AsymDex: Asymmetry and Relative Coordinates for RL-based Bimanual Dexterity

Zhaodong Yang[1], **Yunhai Han**[1], **Ai-Ping Hu**[1], **Harish Ravichandar**[1]
[1]Georgia Institute of Technology
{halyang, yhan389, ahu6, harish.ravichandar}@gatech.edu

Abstract: We present *Asymmetric Dexterity (AsymDex)*, a novel and simple reinforcement learning (RL) framework that can efficiently learn a large class of bimanual skills in multi-fingered hands without relying on demonstrations. Two crucial insights enable AsymDex to reduce the observation and action space dimensions and improve sample efficiency. First, true ambidexterity is rare in humans and most of us exhibit strong "handedness". Inspired by this observation, we assign complementary roles to each hand: the *facilitating hand* repositions and reorients one object, while the *dominant hand* performs complex manipulations to achieve the desired result (e.g., opening a bottle cap, or pouring liquids). Second, controlling the *relative* motion between the hands is crucial for coordination and synchronization of the two hands. As such, we design relative observation and action spaces and leverage a relative-pose tracking controller. Further, we propose a two-phase decomposition in which AsymDex can be readily integrated with recent advances in grasp learning to facilitate both the acquisition and manipulation of objects using two hands. Unlike existing RL-based methods for bimanual dexterity with multi-fingered hands, which are either sample inefficient or tailored to a specific task, AsymDex can efficiently learn a wide variety of bimanual skills that exhibit asymmetry. Detailed experiments on seven asymmetric bimanual dexterous manipulation tasks (four simulated and three real-world) reveal that AsymDex consistently outperforms strong baselines that challenge our design choices. The project website is at https://sites.google.com/view/asymdex-2025/.

Keywords: Dexterous Bimanual Manipulation, Multi-Fingered Hands

1 Introduction

Bimanual dexterity is crucial for robots operating in human environments as they allow for complex yet flexible manipulation compared to a single hand [1–6]. We are interested in learning a wide range of bimanual dexterous skills on multi-fingered hands purely from reinforcement.

While learning on a single multi-fingered hand is known to be challenging [7–13], learning bimanual dexterity is made more challenging due to the higher-dimensionality and the need for coordination and synchronization of two hands [1, 14]. These challenges are only exacerbated when these skills have to be learned from reinforcement, explaining why existing efforts often either resort to expert demonstrations [15–18] or limit themselves to specific tasks [19, 20].

We rely on two insights to tackle the challenges of bimanual dexterity. First, we are inspired by how humans and other great apes approach these challenges: there is a natural *asymmetry* in how we use each of our hands when we perform most bimanual tasks [21]. Specifically, we tend to use one hand to reposition and reorient an object so as to make it easier for the other hand to perform complex manipulation. While leveraging such asymmetry might appear to restrict the class of bimanual skills we can learn, rich bodies of work in human biomechanics and evolution reveal its prevalence and necessity [21–24]. Evolutionary biologists posit handedness evolved to meet the escalating cognitive demands of tool use, language, and complex manipulation [25]. Indeed, a large number of real-world tasks admit such asymmetry (e.g., attachment, detachment, assembly, and pouring).

Looking closely at the asymmetry in bimanual dexterity reveals our second insight. Our non-dominant hand tends to hold an object firmly as we reorient and reposition it *relative* to the dominant hand or the object being held by the dominant hand (e.g., tilting a pen before uncapping). This sug-

RSS 2025 Workshop on Whole-Body Control and Bimanual Manipulation (RSS2025WCBM).

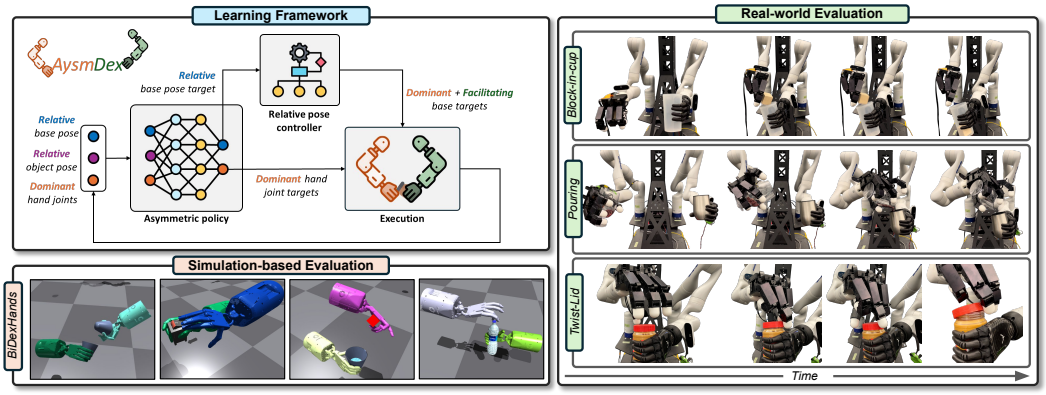


Figure 1: Our approach (AsymDex) efficiently learns asymmetric bimanual dexterous manipulation skills based on reinforcement learning by effectively leveraging i) the natural asymmetry in the hands' roles and ii) relative state and action spaces that prioritize synchronization.

gests that there is often little to no in-hand movement of the object grasped by the non-dominant hand, and robust synchronization can be achieved by ensuring relative movement of the two hands.

We contribute a novel RL-based learning framework for bimanual dexterity, dubbed *Asymmetric Dexterity (AsymDex)* by operationalizing the above two insights (see Fig. 3). To incorporate asymmetry, we define a *dominant hand* and a *facilitating hand*. While the dominant hand learns complex skills across all its degrees of freedom, the facilitating hand learns to reposition and reorient the object by controlling the 6D pose of its base (i.e., no finger movement). This allows us to both reduce the dimensionality of the observation and action spaces and tightly integrate the roles of the two hands. To ensure coordination and synchronization, AsymDex operates over relative observation and action spaces that incentivize flexible coordination of the two hands without resorting to explicit time-dependence or task-specific coordinate frame designs.

We also leverage the observation that bimanual manipulation in practice is composed of two distinct phases: i) the *acquisition phase* in which objects are grasped, and ii) the *interaction phase* in which the two hands coordinate to perform the bimanual task. Unlike many existing methods that entirely ignore the acquisition phase[2, 14, 19, 26], we show that this decomposition enables AysmDex to be seamlessly integrated with learned grasping policies to enable fluent execution.

In summary, we contribute AsymDex – a novel Rl-based framework for learning a wide variety of bimanual dexterous skills by taking inspiration from two key aspects of human bimanual dexterity: i) asymmetric hand roles, and ii) relative hand movement. We conduct comprehensive experiments on seven complex tasks (four simulated and three real) and compare against strong baselines that challenge the need for AsymDex's structural inductive biases. Our results show that AsymDex consistently outperforms these baselines in terms of both task performance and sample efficiency.

2 Related Work

Learning Bimanual Manipulation: Several existing methods focus on learning bimanual skills, but are often limited to simple end-effectors. Imitation learning (IL) based approaches have been particularly successful in bimanual manipulation [3, 26–28], and have led to novel and low-cost infrastructure to collect bimanual manipulation data [4, 29]. These approaches rely on demonstrations to provide the necessary supervision to learn effective coordination strategies. Reinforcement learning (RL) has also been shown to be successful in learning bimanual manipulation skills [30–33]. These methods implicitly incentivize coordination by learning to optimize reward functions that favor task success and efficiency. In contrast to all of these works that only consider parallel jaw grippers, AsymDex learns bimanual dexterous manipulation skills involving multi-fingered hands.

