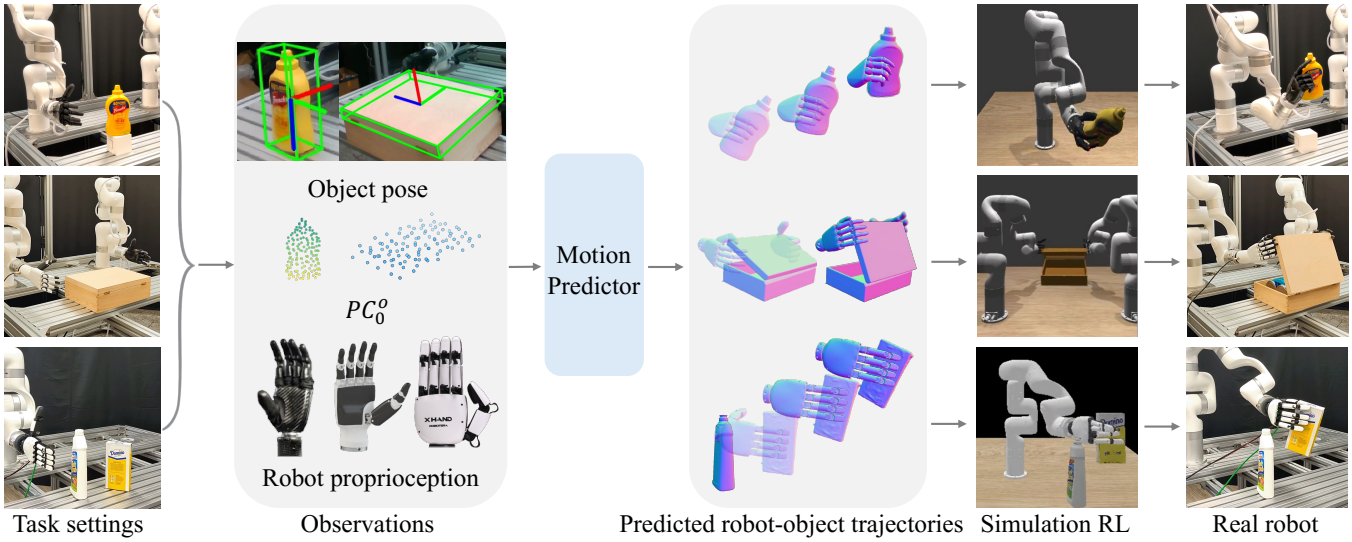

Figure 1: Our method begins by extracting observations from the initial robot-object configuration. These observations are input into a motion predictor to generate a reference trajectory. Guided by this trajectory, we employ reinforcement learning to train the target policy. Finally, the learned policy is deployed directly to the physical robot via zero-shot Sim-to-Real transfer. Notably, since the learned priors are general to hand morphologies, our framework is adaptable to various dexterous hands.

Related Work

Dexterous Manipulation

Dexterous manipulation is complicated by the high dimensionality of the end effector. This complexity poses a problem for traditional trajectory optimization techniques (Rus 1999; Mordatch, Popović, and Todorov 2012; Wu, Guo, and Liu 2022), as they require precise dynamic models that are impractical for real-world applications. Learning-based methods address this by learning an observation-to-action policy directly from data, but they introduce new challenges. Reinforcement learning approaches (Qi et al. 2023; Yuan et al. 2024; Wang et al. 2024b) can be sample-inefficient and depend on meticulous reward engineering. On the other hand, imitation learning methods (Bain and Sammut 1995; Chi et al. 2023), which use data from teleoperation (Qin et al. 2023; Wang et al. 2024a), are often constrained by the expense of data collection and the quality of the expert demonstrations. Vision-Language-Action methods (Pan, Junge, and Hughes 2024; Zhong et al. 2025) aim at utilizing pretrained language models to boost the network performance, while only demonstrating tasks not harder than pick-and-place. Grasp synthesis methods (Ye et al. 2025; Wan et al. 2023) are mainly tested in simulation, and the potential in more challenging tasks beyond pick-and-place is not fully investigated. Our methods combine the merits of using human data and scaled-up simulation, enabling real-robot dexterous manipulation with high success rates.

Learning Dexterous Manipulation from Human

Internet human video data contains abundant information about human experiences. Previous research (Mendonca, Bahl, and Pathak 2023; Ma et al. 2022; Nair et al. 2022) seeks to learn general visual representations from internet-scale human video data to guide robot learning. However, those methods only showcase results on a 1-DoF robot gripper since the features trained on 2D human videos lack the 3D information necessary for learning high-dimensional skills like dexterous manipulation. Different from internet video human data, human-object interaction (HOI) data (Chao et al. 2021; Fan et al. 2023; Liu, Liu, and Jiang 2022) are usually collected by multi-view RGBD cameras, with extracted and labeled 3D human pose estimations and 6-DoF object pose estimations, which contain abundant 3D information necessary for learning dexterous manipulation. Given the 3D motions of the human hand, an intuitive idea is to directly transfer human motions to robot actions. (Qin et al. 2023) proposed a retargeting system based on inverse kinematics to map the human hand motions to robot hand states and actions. (Qin, Wu, and Liu 2022; Liu et al. 2025) further prove that the retargeted trajectories can be used to train imitation learning policies in different tasks.

