

You Only Teach Once: Learn One-Shot Bimanual Robotic Manipulation from Video Demonstrations

Huayi Zhou*, Ruixiang Wang[†], Yunxin Tai[‡], Yueci Deng[‡], Guiliang Liu* and Kui Jia^{*§}

*The Chinese University of Hong Kong, Shenzhen. [†]Harbin Institute of Technology, Weihai.

[‡]DexForce, Shenzhen. [§]The Corresponding Author.

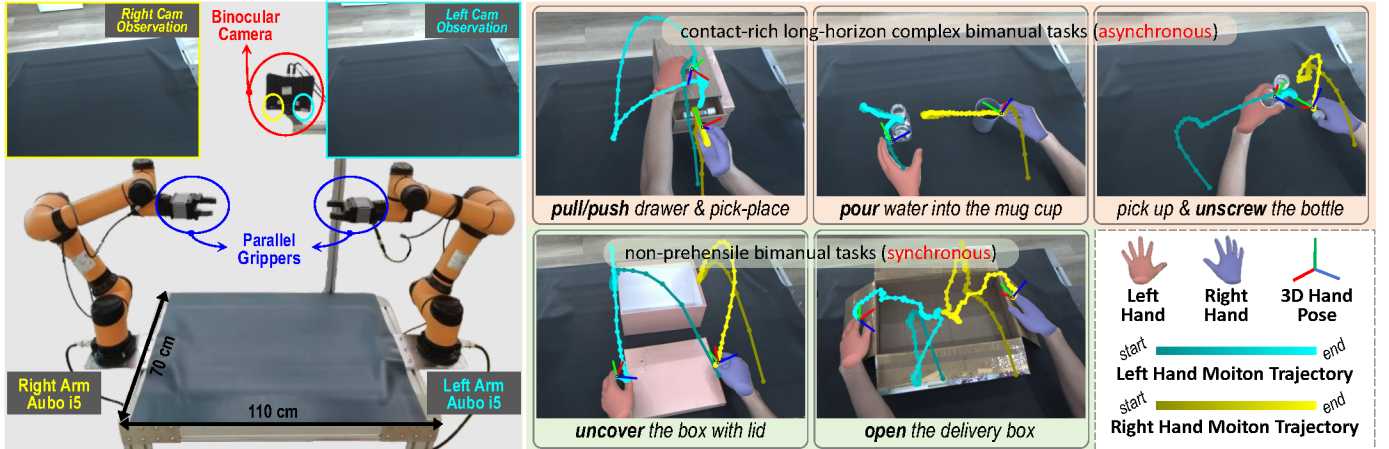


Fig. 1: Our proposed **YOTO** (You Only Teach Once) facilitates various complex long-horizon bimanual tasks. It needs only a one-shot observation of a single third-person binocular camera to extract the fine-grained motion trajectory of human hands, which can then be utilized for the dual-arm coordinated action injection and rapid proliferation of training demonstrations.

Abstract—Bimanual robotic manipulation is a long-standing challenge of embodied intelligence due to its characteristics of dual-arm spatial-temporal coordination and high-dimensional action spaces. Previous studies rely on pre-defined action taxonomies or direct teleoperation to alleviate or circumvent these issues, often making them lack simplicity, versatility and scalability. Differently, we believe that the most effective and efficient way for teaching bimanual manipulation is learning from human demonstrated videos, where rich features such as spatial-temporal positions, dynamic postures, interaction states and dexterous transitions are available almost for free. In this work, we propose the YOTO (You Only Teach Once), which can extract and then inject patterns of bimanual actions from as few as a single binocular observation of hand movements, and teach dual robot arms various complex tasks. Furthermore, based on keyframes-based motion trajectories, we devise a subtle solution for rapidly generating training demonstrations with diverse variations of manipulated objects and their locations. These data can then be used to learn a customized bimanual diffusion policy (BiDP) across diverse scenes. In experiments, YOTO achieves impressive performance in mimicking 5 intricate long-horizon bimanual tasks, possesses strong generalization under different visual and spatial conditions, and outperforms existing visuomotor imitation learning methods in accuracy and efficiency. Our project link is <https://hnuzhy.github.io/projects/YOTO>.