Asymmetry in Bimanual Manipulation: Motivated by the asymmetry in how humans use their two hands (referred to as *role-differentiated bimanual manipulation* [22–24, 34]), recent works assign different roles to each robot hand in the bimanual system [2, 35–38]. However, some of these approaches restrict the role of the facilitating hand to stabilizing the object while the dominant hand manipulates it [2, 35, 36]. In contrast, AsymDex allows the facilitating hand to reposition

and reorient the object *simultaneously* as the dominant hand executes its role. Importantly, unlike AsymDex, all these prior methods are limited to parallel jaw grippers.

Learning Dexterous Manipulation: Learning dexterous manipulation skills involves addressing numerous challenges due to high dimensional state and action spaces and highly nonlinear dynamics. Recent works have tackled these challenges using imitation learning (IL) or reinforcement learning (RL) and demonstrate impressive performance [7–11, 13, 39–42]. However, IL-based methods rely either on complex infrastructure and retargeting methods to collect demonstrations [7, 13, 40, 43, 44] or pre-trained expert policies [10, 11, 42]. On the other hand, RL-based methods do not share these constraints as they learn skills via reinforcement, but tend to require significant exploration even for dexterous manipulation with a single hand [8, 9, 39, 41]. As we show in our experiments, naive application of RL-based methods is not effective for bimanual dexterous manipulation due to the increased dimensionality and the need for coordination.

Learning Bimanual Dexterous Manipulation: A few recent studies have focused on learning bimanual dexterity. Some of these methods require the collection of expert demonstrations [5] and suffer from the same limitations we discussed earlier for IL-based methods that use parallel jaw grippers. To circumvent the need for collecting demonstrations, recent efforts have led to methods that only leverage RL and yet are capable of learning impressive bimanual manipulation skills, such as playing the piano [6], twisting lids off containers [19], and dynamic handover [20]. While these methods are specifically designed to solve a particular task, AsymDex is capable of efficiently learning different bimanual dexterous manipulation tasks. Some recent studies investigate generalized learning method by utilizing expert demonstration for efficient RL training [15–18], while AsymDex can efficiently learn bimanual manipulation skills without relying on demonstration.

3 Problem Formulation

We first formulate the general problem of bimanual dexterous manipulation, and then introduce asymmetric bimanual dexterity.

Consider the general problem of bimanual dexterous manipulation, in which two multi-fingered hands coordinate to manipulate up to two objects. Formally, this problem can be defined as a Partially-Observable Markov Decision Process (POMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{Z}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{S} \in \mathbb{R}^n$ is the state space, $\mathcal{Z} \in \mathbb{R}^m$ is the observation space, $\mathcal{A} \in \mathbb{R}^u$ is the action space, $\mathcal{R} : \mathbb{R}^m \times \mathbb{R}^u \to \mathbb{R}$ is the reward function, and $\mathcal{P} : \mathbb{R}^n \times \mathbb{R}^u \to \mathbb{R}^n$ is the environment dynamics. Note that we do not assume access to any demonstrations. Instead, we tackle of challenge of learning purely based on reinforcement. Given this formulation, the problem boils down to learning a policy $\pi : \mathcal{Z} \to \mathcal{A}$ that maximizes the expected discounted cumulative reward $E_\pi[\Sigma_{t=0}^{T-1} \gamma^t \mathcal{R}(z(t), a(t))]$.

Observation and Action Spaces: The observation space $\mathcal{Z}$ is composed of hand and object states. At time step t, $z(t) = [\xi_1(t), \xi_2(t), \xi^{obj}(t)]$, where $\xi_1(t)$ contains the first hand's current full (fingers + wrist) configuration $\xi_1^h(t) \in \mathbb{R}^{n_1}$ and the 6 DOF pose of its base $\xi_1^b(t) \in \mathbb{SE}(3)$, $\xi_2(t)$ contains the corresponding elements for the second hand, and $\xi^{obj}(t)$ contains the 6 DOF poses of either two objects (e.g., stacking two cups) or parts of one object (e.g., bottle and lid). The joint action at time step t is given by $a(t) = [\hat{\xi}_1(t), \hat{\xi}_2(t)]$, where $\hat{\xi}_1(t)$ denotes the target joint configuration of the first hand $\hat{\xi}_1^h(t)$ and the target 6 DOF pose of its base $\hat{\xi}_1^b(t)$, and $\hat{\xi}_2(t)$ denotes the corresponding targets for the second hand. With the above definitions, we can now define the problem of learning bimanual dexterity as one of learning a monolithic symmetric policy: $\pi_{\text{sym}}(a(t)|z(t))$.

Asymmetric Dexterity Problem: Asymmetric dexterity can be viewed as a broad subclass to the above general class of problems. Inspired by strategies employed by humans and other great apes, we are interested in tasks in which one multi-fingered hand performs complex and precise manipulations while the other plays a facilitating role by supporting and actively reorienting objects of interest (e.g., stacking, attachment, detachment, etc.). Note that our formulation does not restrict the movement of the second hand; it merely restricts the relative movement between the second hand and the object being held. As explained in Sec. 1, a large number of bimanual tasks exhibit asymmetry, hinting at handedness in most humans. We are thus interested in learning an asymmetric policy $\pi_{\text{AsymDex}}(\cdot)$ with carefully-defined observation $\mathcal{Z}_{\text{AsymDex}}$ and action $\mathcal{A}_{\text{AsymDex}}$ spaces in an effort to improve both effectiveness and sample efficiency.

Note that our primary contributions and the asymmetric assumption pertain to the *interaction phase* of bimanual dexterity, in which the two hands actively coordinate to complete the task after having grasped the necessary object(s). Most existing works focus solely on the interaction phase [19, 20, 45]. In Sec. 4.4, we discuss how our approach can be readily extended to also tackle the *acquisition phase* (learning to grasp the necessary objects before coordinating).

4 AsymDex: Learning Asymmetric Dexterity

While one could learn the symmetric policy $\pi_{sym}(\cdot)$ as defined in Sec. 3, training such a policy can be inefficient or ineffective due to the high dimensionality of the observation and action spaces (see Sec. 5 for empirical evidence). Importantly, a symmetric approach ignores the natural asymmetry found in most bimanual tasks. Below, we explain how AsymDex overcomes these challenges.

4.1 Incorporating Asymmetry

Motivated by the natural asymmetry in human bimanual manipulation [22–24], we assign different roles to each robot hand during their interaction: a *facilitating hand* that is responsible for holding and repositioning and reorienting the object, and a *dominant hand* which is responsible for fine-grained dexterous manipulation of the object(s). We note that there tends to be no relative motion between the facilitating hand and the grasped object in asymmetric dexterity since the facilitating hand need only hold, move, and reorient the object (i.e., no in-hand reorientation). In contrast, the dominant hand can interact freely with the object(s). This suggests that the asymmetric dexterity is neither dependent on nor influences the finger joints of the facilitating hand during the interaction phase. As such, we can considerably reduce the observation and action spaces by defining an asymmetry-only bimanual policy: $\pi_{\text{asym}}(a_{\text{asym}}(t)|z_{\text{asym}}(t))$ with actions $a_{\text{asym}}(t) = [\hat{\xi}_d(t), \hat{\xi}^b_f(t)]$ and observations $z_{\text{asym}}(t) = [\xi_d(t), \xi^b_f(t), \xi^{obj}(t)]$. This change both reduces the dimensionality and ensures that the facilitating hand doesn't learn unproductive or unnecessary behaviors.