Following them, (Chen et al. 2024b; Zhou et al. 2024; Shi et al. 2025) apply trajectory augmentation and train a state-based policy based on that. Different from direct mapping, (Chen et al. 2024a) proposes to train a human motion prediction network given a target object trajectory, then augment and refine it in physical simulation using Reinforcement Learning (RL). However, defining target object trajectories in open world requires extra efforts. (Li et al. 2025) train a 2-stage process imitating human, but their input is fixed object initial poses with object pose randomization as 0, constraining open-world applications. To address this, (Zhao et al. 2024) introduces human-in-the-loop adjustments in generating reference trajectories, but it requires human labor. (Singh et al. 2024) requires real-world robot demonstrations or RL

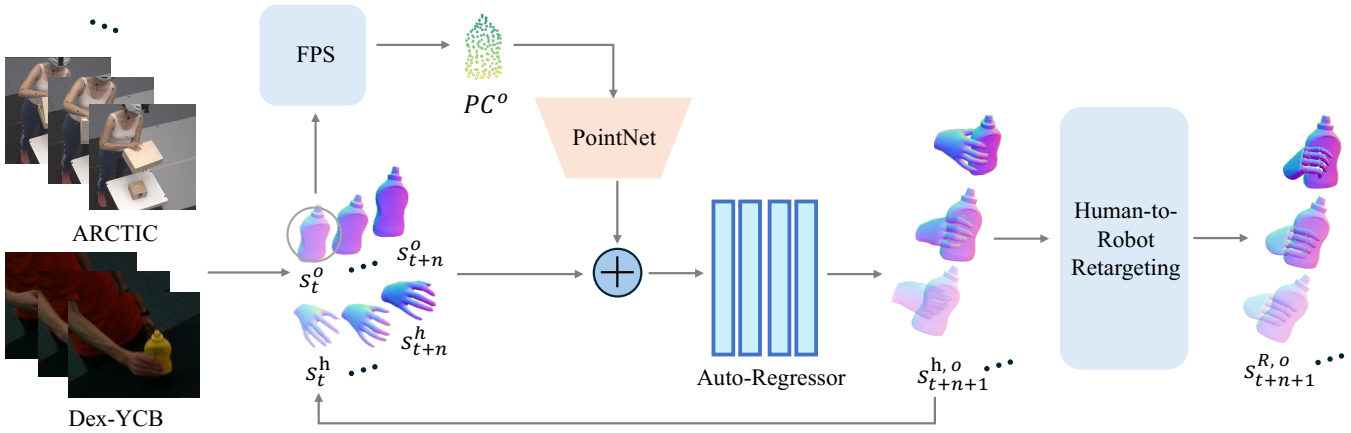

Figure 2: Illustration of components in our HOI motion predictor.

to finetune the policy. Most of them focus on a single task and can hardly generalize to novel object configurations. We demonstrate our applicability in various challenging tasks under different object configurations.

Methods

Our proposed pipeline is illustrated in Fig. 1. The process begins by extracting observations from the initial state of a real-world robot and object scene. These observations are then fed into a motion predictor to generate a reference trajectory. Using this trajectory as guidance, we employ a reinforcement learning framework to learn the desired skill policy. Finally, this learned policy is transferred directly to the physical robot using a zero-shot sim-to-real approach.

Hand-Object Poses Prediction

Fig. 2 is our hand-object-interaction (HOI) motion predictor. We use Dex-YCB (Chao et al. 2021) and ARCTIC (Fan et al. 2023) as our datasets. Both of them have well-labeled 6DoF hand-object pose pairs for each timestamp, noted as $\hat{s_t^h}$ which is represented by MANO (Romero, Tzionas, and Black 2022) hand poses and $\hat{s_t^o}$ which are the quaternion $\mathbf{q_t^o}$ and the translation $\mathbf{T_t^o}$ of a object pose. We use a coordinate system with the geometric center of the object at $t = 0$ as the origin. For articulated objects like a laptop and a box, we record the relative pose of the upper part to the lower part (base). For each motion sequence, we first use farthest point sampling (FPS) (Qi et al. 2017b) to extract a 3D point cloud PC^o consisting of 100 points from the object mesh, which is later processed by a PointNet (Qi et al. 2017a) to get a feature about the object shape information. We use a 6-layer GPT-2 transformer (Radford et al. 2019) to predict future hand-object pose pairs $s_{t+n+1}^{h,o}$ using the history observations $s_t^{h,o}, s_{t+1}^{h,o}, ..., s_{t+n}^{h,o}$ in an auto-regressive manner. The history observations are concatenated with the PointNet feature to feed into the transformer, and the training is performed as supervised learning:

$$\mathcal{L}(s_t^h, s_t^o) = \sum_{t=1}^{\Omega} \alpha \|s_t^h - \hat{s_t^h}\|^2 + \beta \|s_t^o - \hat{s_t^o}\|^2.$$

We use $n = 10$ in our experiments. The predicted observations $s_{t+n+1}^{h,o}$ are taken as input again for the next prediction until the length of the predicted sequence reach the sequence length Ω in datasets. We utilize a single, unified model for all datasets, adapting the input as necessary. For the bimanual ARCTIC dataset (Fan et al. 2023), both hands are directly fed into the network. To accommodate data from Dex-YCB (Chao et al. 2021), we apply zero-padding to align input dimensions. In practice, we use $\alpha = 1, \beta = 2$. β is larger than α since the dimensions that belong to the object in the input have a lower ratio than the hand.