I. INTRODUCTION

Bimanual manipulation is an enduring topic in the robotics community [5, 68, 82, 35, 91, 16, 21]. It has been widely involved in many other fields such as bionics, high-end

manufacturing, mechanical control, reinforcement learning and computer vision. Despite this, achieving efficient, precise and robust manipulation of dual-arm robots to accomplish various daily tasks remains a difficult research area. Generally, there are two main challenges: coordination and state complexity [32, 51]. On the one hand, the two arms working together need to move alternately or simultaneously in a coordinated, non-procrastinated manner and avoid collisions with the scene or each other. This places stringent demands on the control and scheduling scheme. On the other hand, the total degrees of freedom of two arms and their respective end effectors are distributed in a higher-dimensional space than a single arm. This makes the design of motion planning and action prediction more challenging. Given these difficulties, it is no small feat to drive two robot arms to perform tasks that human toddlers can do with ease, such as uncovering lids, assembling blocks and lifting large-size objects, let alone mastering many more complex long-horizon skills.

The mainstream bimanual manipulation research includes two major branches: explicitly classifying tasks based on pre-defined taxonomy [82, 45, 32, 51] and implicitly learning from demonstrations collected by teleoperation [104, 65, 62, 53]. The former often fails to uniformly cover arbitrary tasks and also limits the flexibility of the robot arm. While the latter requires substantial training data which is inconvenient

to scale up. And collected demonstrations are intrinsically non-stationary and despatialized, which is not conducive to training robust and generalizable action policies. In addition to taxonomy and teleoperation, an indirect but more plausible and interpretable route is to learn from human action videos [4, 107, 25, 13, 47, 69, 42]. This route is based on relatively mature vision techniques to process human demonstrations and extract high-level features for generating robot manipulation-relevant elements. In this paper, we also follow this promising path. Our dual-arm workbench, hardware settings, and selected bimanual tasks are shown in Fig. 1. And the overall framework is shown in Fig. 2.

Specifically, we focus on understanding human hands, including their location, left-rightness, 3D shape, joints, pose, contact, and open/closed state. These features can be perceived using hand-related vision methods [71, 78, 67]. After extracting hand motion trajectories, we do not simply inject step-wise actions into robots. Because visual perception results are inevitably erroneous, and real hand motions are jittery and discontinuous. We thus simplify the consecutive trajectory into discrete keyframes [38, 80], and assign the corresponding keyposes to two arms to execute by applying inverse kinematics interpolation. Besides, we also record and replay the order of dual-hand movements (termed as *motion mask*), which can help to address the dual-arm coordination issue in long-horizon bimanual tasks. Now, we successfully obtain a stable and refined manipulation motion exemplar.

More than that, thanks to the editability of obtained single teaching, we devise rapid proliferation strategies of training demonstrations. First, we change the 6-DoF pose of task-related objects and adjust corresponding keyposes to let real robots replay similar actions. Objects can also be replaced with other ones of analogous shape and size. This auto-rollout operation is stable and much faster than teleoperation [104, 86]. For example, we can collect about 300 demonstrations in 8 hours based on a well-taught task. On the other hand, after knowing the reachable area of manipulators, we can perform geometric transformation on segmented object point clouds, which can be extracted by using open vocabulary segmentation [90, 73] and binocular stereo matching [92, 93]. Such augmentation is more reliable and efficient than rollout. Therefore, mixing the above two data expansion schemes, we call it proliferation, just like the generation of cells.

With sufficient training data, we follow diffusion-based visuomotor imitation methods [15, 98, 95] and propose a specialized bimanual diffusion policy (BiDP), which is customized for learning long-horizon dual-arm tasks. It has three major improvements. First, we reduce observations (*e.g.*, 3D point clouds) from the entire scene to manipulated objects to accelerate training convergence and eliminate irrelevant terms [26, 51]. Then, instead of modeling continuous actions, we choose to predict essential keyposes [55, 89, 41, 99], which can greatly decrease the diffusion space dimensionality. Third, we utilize the motion mask to determine the alternating or synchronous dual-arm moving, and reorganize the bimanual action space to train a unified action policy. In experiments,

we have verified the high efficiency and effectiveness of BiDP on challenging bimanual tasks.