4.2 Incorporating Relative Observation and Action Spaces

In addition to asymmetry, a key characteristic of bimanual dexterity is the synchronized and responsive movement of the two hands. We can further reduce the size of the observation and action spaces and introduce tight coupling between the hands' behaviors by defining *relative* and *object-centric* coordinates that capture the relationships between the movements of two hands and the object(s) being manipulated. Indeed, the use of relative state spaces has shown to considerably benefit bimanual manipulation with simple end effectors [26, 46–48]. While some of these prior works limit the relative space to a one-degree-of-freedom (1-DoF) action space [26], AsymDex allows for complete 6-DoFs relative space.

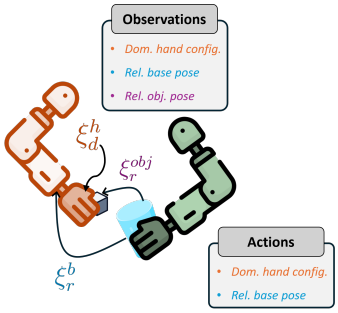


Figure 2: AsymDex's observation and Action Spaces.

Let $\xi^{obj}_f(t)$ be the state of the object being held by the facilitating hand, and let $\xi^{obj}_d(t)$ be the state of the object being manipulated by the dominant hand. We attach a coordinate frame to the object being held by the facilitating hand: P_f. Now, we can transform the asymmetry-only observations $z_{\text{asym}}(t)$ (originally defined in the absolute or world coordinate frame) into the new coordinate frame P_f. Note that since there is no relative motion between the facilitating hand and the object that it is holding, neither $\xi^{obj}_f(t)$ nor $\xi^b_f(t)$ change in P_f, and can thus be removed from our observation space without losing any information. Now, transforming the observations $(\xi^b_d(t), \xi^{obj}_d(t))$ onto P_f yields $z_{\text{AsymDex}} = [\xi^h_d(t), \xi^b_r(t), \xi^{obj}_r(t)]$, where $\xi^b_r(t)$ and $\xi^{obj}_r(t)$ respective denote the 6D relative poses of the dominant hand base and the object being manipulated by the dominant hand, both defined with respect to the object being held by the facilitating hand. Note that since $\xi^h_d(t)$ denotes the dominant hands' joint states, it is not impacted by the change of coordinates. Similarly, we apply the same modifications to the asymmetry-only action $a_{asym}(t)$, yielding $a_{\text{AsymDex}}(t) = (\hat{\xi}^b_r(t), \hat{\xi}^h_d(t))$, where $\hat{\xi}^b_r(t)$ is the target relative pose of the dominant hand now defined relative to the object being held by the facilitating hand. Incorporating the above change of coordinates in addition to leveraging asymmetry, allows us to define AsymDex's policy as $\pi_{\text{AsymDex}}(a_{\text{AsymDex}}(t)|z_{\text{AsymDex}}(t))$. Note

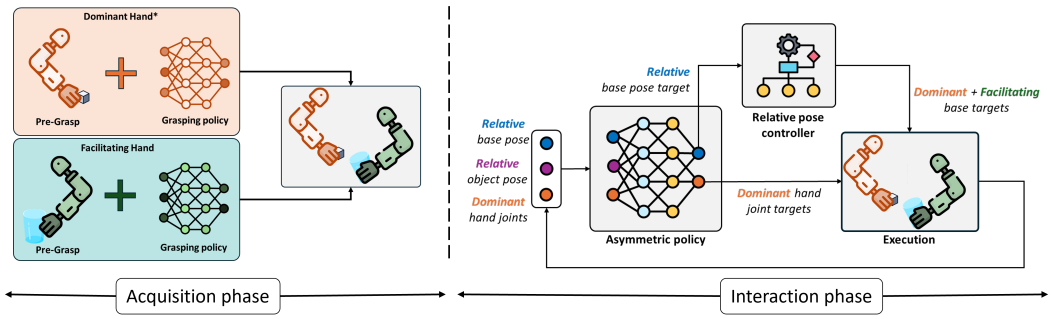


Figure 3: We decompose asymmetric bimanual dexterous manipulation into two phases: An *acquisition* phase and an *interaction* phase. We show that AsymDex can be readily integrated with learned grasping policies in order to seamlessly acquire and manipulate objects.

that our formulation has significantly reduced the dimensions of both the state and action spaces, compared to the symmetric policy π_{sym} as defined in Section 3.

We parameterize the AsymDex policy $\pi_{\text{AsymDex}}(\cdot)$ using an MLP and using Proximal Policy Optimization (PPO) [49] to train it. See Appendix. A for details of the algorithm and policy architecture.

4.3 Relative Pose Tracking Controller

To control the hand bases based on the target relative pose $\hat{\xi}_r^b(t)$ provided by π_{AsymDex}, we designed a bimanual controller that computes both the target dominant hand base pose $\hat{\xi}_d^b(t)$ and the target facilitating hand base pose $\hat{\xi}_f^b(t)$ as follows

$$\begin{aligned} \hat{\xi}_d^b(t) &= \alpha R_{world}^{o_f} \cdot dist(\hat{\xi}_r^b(t), \xi_r^b(t)) + \xi_d^b(t), \\ \hat{\xi}_f^b(t) &= (\alpha - 1) R_{world}^{o_f} \cdot dist(\hat{\xi}_r^b(t), \xi_r^b(t)) + \xi_f^b(t), \end{aligned} \tag{1}$$

where $R_{world}^{o_f}$ denotes the rotational transformation from Frame P_f to the world frame P_W, $dist(\cdot)$ denotes the difference between two 6D poses, and α is a hyperparameter that controls the relative involvement of each hand. The pseudo-code of the training process is included in Alg. 1.

4.4 Acquisition Phase

While our approach as explained thus far deals with the challenge of coordinating two hands to accomplish asymmetric dexterous manipulation tasks, it assumes that the object(s) of interest have already been grasped at the beginning of the task. However, in practice, robots must be learn to grasp and pick up the necessary objects before the interaction between the two hands (and the objects) can begin. We refer to this initial phase as the *acquisition* phase. Most recent works on bimanual dexterous manipulation often entirely ignore the acquisition phase and focus purely on the interaction phase [6, 19, 20]. In contrast, we demonstrate that our approach can seamlessly accommodate the acquisition phase by i) leveraging the observation that the acquisition phase doesn't require the coordination of two arms, and ii) employing recent advances in learning to grasp. Specifically, we demonstrate that we can seamlessly integrate AsymDex with PDGM [50], which can efficiently learn multi-fingered grasping policies by leveraging pre-grasp poses (see Fig. 3). Details about the grasping reward design are available in Appendix. B. We begin by executing the grasping policy in isolation and then "turn on" the asymmetric policy learned by AsymDex after the object has been firmly grasped by the facilitating hand. If the task requires the dominant hand to also grasp a second object, we employ the same method to train a grasping policy for dominant hand to acquire the object, but switch the control of the dominant hand's joints over the asymmetric policy after the object has been grasped.

5 Experimental evaluation

We evaluated AsymDex on four simulated and three real-world bimanual dexterous tasks and compared its performance against strong baselines that challenge our key design choices.