During inference on real robots, we regard robot hands as subcollections of human hands and directly apply MANO estimation on them. We extract 3D coordinates of robot joints $J_{Robot,i}$ (semantically corresponding to a subset $J_{MANO,i}$ of 21 joints ($N = 21$) in MANO), then modify FrankMocap (Rong, Shiratori, and Joo 2020) by replacing the original optimization goal

$$\sum_{i=1}^{N} ||\Pi((R,T) \cdot J_{MANO,i}(\theta,\beta)) - J_{2D,i}||^2 + L_{reg}(\theta,\beta)$$

with

$$\mathcal{L}_{align} = \sum_{i=1}^{N} ||J_{MANO,i}(\theta,\beta) - J_{Robot,i}||^2 + L_{reg}(\theta,\beta),$$

using robot proprioception and forward kinematics to get 3d joint positions $J_{Robot,i}$, instead of projecting ($\Pi((R,T))$ the 3d joints $J_{MANO,i}$ to 2D and optimize in 2D. During optimization, shape parameters β will be adaptively adjusted for different limb lengths, making it compatible with different types of robot hands with distinct sizes. During inference time, the object point cloud is attained by applying farthest point sampling (FPS) (Qi et al. 2017b) to the object mesh, which is placed under 6DoF object pose estimation by model-based tracking in FoundationPose (Wen et al. 2024).

Robot-Object Trajectories Extraction

The predicted hand motions s_t^h are further converted to motions of a robotic hand s_t^R using optimization-based hand

motion retargeting. Following AnyTeleop (Qin et al. 2023), the loss function minimizes the distance between the robot and human fingertip positions:

$$\mathcal{L}_{\text{retarget}}(q_t) = \sum_{i=1}^{5} \|x_t^i - f_i(q_t)\|^2 + \beta \|q_t - q_{t-1}\|^2,$$

subject to:

$$q_l \leq q_t \leq q_u,$$

where q_t is the robot's joint configuration at time t, and $f_i(q_t)$ is the corresponding robot fingertip position calculated via forward kinematics to match the human's fingertip position, x_t^i. The robot's joints are constrained by lower (q_l) and upper (q_u) bounds. To ensure temporal smoothness, an additional penalty term is weighted by β. We solve this optimization problem using the NLopt solver (Qin et al. 2023). In practice, we use the weight $\beta_{retargeting} =$ with $q_l = 0°$ and $q_u = 90°$.

Trajectory-Guided Reinforcement Learning

Due to the embodiment gap, the retargeted robot trajectories are imperfect and require further refinement. However, these predicted trajectories provide crucial guidance for learning robotic skills via reinforcement learning (RL). We represent the task using the Markov Decision Process (MDP) framework $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where the components correspond to the state space ($\mathcal{S}$), action space ($\mathcal{A}$), transition model (P), reward function (r), and discount factor (γ), respectively.

We train a state-based skill policy, π_i. The policy's input state $s_t \in \mathcal{S}$ is composed of robot joint positions and 6D object poses. The output action $a_t \in \mathcal{A}$ consists of delta end-effector poses and delta hand joint positions.

$$a_t \sim \pi(a_t \mid s_t).$$

At each time step t of the interaction, the environment transitions to s_{t+1} based on $P(s_{t+1} \mid s_t, a_t)$ and provides a reward r_t. During control, the target end-effector pose is converted to arm joint positions via inverse kinematics (IK) using Pinocchio (Carpentier et al. 2019). Each policy π_i from a set Π is optimized to maximize the expected cumulative reward, conditioned on a reference trajectory τ_i from a human demonstration with task horizon H:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t \mid \tau_i)\right].$$

Our reward function consists of four items:

$$R = \lambda_1 R_f + \lambda_2 R_o + \lambda_3 R_{\text{contact}} + \lambda_4 R_{\text{success}}.$$

The first term, robot following reward $R_f = \beta_1 R_j + \beta_2 R_{\text{ee},x} + \beta_3 R_{\text{ee},r}$, encourages the robot to follow reference hand motions. The first component $R_j = 1 - \tanh(\alpha_1 \|q^t - \hat{q}^{t'}\|)$ encourage the robot hand to follow reference joint positions, where q_t and $\hat{q}_{t'}$ are robot hand joint positions at timestep t and reference robot hand joint positions at corresponding timestep t'. The second one $R_{\text{ee},x} = 1 - \tanh(\alpha_2 \|x_{\text{ee}}^t - \hat{x}_{\text{ee}}^{t'}\|)$ and the third one $R_{\text{ee},r} = 1 - \tanh(\alpha_3 \phi(\theta^t, \hat{\theta}^{t'}))$ constrains robot End-Effector (EE) pose where x_{ee} are robot EE position and ϕ computes the angular distance between robot EE orientation θ^t with reference $\hat{\theta}^{t'}$.