Overall, we have the following four contributions:

- We present a paradigm for extracting and injecting dual-arm movements from a one-shot observation of human hands demonstration, which supports the fast transfer of bimanual manipulation skills to two robotic arms.
- We develop a solution for rapidly proliferating training demonstrations based on one-shot teaching, which is more convenient and reliable than teleoperation.
- We propose a dedicated bimanual diffusion policy (BiDP) algorithm that can efficiently and effectively assist dual-arm manipulators in imitating complex skills.
- Our framework YOTO is compatible with most bimanual tasks. We verified its effectiveness and superiority on 5 complex long-horizon manipulation tasks (including synchronous and asynchronous).

II. RELATED WORKS

A. Bimanual Robotic Manipulation

Many bimanual manipulation methods focus on specialized tasks or primitive skills, such as cloth-folding [56, 6, 19, 88, 2, 9, 77], bagging [11, 10, 3], untangling [27, 69], untwisting [49, 4], throwing/catching [37, 94, 48], scooping [28], carrying [81] and dressing [108]. For general bimanual manipulation, typical research [82, 35, 61, 45, 33, 106, 97] tends to explicitly classify them into uncoordinated and coordinated, or symmetrical and asymmetrical according to task characteristics. Some homologous approaches assume that two arms form a leader-follower [52, 32] or stabilizer-actor [29, 51] pair. Most recently, the ALOHA series [104, 24, 1, 105] have revolutionized bimanual manipulation by dexterous teleoperating and upgrading low-cost hardwares of real-world robotics. These similar works [104, 84, 43, 62, 53, 7] implicitly train an end-to-end imitation network using massive and diverse teleoperated data, expecting to get generalized large robotic models. To further improve dual-arm reachability and dexterity, some studies have equipped multi-finger hands [50, 86, 79, 20, 14, 23], mobile footplates [96, 95, 102, 24], tactile feedbacks [50, 20, 12] or active cameras [17, 14]. In contrast to them, our manipulators are two fixed-base robot arms with parallel-jaw grippers. We propose an universal framework that learns bimanual policies with considering the dual-arm coordination. And the training data is not collected via teleoperation but proliferated from a single-shot demonstration.

B. Learn from Human Hand Videos

Human hand videos are valuable resources for learning complex manipulation behaviors [30, 22, 100, 54, 31]. Extensive research has leveraged human demonstrations to learn robot manipulation by extracting rich non-privileged features, such as keypoints [66, 25, 87], affordances [46, 101, 63], 3D hand poses [47, 42, 4], motion trajectories [47, 42, 13, 101] and invariant correspondences [69, 44, 103]. These features can be tailored to robot-specific variables to alleviate morphology gaps, such as manipulation plans, retargeted motions