5.1 Simulation Experiments

Our experiments in simulation both systematically and rigorously evaluate AsymDex.

Tasks: We evaluated AsymDex and the baselines on the following four bimanual manipulation tasks which contain both original (*Block in cup*, *Bottle cap*) and adapted tasks (*Stack*, *Switch*) from BiDexHand [14] (see Fig. 1). All these tasks use two Shadow Hands – each a 30-DoF simulated multi-finger hand system (24-DoF hand + 6-DoF floating wrist base) built with Isaac Gym [51].

- *Block in cup*: The two hands must coordinate to ensure that one hand places a block inside a cup that is being held by the other without letting either the cup or the block fall to the ground.
- *Stack*: Two cups need to be stacked together. Each hand must hold a cup, and both must coordinate such that the two cups are aligned as one slides into the other.
- *Bottle cap*: One hand must hold and reorient a bottle such that the other hand can grasp and separate the bottle cap from the bottle.
- *Switch*: One hand holds and reorients a switch in a way that allows the other hand to turn it on.

Note that our task designs are more challenging than their counterparts in BiDexHand [14]. We require that the two hands coordinate and synchronize to achieve success in each of the above four tasks, especially since (unlike the original designs) we do not provide a support surface (e.g., a table) that would significantly reduce the need for bimanual coordination. See Appendix. C for details on state space design, sampling procedure, success criteria, and reward design.

Metrics: We quantify performance in terms of i) *success rate* (see Appendix. C for criteria) and ii) *sample efficiency*. We report both metrics across five random seeds in all experiments.

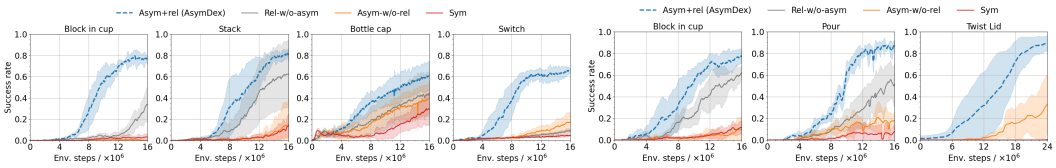
5.1.1 Learning Bimanual Coordination

We first evaluated AsymDex's effectiveness during the interaction phase. Following contemporary practice in methods that learn bimanual dexterous skills [19, 20], we initialized the environment such that the hands are at appropriate pre-grasp poses near the respective objects. This allows us to isolate and examine AsymDex's ability to learn to coordinate two multi-fingered hands. See Section 5.1.2 for the second experiment in which we also consider the challenge of acquiring the objects from a tabletop surface before interaction begins.

We compared AsymDex against the following baselines:

- `Sym`: This policy assumes that both hands play an equal role in bimanual manipulation (see π_{sym} in Sec. 3). This baseline allows us to examine the necessity and effectiveness of leveraging the asymmetry in hand roles as well as the relative action and observation spaces.
- `Asym-w/o-rel`: This policy leverages asymmetry in hand roles, but learns over absolute observation and action places (see Sec. 4.1). As such, this baseline allows us to examine the necessity and effectiveness of relative action and observation spaces.
- `Rel-w/o-asym`: This policy leverages the relative observation and action places, but ignores asymmetry. As such, it allows us to examine the necessity and effectiveness of asymmetry.

Both asymmetry and relative spaces are necessary for consistent performance: We report the learning curves in Fig. 4 (a) and success rates in Table 1. Note that AsymDex consistently outperforms all the baselines across all four tasks in terms of success rate and sample efficiency, with significant margins in two tasks (*Block in cup* and *Switch*). `Rel-w/o-asym` performs better than the other two baselines across all tasks except *Switch*, while `Asym-w/o-rel` performs better than `Monolithic` on all tasks except *Block in cup*, on which both struggle.



(a) BiDexHand task training curve

(b) Real-world task training curve

Figure 4: AsymDex consistently outperforms the baselines in terms of sample efficiency and success rate. Solid lines indicate mean trends and shaded areas show $\pm$ std. dev., over five random seeds.

Task \ Method	Sym	Asym-w/o-rel	Rel-w/o-Asym	AsymDex (ours)
Block in cup	0.0429 ± 0.0266	0.0164 ± 0.0190	0.1086 ± 0.1378	$\mathbf{0.7701 \pm 0.0559}$
Stack	0.0771 ± 0.0611	0.2185 ± 0.3232	0.6560 ± 0.3213	$\mathbf{0.8392 \pm 0.0596}$
Bottle cap	0.3111 ± 0.1813	0.4143 ± 0.1410	0.4730 ± 0.2011	$\mathbf{0.6295 \pm 0.1422}$
Switch	0.0563 ± 0.0126	0.1626 ± 0.0882	0.1149 ± 0.0176	$\mathbf{0.6700 \pm 0.0359}$

Table 1: Success rates (mean $\pm$ std. dev.) for the interaction phase.

Taken together, the above observations reveal a few key insights. First, when used in isolation, neither asymmetry nor relative spaces are sufficient across all tasks. Second, the use of relative spaces offers a larger boost in performance compared to asymmetry, likely due to the fact that relative spaces avoid unnecessary exploration (e.g., when the two hands move in parallel) while allowing the facilitating hand to exhibit more complex behaviors. Third, ignoring both asymmetry and relative spaces (`Sym`) hardly leads to success.

5.1.2 Learning to Grasp and Coordinate

We next evaluated AsymDex's ability to incorporate the object acquisition phase before the interaction phase. Specifically, we initialize the environment for each task such that the objects of interest are placed on a tabletop surface. As such, each method needs to learn both to grasp the necessary objects and to coordinate the two hands to complete the tasks.

For AsymDex, we follow the same strategy introduced in Section 4.4, and comapare its performance against the following baselines:

- `Monolithic`: This baseline uses a single policy to learn both the grasping and interaction phases for both hands, allowing us to investigate the benefits of two-phase decomposition.
- `2-stage-sym`: This policy benefits from the two phase decomposition but leverages neither asymmetry nor relative spaces. As such, this baseline allows us to examine if merely employing two-phase decomposition is sufficient.

To ensure a fair comparison, we provide pre-grasp pose annotations to both baselines. Further, we ensure that the total number of env. interactions (the number of one stage or the sum of two stages) is the same across AsymDex and the baselines. See Appendix. B for details of the grasping learning.

AsymDex can effectively combine the acquisition and interaction phases: We report the overall roll-out success rate of all methods for two tasks across five random seeds in Table. 2. We find that AsymDex significantly outperforms the other two baselines in both tasks, suggesting that combining phase decomposition with AsymDex's other two design choices (asymmetry and relative spaces) results in policies that can effectively handle both the acquisition and the interaction phases of bimanual dexterous manipulation. The fact that `2-stage-sym` baseline outperforms the `Monolithic` baseline points to the inherent benefits of phase decomposition. Our qualitative analysis of *Block in cup* task revealed that `Monolithic` policy learns to tip the cup over and push the block towards the cup. In contrast, both the two-stage policies learn more intuitive behaviors, suggesting that the phase decomposition nudges the grasping and interaction policies to learn reasonable behaviors that complement each other.