The second term is an **object following reward**. Instead of using human-designed task-specific reward like lifting reward in previous work (Chen et al. 2024b; Shi et al. 2025), we utilize the predicted object states S_t^o as guiding signals to encourage the robots to manipulate the object in the way that is in the dataset, which is more generalizable to different tasks:

$$R_o = e^{-\alpha_4 \|\mathbf{T_t^o} - \hat{\mathbf{T}}_\mathbf{t}^\mathbf{o}\|_2} + e^{-\alpha_5 \|log(\hat{\mathbf{q}}_t^{o\,-1} \otimes \mathbf{q}_t^o)\|_2},$$

where $log(\cdot)$ is the Logarithmic Map which converts a quaternion to its axis-angle representation, and $\otimes$ denotes the Hamilton product.

We encourage finger exploration through R_{contact} , which gives a reward only when the thumb finger and at least one other finger contact with the object. The contact information is derived from the collision check in SAPIEN simulation (Xiang et al. 2020). In obstacle avoidance tasks, we further add a large penalty when any part of the hand or the arm contact the obstacle, forcing the policy to explore a plausible route to avoid obstacles. We give a large reward R_{success} when task is finished with success. We use PPO (Schulman et al. 2017) for state-based policy learning with delta end-effector pose control, where we randomize initial object position within ± 5 cm in xy-plane and rotation within $\pm 30°$ around the z-axis.

We carefully tuned the reward weights. In our experiments, we use $\lambda_1 = \lambda_2 = \lambda_3 = 1, \lambda_4 = 15, \lambda_5 = 3, \beta_1 = 0.1, \beta_2 = \beta_3 = 1, \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = 1$ where $R_{success} = 1$ when task success, otherwise $R_{success} = 0$. In *YCB grasping with obstacle* task, the weights of the contact reward are modified to be -20 when there is any contact between the robot hand links or the robot arm links with the obstacle.

Sim-to-Real Transfer

We have several designs to overcome the control gap and perception gap between the simulation and the real world, enabling zero-shot Sim-to-Real transfer. Before training the reinforcement learning policy, we perform system identification (Ma et al. 2025) to mitigate the control gap. Specifically, we control the simulated robot and the real robot to move in the same direction in the same time interval and minimize the error between the two by adjusting the parameters of the simulated robot, including PID parameters and force limits for each joint motors. For different robot hand embodiments, we tune the control parameters for each to minimize the control gap when switched to another embodiment. Besides, we adopt domain randomization (Tobin et al. 2017) by artificially adding random noise to the observations as well as adding simulation parameter randomizations to friction, object scale, and object weights in the simulation to improve the robustness of the trained policies to the real-world noise.