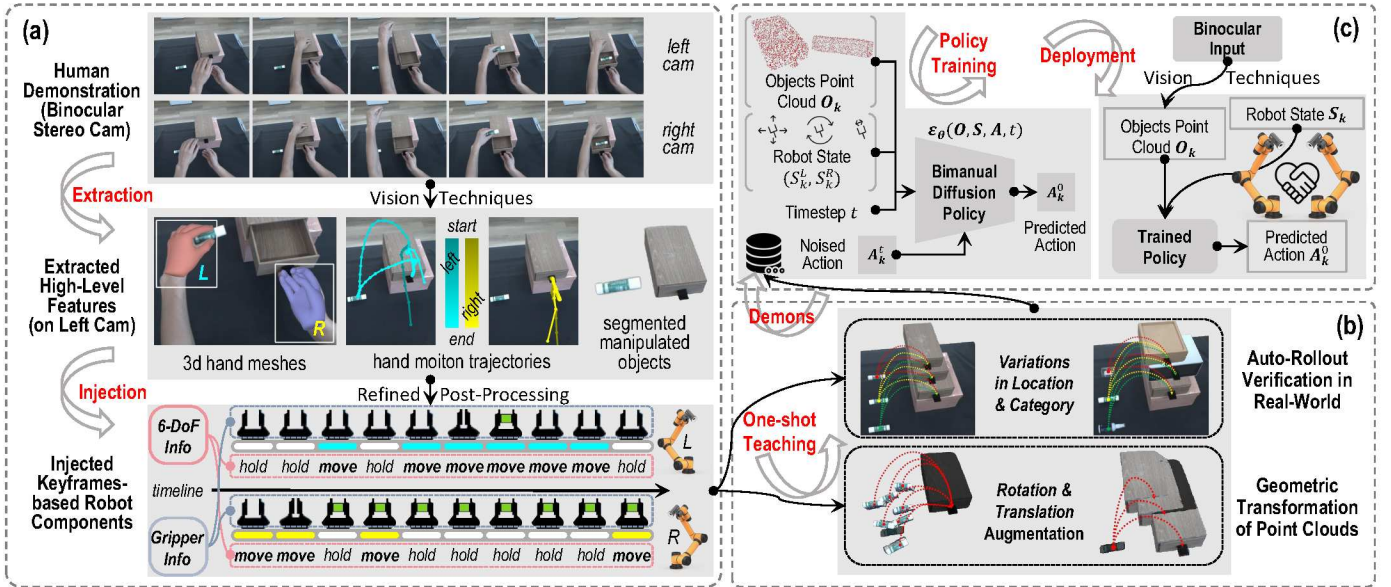


Fig. 2: The overview of our proposed YOTO. It is a general framework consists of three main modules: (a) the human hand motion extraction and injection, (b) the training demonstration proliferation from one-shot teaching, and (c) the training and deployment of a customized bimanual diffusion policy (BiDP). It is best to zoom in to view the details.

and precise actions. Two contemporary works [42, 47] also propose to use a single human demonstration to learn bimanual manipulation similar to us. RSRD [42] roughly recovers 3D part motion of articulated objects from a monocular RGB video, while we adopt a binocular camera to more accurately capture arbitrary object in 3D space. OKAMI [47] applies the object-aware motion retargeting which is noisy and non-smooth, while we devise a keyframes-based motion extraction scheme which is more robust and versatile.

C. Visuomotor Imitation Learning

Visuomotor imitation learning aims to train action prediction policies based on visual observations by exploiting labeled demonstrations [57, 40, 58, 39, 85, 38, 80]. These learned policies can drive robots to complete various manipulation with just dozens of demonstrations, covering long-horizon [57, 85], dexterous [98, 86] and bimanual [104, 79] tasks. Especially, Zhao et al. [104] introduced the action chunking transformers (ACT) to learn high-frequency controls with closed-loop feedback in an end-to-end manner. Chi et al. [15] adopted conditional denoising diffusion models [36, 83, 64] to represent visuomotor policies in robotics, exhibiting impressive training stability in modeling high-dimensional action distributions. Ze et al. [98] incorporated 3D conditioning into the original diffusion policy [15], rather than focusing on RGB images and states as conditions. Yang et al. [95] combined SIM(3)-equivariance [96, 76, 8] with diffusion policy, acquiring a more generalizable and sample-efficient visuomotor policy than [15, 98]. Inspired by them, we propose a bimanual diffusion policy (BiDP), which adds motion mask as a new diffusion condition and simplifies visual observations to task-related object point clouds, making it suitable for learning bimanual manipulation tasks.

III. HARDWARE SYSTEM

Dual-Arm Placement: Most human video-inspired bimanual manipulation works apply humanoid robots [4, 25, 47, 69, 42] or two ipsilateral arms [107, 25] to build workstations. Some bimanual teleoperations also tend to be anthropomorphic [50, 79, 14, 23] or ipsilateral [86, 12, 20]. Despite the similarity to human morphology, they are not necessarily optimal. Comparatively, it is possible to place two manipulators opposite each other, as in ALOHA series [104, 24, 1, 105] and its followers [17, 53, 7]. This heterolateral setup minimizes the overlap of accessible space and is thus compatible with a wider range of bimanual tasks. We also adopt the contralateral placement as shown in the left of Fig. 1, where each arm (Aubo i5¹) has a span of approximately 880 mm.

End Effector Selection: Although some methods utilize multi-fingered dexterous hands as end effectors [86, 79, 14, 23] and even add tactile sensors [50, 20, 12] to the hands, we still use two parallel-jaw grippers (with max opening distance 80 mm of each DH-Robotics²), which are easier to control and interpret. We will show that it is sufficient to complete complex tasks that are non-prehensile or synchronous.

Camera Observation: Many previous methods adopt the multi-view RGB observations [104, 53, 7], mainly including the global third-person camera and the local eye-in-hand camera. Other works have shown that a single third-person RGB-D camera [86, 4, 25, 47] is also acceptable. We use a binocular stereo camera (the DexSense 3D industrial camera³), similar to commercial RGB-D cameras, but providing raw left and right images to enable flexible post-processing.