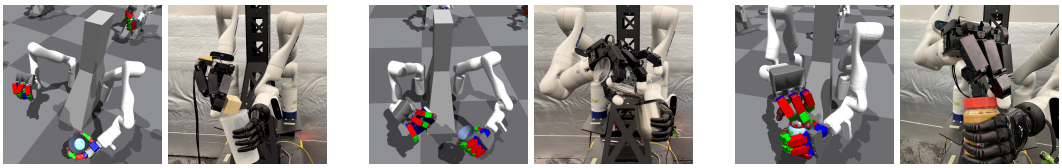
5.2 Real-world Experiments

We finally evaluated AsymDex and the same baselines on the following 3 real-world bimanual manipulation tasks inspired by recent bimanual manipulation works [5, 19, 26, 52–55].

Hardware setup: We use one 16-dof Allegro hand from Wonik Robotics and one 6-DoF Ability Hand from Psyonic. They are mounted on two 7-DoF Kinova Gen3 robotic arms. We used the Ability Hand for one of the hands since we only had access to one physical Allegro hand. For

Table 2: Success rates (mean $\pm$ std. dev.) after combining acquisition and interaction phases

Task \ Method	Monolithic	2-stage-sym	AsymDex
Block in cup	0.0321 ± 0.0251	0.1505 ± 0.1059	$\mathbf{0.7938 \pm 0.0897}$
Bottle cap	0.0 ± 0.0	0.2552 ± 0.1573	$\mathbf{0.6116 \pm 0.1328}$



(a) Block in cup (b) Pour (c) Twist lid

Figure 5: We created simulation environments to match our hardware setup.

object pose information, we either employ an Realsense camera with AprilTag-based object tracking method or estimate it directly from the end-effector's pose, especially during occlusion.

Real-world Tasks: i) *Block in Cup*: Place a block with sides of 5 cm inside a cylindrical cup with an internal diameter of 10 cm, ii) *Pour*: Pour dry beans from one cup into another, and iii) *Twist Lid*: Twist the lid off a jar while holding the jar. Snapshots are shown in Fig. 5.

Simulation Evaluation: For each task, we first construct a simulated counterpart of our hardware setup and then train all policies in simulation. We then compared their performance in terms of *success rate* and *sample efficiency*, as reported in Fig. 4 (b). As observed in previous experiments, AsymDex consistently and significantly outperformed all baselines in terms of both metrics.

Sim2Real Transfer: We deployed only the AsymDex policies on hardware since the baseline policies tended to exhibit either negligible success rates or aggressive behaviors that could damage the hardware. To enable effective sim-to-real transfer, we apply domain randomization during reinforcement learning, and details of the randomization process are provided in Appendix. D.

AsymDex enables zero-shot Sim2Real Transfer: As shown in Table. 3, AsymDex achieves high success rates across all deployments. This once again highlights the effectiveness of AsymDex in learning robust and reliable real-world bimanual dexterous manipulation skills. For the *block in cup* and *pour* task, AsymDex is simply trained with observation noise component of common domain randomization technique to learn a policy that can be deployed in real world successfully. For the most challenging *twist lid* task, we added observation noise and action noise, as well as other randomization components for sim2real transfer. Though we utilized similar reward design as previous work [19] for this task, we observed that AsymDex learns a more natural and human-like behavior compared to the previous work. Besides, unlike previous work which fixes the hand base motion [2, 19], AsymDex is able to reposition and reorient both hand bases while two hands interact, enabling more fluent bimanual coordination. See Appendix. D for the detailed reward design and simulation success criterion for each task.

Block in cup	Pouring	Twist Lid
16/20	17/20	18/20

Table 3: Real-world task success rates for AsymDex.

6 Conclusion

Our framework (AsymDex) is capable of learning complex asymmetric bimanual dexterous manipulation tasks via reinforcement without relying on demonstrations. We introduced and validated the need for AsymDex's two crucial ingredients: assigning asymmetric roles to the two hands, and using relative observation and action spaces. Our evaluation results reveal that the combination of these choices consistently leads to better sample efficiency and success rates across different tasks.

7 Limitations and Future Work

Our work has revealed a number of limitations and avenues for future research. First, AsymDex in its current form cannot handle certain bimanual tasks that require complex multi-finger manipulation from both hands (e.g., reorienting a heavy object, dynamic handover). Second, AsymDex does not consider the kinodynamic constraints that might result from manipulator arms. Three, behaviors produced by AsymDex are not always natural or human-like due to lack of necessary incentives.

References

[1] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous systems*, 60(10):1340–1353, 2012.

[2] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Conference on Robot Learning*, pages 563–576. PMLR, 2023.

[3] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2022.

[4] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

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

[6] K. Zakka, P. Wu, L. Smith, N. Gileadi, T. Howell, X. B. Peng, S. Singh, Y. Tassa, P. Florence, A. Zeng, et al. Robopianist: Dexterous piano playing with deep reinforcement learning. *arXiv preprint arXiv:2304.04150*, 2023.

[7] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[8] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning Dexterous In-Hand Manipulation. *International Journal of Robotics Research (IJRR)*, 2020.

[9] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik. In-hand object rotation via rapid motor adaptation. *arXiv preprint arXiv:2210.04887*, 2022.

[10] Y. Han, M. Xie, Y. Zhao, and H. Ravichandar. On the utility of koopman operator theory in learning dexterous manipulation skills. In *Conference on Robot Learning*, pages 106–126. PMLR, 2023.

[11] Y. Han, Z. Chen, K. A. Williams, and H. Ravichandar. Learning prehensile dexterity by imitating and emulating state-only observations. *IEEE Robotics and Automation Letters*, 9(10): 8266–8273, 2024.

[12] H. Chen, A. ABUDUWEILI, A. Agrawal, Y. Han, H. Ravichandar, C. Liu, and J. Ichnowski. Korol: Learning visualizable object feature with koopman operator rollout for manipulation. In *8th Annual Conference on Robot Learning*.

[13] K. Shaw, S. Bahl, A. Sivakumar, A. Kannan, and D. Pathak. Learning dexterity from human hand motion in internet videos. *The International Journal of Robotics Research*, page 02783649241227559, 2024.

[14] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, Z. Lu, S. McAleer, H. Dong, S.-C. Zhu, and Y. Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5150–5163, 2022.

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

[16] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, L. Paulsen, G. Yang, S. Yi, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[17] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

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

[19] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.

[20] B. Huang, Y. Chen, T. Wang, Y. Qin, Y. Yang, N. Atanasov, and X. Wang. Dynamic handover: Throw and catch with bimanual hands. *arXiv preprint arXiv:2309.05655*, 2023.

[21] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, 1987.

[22] M. Kimmerle, C. L. Ferre, K. A. Kotwica, and G. F. Michel. Development of role-differentiated bimanual manipulation during the infant's first year. *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, 52(2):168–180, 2010.

[23] R. L. Sainburg. Evidence for a dynamic-dominance hypothesis of handedness. *Experimental brain research*, 142:241–258, 2002.

[24] B. E. Studenka and H. N. Zelaznik. The influence of dominant versus non-dominant hand on event and emergent motor timing. *Human Movement Science*, 27(1):29–52, 2008.

[25] L. Cashmore, N. Uomini, and A. Chapelain. The evolution of handedness in humans and great apes: a review and current issues. *Journal of anthropological sciences*, 86(2008):7–35, 2008.

[26] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. *arXiv preprint arXiv:2405.03666*, 2024.

[27] G. Franzese, L. de Souza Rosa, T. Verburg, L. Peternel, and J. Kober. Interactive imitation learning of bimanual movement primitives. *IEEE/ASME Transactions on Mechatronics*, pages 1–13, 2023.