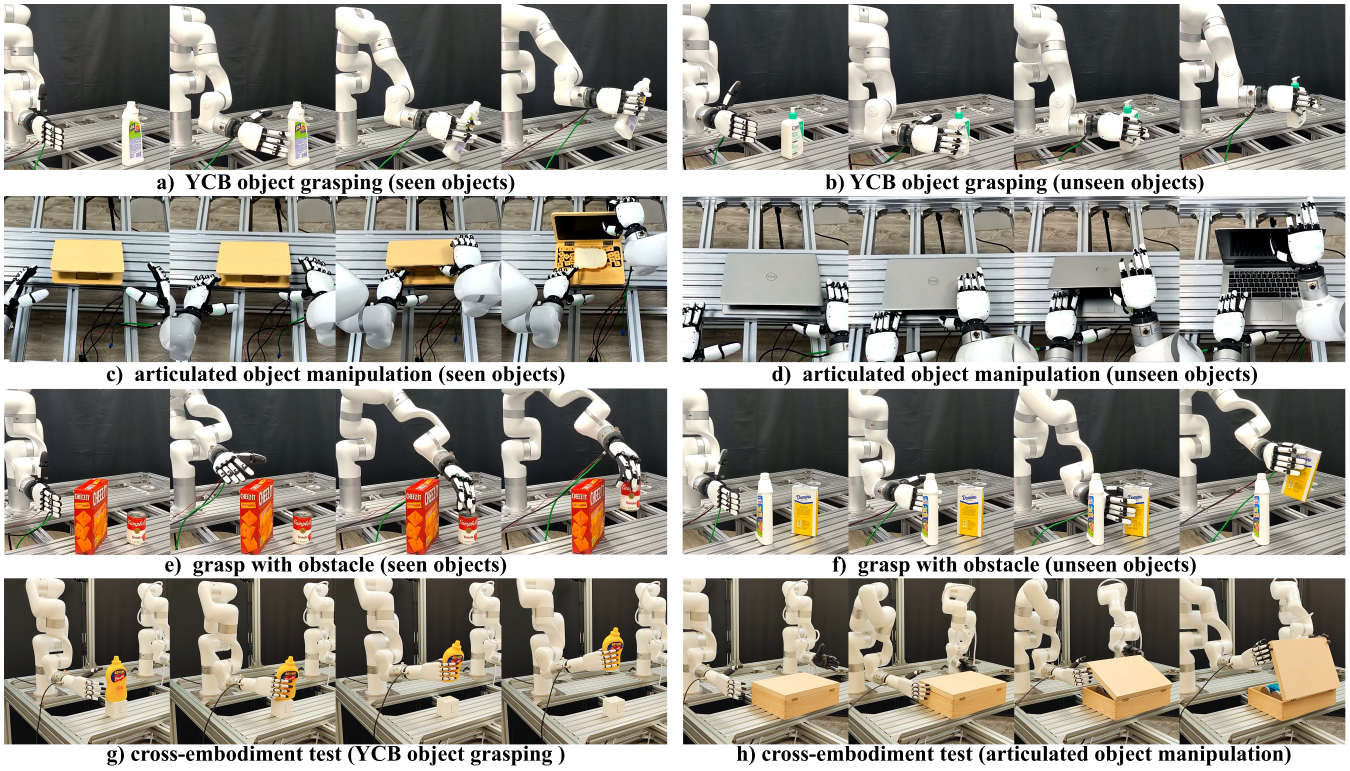

Figure 3: Qualitative results from real-world experiments.

Experiments

Experiment Setting

As shown in Fig. 3, our experiments covers the following 4 settings:

- 1) *YCB object grasping*: the success is defined when the robot hand successfully grasps a target YCB object placed on the platform and lifts it up to 10 cm above its initial height.
- 2) *Articulated object manipulation*: the task success is defined as the hinged object being rotated more than 30 degrees along the axis.
- 3) *Grasping with obstacles*: the distinction between this task and a standard grasping task is the deliberate introduction of an obstacle along the robot's path towards the target object. An additional constraint is that any part of the robotic arm and dexterous hand must not collide with the obstacle at any point during the entire process of grasping the target object.
- 4) *Cross-embodiment test*: we replace the right-side robot hand with a different model. We then reuse the motion predictor to generate human-object interaction trajectories and apply retargeting adapted to the specific robot configuration. Finally, we perform trajectory-guided reinforcement learning to learn the skills, using identical reward terms and parameters for different robot hands.

Parameters	Values
max_steps	1000000
bootstrap at done	True
gamma	0.95
gae_lambda	0.9
learning_rate	3e-4
num_envs	1024

Table 1: Training parameters of PPO policies.

Abbreviation:

In the following paragraphs, the terms **seen** and **unseen** (as applied to objects or settings) refer to whether that configuration was used to train our HOI motion predictor and the corresponding motion extractors for the baseline methods.

Simulation Configuration

For simulation, we employ SAPIEN3 (Xiang et al. 2020) and OpenAI-Gymnasium (Towers et al. 2024) to enable GPU-parallel training and rendering, we have 1024 environments running in parallel on single NVIDIA RTX 4090 GPU (24GB) and i7-14700K CPU, with average steps per second at 10000, and training for 1M steps for the policies to converge. Parameters used in training the PPO reinforcement learning policies are shown in Tab. 1.

Real-World Robot Setup

Our bimanual robot platform consists of two xArm-7 arms, each mounted with a PSYONIC Ability hand, a ROBOTERA XHand1 or an Inspire robot hand. Object pose estimation is performed using model-based tracking in (Wen et al. 2024) based on RGB-D data from a RealSense D435i camera, which was calibrated using EasyHec++ (Hong, Zheng, and Chen 2024).

Baselines

For YCB object grasping tasks, we compare our results with existing SOTA methods. They are ViviDex (Chen et al. 2024b), AdaDexGrasp (Shi et al. 2025), ManipTrans (Ma et al. 2025), HOP (Singh et al. 2024) using RL for finetuning, and PPO w/o following, which substitutes following reward R_f and R_o using a reaching reward:

$$R_{reaching} = \lambda_5(1 - \tanh(\alpha_6 \|T_t^{TCP} - \hat{T}^o\|)),$$

where T_t^{TCP} is the translation of the robot TCP pose, which refers to the pose of the palm of the robot hand, and $\hat{T}^o$ is the global translation of the object. When there is a novel object configuration, ViviDex (Chen et al. 2024b) and AdaDexGrasp (Shi et al. 2025) use existing reference trajectories in the dataset. For ARCTIC articulated object manipulation, we compare with Obj-Dex (Chen et al. 2024a), ManipTrans (Ma et al. 2025), and PPO w/o following. For grasp with obstacle tasks, we also compare ours with HOI Pretrain (Singh et al. 2024). We use the PSYONIC Ability hand, ROBOTERA XHand1 or Inspire robot hand for the cross-embodiment test and compare the results. All the simulation experiments are tested with 5 seeds, and we report the average success rates in policy evaluation using 100 trials each after the convergency of RL training. All the real-world robot experiments are repeated 20 times to get success rates.

YCB Object Grasping

We test the methods on different YCB objects (5 objects in Dex-YCB datasets (Chao et al. 2021), 5 objects from YCB datasets (Calli et al. 2015) which are not included in DexYCB datasets) and 5 different initial object poses with randomized position and rotation inside the size of an A4 paper. The task success is defined as the object being grasped and lifted up over 10 cm. Success rates in simulation and real-world robots are reported in Tab. 2 and Tab. 3, respectively. ViviDex and AdaDexGrasp struggle to handle non-DexYCB objects. ManipTrans has difficulty dealing with novel initial object poses. Our method demonstrates consistent performance across different task configurations.