[28] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[29] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[30] Y. Lin, A. Church, M. Yang, H. Li, J. Lloyd, D. Zhang, and N. F. Lepora. Bi-touch: Bimanual tactile manipulation with sim-to-real deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2023.

[31] S. Kataoka, S. K. S. Ghasemipour, D. Freeman, and I. Mordatch. Bi-manual manipulation and attachment via sim-to-real reinforcement learning. *arXiv preprint arXiv:2203.08277*, 2022.

[32] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Efficient bimanual manipulation using learned task schemas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1149–1155. IEEE, 2020.

[33] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu. Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3867–3874. IEEE, 2023.

[34] H. Jo, W. Choi, G. Lee, W. Park, and J. Kim. Analysis of visuo motor control between dominant hand and non-dominant hand for effective human-robot collaboration. *Sensors*, 20(21):6368, 2020.

[35] R. Holladay, T. Lozano-Pérez, and A. Rodriguez. Robust planning for multi-stage forceful manipulation. *The International Journal of Robotics Research*, 43(3):330–353, 2024.

[36] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh. Learning bimanual scooping policies for food acquisition. *arXiv preprint arXiv:2211.14652*, 2022.

[37] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen. Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166, 2022.

[38] Y. Cui, Z. Xu, L. Zhong, P. Xu, Y. Shen, and Q. Tang. A task-adaptive deep reinforcement learning framework for dual-arm robot manipulation. *IEEE Transactions on Automation Science and Engineering*, 2024.

[39] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.

[40] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022.

[41] G. Khandate, S. Shang, E. T. Chang, T. L. Saidi, J. Adams, and M. Ciocarlie. Sampling-based Exploration for Reinforcement Learning of Dexterous Manipulation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.020.

[42] M. Xie, A. Handa, S. Tyree, D. Fox, H. Ravichandar, N. D. Ratliff, and K. Van Wyk. Neural geometric fabrics: Efficiently learning high-dimensional policies from demonstration. In *Conference on Robot Learning*, pages 1355–1367. PMLR, 2023.

[43] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[44] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023.

[45] T. Chen, E. Cousineau, N. Kuppuswamy, and P. Agrawal. Vegetable peeling: A case study in constrained dexterous manipulation. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[46] R. Laha, J. Vorndamme, L. F. Figueredo, Z. Qu, A. Swikir, C. Jähne, and S. Haddadin. Coordinated motion generation and object placement: A reactive planning and landing approach. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9401–9407. IEEE, 2021.

[47] P. Chiacchio, S. Chiaverini, and B. Siciliano. Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. 1996.

[48] S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, and D. Sallé. Dual-arm relative tasks performance using sparse kinematic control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6003–6009. IEEE, 2018.

[49] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[50] S. Dasari, A. Gupta, and V. Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3889–3896. IEEE, 2023.

[51] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox. Gpu-accelerated robotic simulation for distributed reinforcement learning, 2018.

[52] A. . Team, J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, K. Draper, D. Dwibedi, C. Finn, P. Florence, S. Goodrich, W. Gramlich, T. Hage, A. Herzog, J. Hoech, T. Nguyen, I. Storz, B. Tabanpour, L. Takayama, J. Tompson, A. Wahid, T. Wahrburg, S. Xu, S. Yaroshenko, K. Zakka, and T. Z. Zhao. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation, 2024. URL https://arxiv.org/abs/2405.02292.

[53] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang. Open-television: Teleoperation with immersive active visual feedback. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=Yce2jeILGt.

[54] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning, 2024. URL https://arxiv.org/abs/2407.03162.

[55] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=55tYfHvanf.

Appendices

A Training Details

We use Proximal Policy Optimization (PPO) [49] algorithm to train all policies $\pi_{\text{naive}}, \pi_{\text{asym}}$, and π_{AsymDex} with their corresponding value functions. Both policies and value functions are parameterized via a three-layer MLP network. The size of hidden layers for each is i) policy: (256, 256, 128), ii) value function: (512, 512, 512). The activation functions are all set as Exponential Linear Unit (ELU). We use the same PPO hyperparameters for all the baselines and AsymDex ($\gamma : 0.98, \lambda : 0.95$, clip range: 0.2, minibatch size: 8092). We use an adaptive learning rate with KL threshold of 0.016. We train the polices on a computer with a single Nividia RTX 4090 GPU.

Algorithm 1: AsymDex: Interaction Phase

Randomly initialize the two hand bases' poses $\xi_f^b(0)$ and $\xi_d^b(0)$, and initialize object poses $o_f(0)$ and $o_d(0)$ based on $\xi_f^b(0)$ and $\xi_d^b(0)$. Initialize policy π_θ.
for $iter \in \{1, ..., \max\}$ **do**
  Initialize replay buffer $\mathcal{B} = \varnothing$;
  for $t \in \{1, ..., M\}$ **do**
    Simulate:
    Collect hand and object states $\xi_f^b(t)$, $\xi_d^b(t)$, $\xi_d^h(t)$, $o_f(t)$, $o_d(t)$;
    Compute relative states $\xi_r^b(t) = \xi_a^b(t) \ominus P_f$, $o_r(t) = o_d(t) \ominus P_f$;
    Policy $\pi_{\text{AsymDex}}(\hat{\xi}_r^b(t), \hat{\xi}_d^h(t) | \xi_r^b(t), \xi_d^h(t), o_r(t))$ outputs relative actions;
    Bimanual controller (Eqn. 1) computes $\hat{\xi}_f^b(t)$ and $\hat{\xi}_d^b(t)$ based on $\hat{\xi}_r^b(t)$;
    if *Meet reset condition* **then**
      Reset environment;
    end
    Environment physics steps with $\hat{\xi}_f^b(t), \hat{\xi}_d^b(t), \hat{\xi}_d^h(t)$;
    Evaluate:
    Compute reward $r(t)$
    Collect observations $(\xi_r^b(t), \xi_r^h(t), o_r(t))$, actions $(\hat{\xi}_r^b(t), \hat{\xi}_r^h(t))$, and reward $r(t)$ into buffer $\mathcal{B}$;
  end
  Update the Policy π_{AsymDex} based on $\mathcal{B}$;
end
Return: Trained policy π_{AsymDex}

B Grasping Learning

Two-stage policy Both AsymDex (our approach) and `2-stage-sym` policy (one of the baselines in Sec. 5.1.2) are two-stage policies. Therefore, they can first learn a grasping policy for the facilitating hand (or two grasping policies for facilitating hand and dominant hand respectively). Such policy $\pi_{\text{grasp}}(\hat{\xi}_f^h(t) | \xi_f^h(t), \xi_f^b(t) \ominus P_f)$ takes in the hand joint states and the relative pose between hand and the object, and outputs the target hand joint positions to grasp the object firmly. We first provide pre-grasp annotations [50], which allows the hands to initialize at the position close to the objects with proper joint positions. Then we script the $6D$ lifting hand base motions and design the the following rewards, which is the same across all objects.

$$Reward = R_{rel\ pos} + R_{rel\ rot}$$

The relative position reward $R_{rel\ pos} = (\alpha - ||x_{obj} - x_{initial}||) * \beta$, where x_{obj} is the current relative position between the object and the hand, and $x_{initial}$ is the initial relative position between the object and the hand. The $\alpha, \beta \in \mathcal{R}_+$ are hyper-parameters. The relative rotation reward $R_{rel\ pos} =< u_{obj}, u_{hand} >$, where u_{obj} is the object direction vector, u_{hand} is the hand direction vector, and $< \cdot, \cdot >$ denotes the inner product of two vectors. We define the object direction vector and hand direction vector to be the same at the beginning of the grasping phase. Both rewards encourage the hand to keep a constant relative pose, i.e., grasping the object, during the script motion.

One-stage policy Another baseline in Sec. 5.1.2, i.e., the `monolithic` policy, does not incorporate the task decomposition. Therefore, it only uses the task-specific interaction rewards (see Appendix. C) to learn how to complete the entire bimanual task. For a fair comparison, both hands also start at the pre-grasp poses.

C BiDexHands Task Design

In this section, we show the details for each BiDexHands simulation task.

State Space Design For each task, the hand joint states $\xi_f^h(t)$, $\xi_d^h(t)$ include the 24-DoF hand joint positions and the 24-DoF hand joint velocities. We use quaternions to represent rotation part of object and hand base poses. And for all policies, we also include the previous actions in the policy input. For the *block in cup* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of the cup and the block respectively. For the *stack* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of two cups. For the *Bottle cap* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of the bottle and the cap respectively. For the *Switch* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of the switch body and the button respectively. The dimensions of the observation and action spaces of each policy are shown in Table. 4. It is obvious that AsymDex policy significantly reduces the state dimensions.

Table 4: Dimension of Observation and action spaces. For all tasks, the dimensions are identical.

	`Sym`	`Asym-w/o-rel`	`Rel-w/o-Asym`	`AsymDex` (ours)
Observation	176	108	163	88
Action	52	32	46	26

Sampling Procedure

- *Block in cup*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.3, 0.7)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.2, 0.0)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.7, 1.1)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.57, 1.57)$, in radians. The block is initialized in the dominant hand. Thus its position and rotation is calculated based on the initial position and rotation of dominant hand base. The initial position of the facilitating hand base is at $[0.55, 0.6, 0.8]$.
- *Stack*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.3, 0.7)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.2, 0.0)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.7, 1.1)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.57, 1.57)$, in radians. The cup is initialized in the dominant hand. Thus its position and rotation is calculated based on the initial position and rotation of dominant hand. The initial position of the facilitating hand base is at $[0.55, 0.6, 0.8]$.
- *Bottle cap*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.58, 0.62)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.21, -0.19)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.58, 0.62)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.0, 1.0)$, in radians. The initial position of the facilitating hand base is randomized: $x_f \in \mathcal{X} \sim \mathcal{U}(0.53, 0.57)$, $y_f \in \mathcal{Y} \sim \mathcal{U}(0.59, 0.61)$, $z_f \in \mathcal{Z} \sim \mathcal{U}(0.43, 0.45)$. For the rotation of the facilitating hand, we randomly rotate it around the axis along the arm at a random angle, $\beta \in \mathcal{B} \sim \mathcal{U}(-0.5, 0.5)$, in radians. The bottle is initialized in the facilitating hand. Thus its position and rotation is calculated based on the initial position and rotation of facilitating hand base.
- *Switch*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.2, 0.6)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.25, -0.05)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.5, 0.9)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.0, 1.0)$, in radians. The initial position of the facilitating hand base is randomized: $x_f \in \mathcal{X} \sim \mathcal{U}(0.2, 0.6)$, $y_f \in \mathcal{Y} \sim \mathcal{U}(0.05, 0.25)$, $z_f \in \mathcal{Z} \sim \mathcal{U}(0.41, 0.81)$. For the rotation of the facilitating hand, we randomly rotate it around the axis along the arm at a random angle, $\beta \in \mathcal{B} \sim \mathcal{U}(-1.0, 1.0)$, in radians. The switch is initialized in the facilitating hand. Thus its position and rotation is calculated based on the initial position and rotation of facilitating hand base.

Success Criteria

- *Block in cup* The task is considered successful if the distance of the block center and the cup center is smaller than 0.035 meters. This distance makes sure the task is only considered successful when the block is inside the cup. If the block falls on the ground or has not entered the cup within a certain time step, the task is considered failed.
- *Stack* The task is considered successful if the distance between the cup centers is smaller than 0.02 meters. If either cup falls on the ground or has not been stacked within a certain time step, the task is considered failed.
- *Bottle cap* The task is considered successful if the cap is taken off from its original position 0.05 meters away within a time duration, and is considered failed otherwise.
- *Switch* The button and the switch body are connected by a revolute joint ranging from 0 to 0.5585 rads. The task is considered successful if the button is pressed and rotated 0.3585 rads within a time duration, and is considered failed otherwise.

Reward Design The reward design is similar across all tasks:

$$Reward = \alpha_1 R_{hand\ distance} + \alpha_2 R_{progress} + \alpha_3 R_{action\ penalty} + \alpha_4 R_{success\ bonus}$$

For each task, $R_{action\ penalty} = -||a(t)||^2$, and the $R_{success\ bonus}$ is the task success reward. $R_{hand\ distance}$ and $R_{progress}$ are slightly different for each tasks.

- *Block in cup*: $R_{hand\ distance} = e^{-||x_{palm} - x_{cup\ mouth}||}$, where x_{palm} is the dominant hand palm position, and $x_{cup\ mouth}$ is the position of the cup mouth. $R_{progress} = -||x_{cup} - x_{block}||$, where x_{cup} is the position of the cup, and x_{block} is the position of the block.
- *Stack*: $R_{hand\ distance} = e^{-||x_{palm} - x_{cup\ mouth}||}$, where x_{palm} is the dominant hand palm position, and $x_{cup\ mouth}$ is the position of the cup mouth, which is grasped by the facilitating hand. $R_{progress} = -||x_{cupd} - x_{cupf}||$, where x_{cupf} is the position of the cup grasped by the facilitating hand, and x_{cupd} is the position of the cup grasped by the dominant hand.
- *Bottle cap*: $R_{hand\ distance} = (1 - (||x_{index} - x_{cap}|| + ||x_{thumb} - x_{cap}||))^3$, where x_{index} and x_{thumb} are the tip position of index finger and thumb respectively, and x_{cap} is the position of the bottle cap. $R_{progress} = ||x_{cap} - x_{bottle\ top}||$, where x_{cap} is the position of the cap, and $x_{bottle\ top}$ is the position of the top of the bottle.
- *Switch*: $R_{hand\ distance} = (1 - (||x_{index} - x_{button}|| + ||x_{thumb} - x_{button}||))^3$, where x_{index} and x_{thumb} are the tip position of index finger and thumb respectively, and x_{button} is the position of the button. $R_{progress} = 2 * \theta_{button}$, where θ_{button} is the rotated angle of the joint that connects the button and the switch body.

D Real-world Task Design

In this section, we present the details for each real-world task.

State Space Design For each task, the hand joint states $\xi_f^h(t)$, $\xi_d^h(t)$ include the 16-DoF hand joint positions. We use quaternions to represent rotation part of object and hand base poses. For the *block in cup* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of the cup and the block respectively. For the *pour* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of two cups. For the *twist lid* task, $\xi_f^{obj}(t)$ and $\xi_d^{obj}(t)$ represent the poses of the jar and the lid respectively. The dimensions of the observation and action spaces of each policy are shown in Table. 5. AsymDex policy significantly reduces the state dimensions in real-world tasks consistently.