Articulated Object Manipulation

We test the performance of the methods on 3 objects in ARCTIC (Fan et al. 2023) dataset, which are *wooden laptop*, *wooden box*, *mixer*, and 3 objects in our daily life: *real laptop*, *larger box*, *toy coffee machine*. The task success is defined as the hinged object rotating more than 30 degrees along the axis. We report success rates in simulation and real-world robots in Tab. 4 and Tab. 5, respectively, where (Li et al. 2025), (Chen et al. 2024a) fail to handle the novel daily objects, while we still observe success in ours in simulation and real robots.

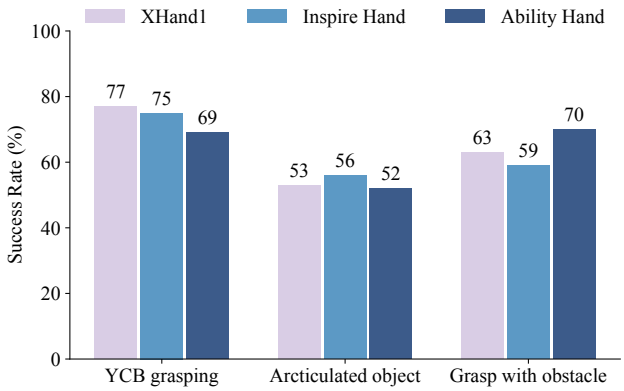

Figure 4: Cross-embodiment test on 3 robot hands.

Grasping with Obstacle

The distinction between this task and a standard grasping task is the deliberate introduction of an obstacle along the robot's path towards the target object. While the obstacle's mesh is known, its type and position are variable. An additional constraint is that the robotic arm and dexterous hand must not collide with the obstacle at any point during the process of grasping the target object; otherwise, the task is considered a failure. We select 5 objects in the DexYCB datasets and 5 3D-printed object to make up our obstacle library and choose one obstacle from it each time. In this experiment, a large penalty item $R_{obstacle}$ is introduced to each method when any part of the hand or the arm contacts the obstacle during RL training, forcing the policy to explore a plausible route to avoid obstacles. Besides, we selectively filters out all time frames where the dexterous hand meshes from reference trajectories collide with obstacles by cuRobo motion planning, disabling the following reward in these frames for ours and baselines. We report success rates in simulation and real-world robots in Tab. 6 and Tab. 7.

Cross-Embodiment Test

To validate the generalizability of our method across different robotic platforms, we replace the right-side robot hand in the robot setting with another robot hand. We then re-execute the retargeting and reinforcement learning algorithms based on the trajectories generated by our motion predictor and evaluated the performance on various tasks in the real-world robots. See Fig. 4, our methods perform well across different embodiments of dexterous hands, showing consistent performance in different tasks.

Ablation Study on the Reward Design

To validate the effectiveness of our reward design, we conduct ablation studies by: 1) removing the object following term R_o, and 2) removing the contact reward $R_{contact}$. We evaluate these on the YCB object grasping task using 5 objects (5 seeds each) from the DexYCB dataset and report the

	DexYCB objects	Non-DexYCB objects	DexYCB objects (novel poses)	Avg.
ViviDex (Chen et al. 2024b)	0.87	0.37	**0.75**	0.66
AdaDexGrasp (Shi et al. 2025)	0.82	0.21	0.64	0.56
ManipTrans (Li et al. 2025)	0.61	0.28	0.03	0.31
HOP (Singh et al. 2024)	0.63	0.59	0.54	0.59
PPO w/o following items	0.37	0.42	0.27	0.35
Ours	**0.90**	**0.86**	**0.75**	**0.84**

Table 2: Quantitative comparisons in simulation for *YCB object grasping* task, with task success rates reported.

	DexYCB objects	Non-DexYCB objects	DexYCB objects (novel poses)	Avg.
ViviDex (Chen et al. 2024b)	0.35	0.40	0.25	0.33
AdaDexGrasp (Shi et al. 2025)	0.80	0.20	**0.65**	0.55
ManipTrans (Li et al. 2025)	0.60	0.30	0.05	0.32
HOP (Singh et al. 2024)	0.60	0.45	0.40	0.48
PPO w/o following items	0.30	0.40	0.20	0.30
Ours	**0.85**	**0.80**	**0.65**	**0.77**

Table 3: Quantitative comparisons in real-world robots for *YCB object grasping* task, with task success rates reported.

	ARCTIC objects	Daily objects	Avg.
ManipTrans	0.61	0.00	0.31
Obj-Dex	0.74	0.05	0.39
PPO w/o follow.	0.00	0.00	0.00
Ours	**0.79**	**0.53**	**0.66**

Table 4: Success rates in simulation for *articulated object manipulation* task.

	ARCTIC objects	Daily objects	Avg.
ManipTrans	0.55	0.00	0.28
Obj-Dex	**0.70**	0.00	0.35
PPO w/o follow.	0.00	0.00	0.00
Ours	**0.70**	**0.35**	**0.53**

Table 5: Success rates in real-world robots for *articulated object manipulation* task.

	DexYCB obstacle	3D-printed obstacle	Avg.
ViviDex	0.17	0.38	0.28
AdaDexGrasp	0.55	0.67	0.61
ManipTrans	0.00	0.00	0.00
HOP	0.32	0.41	0.37
PPO w/o foll.	**0.58**	0.63	0.61
Ours	0.57	**0.74**	**0.66**

Table 6: Success rates in simulation for *grasping with obstacle* task.

	DexYCB obstacle	3D-printed obstacle	Avg.
ViviDex	0.15	0.40	0.28
AdaDexGrasp	**0.55**	0.60	0.58
ManipTrans	0.00	0.00	0.00
HOP	0.25	0.40	0.33
PPO w/o foll.	0.50	0.55	0.61
Ours	**0.55**	**0.70**	**0.63**

Table 7: Success rates in real-world robots for *grasping with obstacle*, using xArm's built-in collision detector.

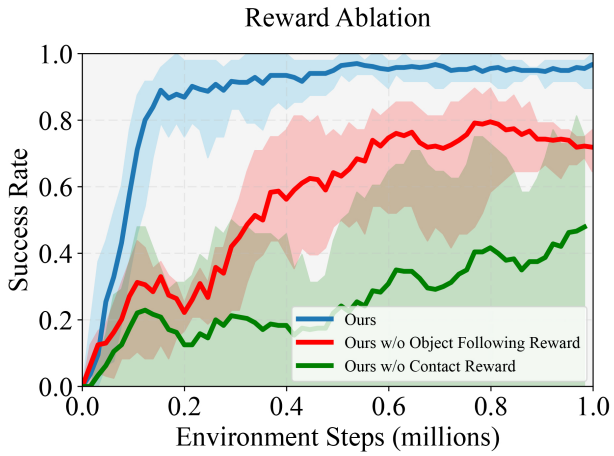

Figure 5: Learning curves for reward ablation (Solid lines: the mean over all seeds. Shaded areas: standard deviation).

average success rates in Fig. 5. We observe that when R_o is removed, sample efficiency decreases dramatically. When $R_{contact}$ is removed, training becomes unstable, resulting in lower success rates. These results demonstrate that the object following reward R_o is critical for ensuring stable and efficient policy learning.

Conclusion

We propose a hierarchical framework for learning dexterous manipulation from object-centric human motion priors to achieve generalizable manipulation, enabling zero-shot Sim-to-Real transfer. Our method outperforms existing baselines in YCB object grasping, articulated object manipulation, and grasping with obstacles in terms of success rates and generalizations to unseen objects. We demonstrate the importance of object following reward through ablation studies. We also validate the cross-embodiment potentials of our methods.

Acknowledgements

This work was partially supported by the NSF of China (No. 62425209). Besides, great thanks to Professor Hao Su for his support and useful advice in my research career.

References

Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; Ribas, R.; et al. 2019. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.

Bain, M.; and Sammut, C. 1995. A Framework for Behavioural Cloning. In *Machine intelligence 15*, 103–129.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. π0: A vision-language-action flow model for general robot control. *URL https://arxiv. org/abs/2410.24164*.

Calli, B.; Walsman, A.; Singh, A.; Srinivasa, S.; Abbeel, P.; and Dollar, A. M. 2015. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*.

Carpentier, J.; Saurel, G.; Buondonno, G.; Mirabel, J.; Lamiraux, F.; Stasse, O.; and Mansard, N. 2019. The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*.

Chao, Y.-W.; Yang, W.; Xiang, Y.; Molchanov, P.; Handa, A.; Tremblay, J.; Narang, Y. S.; Van Wyk, K.; Iqbal, U.; Birchfield, S.; et al. 2021. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9044–9053.

Chen, T.; Tippur, M.; Wu, S.; Kumar, V.; Adelson, E.; and Agrawal, P. 2023. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84): eadc9244.

Chen, Y.; Wang, C.; Yang, Y.; and Liu, C. K. 2024a. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*.

Chen, Z.; Chen, S.; Etienne, A.; Laptev, I.; and Schmid, C. 2024b. ViViDex: Learning Vision-based Dexterous Manipulation from Human Videos.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 02783649241273668.

Fan, Z.; Taheri, O.; Tzionas, D.; Kocabas, M.; Kaufmann, M.; Black, M. J.; and Hilliges, O. 2023. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12943–12954.

Gupta, A.; Eppner, C.; Levine, S.; and Abbeel, P. 2016. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3786–3793. IEEE.

Hong, Z.; Zheng, K.; and Chen, L. 2024. Fully Automatic Hand-Eye Calibration with Pretrained Image Models. *International Conference on Intelligent Robots and Systems (IROS)*.

Jin, W. 2024. Complementarity-free multi-contact modeling and optimization for dexterous manipulation. *arXiv preprint arXiv:2408.07855*.

Li, K.; Li, P.; Liu, T.; Li, Y.; and Huang, S. 2025. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 6991–7003.