Table 5: Dimension of Observation and action spaces. For all tasks, the dimensions are identical.

	`Sym`	`Asym-w/o-rel`	`Rel-w/o-Asym`	`AsymDex` (ours)
Observation	60	44	53	30
Action	44	28	38	22

Sampling Procedure The corresponding simulation environments of real-world tasks include two arms and two hands attached to them. Hence, we randomize the initial position of objects and hand poses by randomizing the initial joint angles of two Kinova arms. Then we initialize the poses of objects according to the initial hand poses.

- *Block in cup*: The default initial joint angles of the 7-DoF Kinova arms attached to the dominant hand are $[0.0, 0.8, 0, \pi/2+0.5, 0, -1.3, -\pi/2]$. We randomized each angle by adding a $\delta\theta_i$: $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.1, 0.1), \quad \forall i \in \{1, \ldots, 4\}$; $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.2, 0.2), \quad \forall i \in \{5, 6\}$; $\delta\theta_7 \in \Theta_7 \sim \mathcal{U}(-0.3, 0.3)$. The default initial joint angles of the 7-DoF Kinova arms attached to the facilitating hand are $[-0.0, 0.8, 0, \pi/2 + 0.5, 0, -1.3, 0]$. We randomized each angle by adding a $\delta\theta_i$: $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.0, 0.0), \quad \forall i \in \{1, \ldots, 3\}$; $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.1, 0.1), \quad \forall i \in \{4, 7\}$.
- *Pour*: The default initial joint angles of the 7-DoF Kinova arms attached to the dominant hand are $[0.2, 0.8, 0, \pi/2+0.5, 0, -1.3, -\pi/2]$. We randomized each angle by adding a $\delta\theta_i$: $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.1, 0.1), \quad \forall i \in \{1, \ldots, 4\}$; $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.2, 0.2), \quad \forall i \in \{5, 6\}$; $\delta\theta_7 \in \Theta_7 \sim \mathcal{U}(-1.1, 0.6)$. The default initial joint angles of the 7-DoF Kinova arms attached to the facilitating hand are $[-0.2, 0.8, 0, \pi/2 + 0.5, 0, -1.3, 0]$. We randomized each angle by adding a $\delta\theta_i$: $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.0, 0.0), \quad \forall i \in \{1, \ldots, 3\}$; $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.1, 0.1), \quad \forall i \in \{4, 7\}$.
- *Twist lid*: The default initial joint angles of the 7-DoF Kinova arms attached to the dominant hand are $[0.1, 0.5, 0, \pi/2 + 0.55, 0.75, -1.5, -\pi/2 - 0.2]$. We randomized each angle by adding a $\delta\theta_i$: $\delta\theta_i \in \Theta_i \sim \mathcal{U}(-0.0, 0.0), \quad \forall i \in \{1, \ldots, 4\}$; $\delta\theta_5 \in \Theta_5 \sim \mathcal{U}(-0.05, 0.05)$; $\delta\theta_6 \in \Theta_6 \sim \mathcal{U}(-0.2, 0.0)$; $\delta\theta_7 \in \Theta_7 \sim \mathcal{U}(-0.1, 0.1)$. The default initial joint angles of the 7-DoF Kinova arms attached to the facilitating hand are $[-0.1, 0.8, 0, \pi/2 + 0.5, -0.7, -1.35, -0.1]$.

Success Criteria

- *Block in cup* The task is considered successful if the distance of the block center and the cup center is smaller than 0.035 meters. This distance makes sure the task is only considered successful when the block is inside the cup. If the block falls on the ground or has not entered the cup within a certain time step, the task is considered failed.
- *Pour* The task is considered successful if the distance between the rims of the two cups is less than 0.035 meters and the cup grasped by the facilitating hand is up-right. If either cup falls on the ground or the task is not success within a certain time step, the task is considered failed.
- *Twist lid* We utilize the articulated bottle simulation of the previous work [19]. The lid and bottle are connected by a single revolute joint. The task is considered successful if the lid is rotated over $3 \times \pi$ rad within a time duration, and is considered failed otherwise.

Reward Design The reward function structure is similar across all tasks:

$$Reward = R_{task} + \alpha_2 R_{action\ penalty} + \alpha_3 R_{success\ bonus}$$

For each task, $R_{action\ penalty} = -||a(t)||^2$, and the $R_{success\ bonus}$ is the task success bonus.

- *Block in cup*: $R_{task} = \beta_1 R_{hand\ dist} + \beta_2 R_{progress}$. $R_{hand\ dist} = e^{-||x_{palm} - x_{cup\ rim}||}$, where x_{palm} is the dominant hand palm position, and $x_{cup\ rim}$ is the position of the cup rim. $R_{progress} = -||x_{cup} - x_{block}||$, where x_{cup} is the position of the cup, and x_{block} is the position of the block.
- *Pour*: $R_{task} = \beta_1 R_{hand\ dist} + \beta_2 R_{progress} + \beta_3 R_{cup\ orient}$. $R_{hand\ dist} = e^{-||x_{palm} - x_{cup\ rim}||}$, where x_{palm} is the dominant hand palm position, and $x_{cup\ rim}$ is the position of the cup rim, which is grasped by the facilitating hand. $R_{progress} = -||x_{cupd} - x_{cupf}||$, where x_{cupf} is the position of the cup rim grasped by the facilitating hand, and x_{cupd} is the position of the cup rim grasped by the dominant hand. $R_{cup\ orient} = \mathbf{z}_{cupf} \cdot \mathbf{z}_{world}$, where $\mathbf{z}_{cupf}$ is the z-axis unit vector of the cup grasped by the facilitating hand, and $\mathbf{z}_{world}$ is the z-axis unit vector of the world frame.

- *Twist lid*: $R_{task} = \beta_1 R_{orient} + \beta_2 R_{twist} + \beta_3 R_{finger\ dist} + \beta_4 R_{hand\ dist\ penalty}$. $R_{orient} = \mathbf{z}_{bottle} \cdot \mathbf{z}_{world}$, where $\mathbf{z}_{bottle}$ is the z-axis unit vector of the bottle grasped by the facilitating hand, and $\mathbf{z}_{world}$ is the z-axis unit vector of the world frame. $R_{twist} = \theta^t_{lid} - \theta^{t-1}_{lid}$, where θ^t_{lid} is the current bottle-lid revolute joint angle, and θ^{t-1}_{lid} is the previous bottle-lid revolute joint angle. And $R_{finger\ dist}$ is the *finger contact reward* we adopted from a previous lid twisting work [19]. $R_{hand\ dist\ penalty} = -min((||x_f - x_d|| - 0.1), 0.0)$, where x_f and x_d are the position of the facilitating hand palm and the dominant hand palm.

Domain Randomization The domain randomization details are shown in Table. 6

Table 6: Domain Randomization Setup.

Object: Friction (only for *twist lid* task)	[0.5, 1.5]
Hand: Friction (only for *twist lid* task)	[0.5, 1.5]
Object Pos Observation Noise	$+\mathcal{N}(0, 0.02)$
Hand Joint Observation Noise	$+\mathcal{N}(0, 0.2)$
Hand Pos Observation Noise	$+\mathcal{N}(0, 0.02)$
Hand Orientation Observation Noise	$+\mathcal{N}(0, 0.05)$
Action Noise (only for *twist lid* task)	$+\mathcal{N}(0, 0.1)$