Liu, X.; Adalibieke, J.; Han, Q.; Qin, Y.; and Yi, L. 2025. DexTrack: Towards Generalizable Neural Tracking Control for Dexterous Manipulation from Human References. *arXiv preprint arXiv:2502.09614*.

Liu, Y.; Liu, Y.; and Jiang, e., Che. 2022. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21013–21022.

Ma, Y.; Cramariuc, A.; Farshidian, F.; and Hutter, M. 2025. Learning coordinated badminton skills for legged manipulators. *Science Robotics*, 10(102): eadu3922.

Ma, Y. J.; Sodhani, S.; Jayaraman, D.; Bastani, O.; Kumar, V.; and Zhang, A. 2022. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. *arXiv preprint arXiv:2210.00030*.

Mendonca, R.; Bahl, S.; and Pathak, D. 2023. Structured World Models from Human Videos.

Mordatch, I.; Popović, Z.; and Todorov, E. 2012. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 137–144.

Nair, S.; Rajeswaran, A.; Kumar, V.; Finn, C.; and Gupta, A. 2022. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*.

Pan, C.; Junge, K.; and Hughes, J. 2024. Vision-language-action model and diffusion policy switching enables dexterous control of an anthropomorphic hand. *arXiv preprint arXiv:2410.14022*.

Pang, T.; Suh, H. T.; Yang, L.; and Tedrake, R. 2023. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on robotics*, 39(6): 4691–4711.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Qi, H.; Yi, B.; Suresh, S.; Lambeta, M.; Ma, Y.; Calandra, R.; and Malik, J. 2023. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, 2549–2564. PMLR.

Qin, Y.; Wu, Y.-H.; and Liu, e., Shaowei. 2022. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, 570–587. Springer.

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

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; and Levine, S. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*.

Romero, J.; Tzionas, D.; and Black, M. J. 2022. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*.

Rong, Y.; Shiratori, T.; and Joo, H. 2020. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*.

Rus, D. 1999. In-hand dexterous manipulation of piecewise-smooth 3-d objects. *The International Journal of Robotics Research*, 18(4): 355–381.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, L.; Liu, Y.; Zeng, L.; Ai, B.; Hong, Z.; and Su, H. 2025. Learning Adaptive Dexterous Grasping from Single Demonstrations. *arXiv preprint arXiv:2503.20208*.

Singh, H. G.; Loquercio, A.; Sferrazza, C.; Wu, J.; Qi, H.; Abbeel, P.; and Malik, J. 2024. Hand-object interaction pretraining from videos. *arXiv preprint arXiv:2409.08273*.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 23–30. IEEE.

Towers, M.; Kwiatkowski, A.; Terry, J.; Balis, J. U.; De Cola, G.; Deleu, T.; Goulão, M.; Kallinteris, A.; Krimmel, M.; KG, A.; et al. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032*.

Wan, W.; Geng, H.; Liu, Y.; Shan, Z.; Yang, Y.; Yi, L.; and Wang, H. 2023. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3891–3902.

Wang, C.; Shi, H.; Wang, W.; Zhang, R.; Fei-Fei, L.; and Liu, C. K. 2024a. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*.

Wang, J.; Yuan, Y.; Che, H.; Qi, H.; Ma, Y.; Malik, J.; and Wang, X. 2024b. Lessons from Learning to Spin" Pens". *arXiv preprint arXiv:2407.18902*.

Wen, B.; Yang, W.; Kautz, J.; and Birchfield, S. 2024. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17868–17879.

Wu, A.; Guo, M.; and Liu, C. K. 2022. Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization. *arXiv preprint arXiv:2207.00195*.

Xiang, F.; Qin, Y.; Mo, K.; Xia, Y.; Zhu, H.; Liu, F.; Liu, M.; Jiang, H.; Yuan, Y.; Wang, H.; et al. 2020. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11097–11107.

Ye, J.; Wang, K.; Yuan, C.; Yang, R.; Li, Y.; Zhu, J.; Qin, Y.; Zou, X.; and Wang, X. 2025. Dex1B: Learning with 1B Demonstrations for Dexterous Manipulation. *arXiv preprint arXiv:2506.17198*.

Yuan, Y.; Che, H.; Qin, Y.; Huang, B.; Yin, Z.-H.; Lee, K.-W.; Wu, Y.; Lim, S.-C.; and Wang, X. 2024. Robot synesthesia: In-hand manipulation with visuotactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 6558–6565. IEEE.

Zhao, S.; Zhu, X.; Chen, Y.; Li, C.; Zhang, X.; Ding, M.; and Tomizuka, M. 2024. DexH2R: Task-oriented Dexterous Manipulation from Human to Robots. *arXiv preprint arXiv:2411.04428*.

Zhong, Y.; Huang, X.; Li, R.; Zhang, C.; Liang, Y.; Yang, Y.; and Chen, Y. 2025. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. *arXiv preprint arXiv:2502.20900*.

Zhou, B.; Yuan, H.; Fu, Y.; and Lu, Z. 2024. Learning diverse bimanual dexterous manipulation skills from human demonstrations. *arXiv preprint arXiv:2410.02477*.