¹<https://www.aubo-cobot.com/public/i5product3>

²<https://en.dh-robotics.com/product/pgi>

³<https://dexforce-3dvision.com/productinfo/1022811.html>

IV. METHOD

In this part, we introduce in detail the proposed framework YOTO, which contains three major modules and is illustrated in Fig. 2. We firstly give a basic definition of the problem in Sec. IV-A. Then, a detailed explanation of the three core modules is presented, which includes the standardized hand motion extraction and injection process in Sec. IV-B, the demonstration proliferation solution from one teaching in Sec. IV-C and the proposed visuomotor bimanual diffusion policy (BiDP) method in Sec. IV-D.

A. Problem Formulation

In this paper, we mainly consider bimanual robot manipulation tasks, where the agent (e.g., dual manipulators equipped with parallel-jaw grippers) does not have access to the ground-truth state of the environment, but visual observations O from a binocular camera and robots proprioception states S . As for the action space $A = \{a^p \in \mathbb{R}^3, a^r \in \mathbb{SO}(3), a^g \in \{0, 1\}\}$, it includes the target 6-DoF pose of each robot arm and the binary open/closed state of the gripper. Note, we focus on bimanual tasks sharing the same observations O . For the chirality, we utilize $\diamond \in \{L, R\}$ to distinguish two robot arms, such as S^L, S^R, A^L and A^R . The same applies to the difference between left and right hands below.

For imitation learning, the agent mimics manipulation plans from labeled demonstrations $\mathcal{D} = \{(\mathbf{O}, \mathbf{A})_i\}_{i=1}^N$, where N is the number of trajectories, $\mathbf{O} = \{O_t, S_t^L, S_t^R\}_{t=1}^T$ are observations of all T steps, and $\mathbf{A} = \{A_t^L, A_t^R\}_{t=1}^T$ are actions to complete the task. The learning objective can be simply concluded as a maximum likelihood observation-conditioned imitation objective to learn the policy π_θ :

$$\ell = \mathbb{E}_{(\mathbf{O}, \mathbf{A})_i \sim \mathcal{D}} \left[\sum_{t=0}^{|O|} \log \pi_\theta(A_t^\diamond | O_t, S_t^\diamond) \right]. \quad (1)$$

Next, we present how to obtain sufficient training demonstrations proliferated from only a single-shot human teaching and how to improve existing diffusion-based imitation policies for addressing the bimanual manipulation problem.

B. Hand Motion Extraction and Injection

This part corresponds to the module in Fig. 2 (a). We first manually demonstrate a long-horizon bimanual task using two hands on the dual-arm accessible operating table. Then, we leverage favourable vision techniques to extract rich manipulation features from recorded videos by a single binocular camera. Extracted features will be post-processed to obtain keyframes-based motion variables (such as 6-DoF poses and gripper states) that can drive dual arms.

1) *Human Demonstration Capturing*: By default, we capture dual-stream synchronized RGB videos with slight necessary visual difference between left and right cameras to estimate disparity and depth map. We mainly observe the left RGB view to extract a series of hand-related features, and thus always keep both hands visible to the left camera. The right view is only awakened when accurate 3D information is needed in a particular frame. This reduces the computational burden of stereo matching [92] by at least half.

Algorithm 1 3D Hand Pose Calculation.

• **Input**: 3D hand shapes $\mathcal{H}_j^\diamond$, index array of 21 pre-defined 3D hand joints I_{hand} , index numbers of wrist joint i_{wri} / index-fingertip i_{ind} / ring-fingertip i_{ring} , the given chirality $\diamond = L$ or $\diamond = R$.
• **Output**: 3D hand poses $h_j^{r, \diamond}$. // either L or R
Initialize $\mathbf{P}_j^\diamond \leftarrow \text{MANO}(\mathcal{H}_j^\diamond, I_{\text{hand}})$; // 3D hand joints indexing
 $p_{\text{wri}} \leftarrow \mathbf{P}_j^\diamond[i_{\text{wri}}]$, $p_{\text{ind}} \leftarrow \mathbf{P}_j^\diamond[i_{\text{ind}}]$, $p_{\text{ring}} \leftarrow \mathbf{P}_j^\diamond[i_{\text{ring}}]$;
 $l_{\text{iw}} \leftarrow (p_{\text{ind}} - p_{\text{wri}})$, $l_{\text{rw}} \leftarrow (p_{\text{ring}} - p_{\text{wri}})$; // two 3D lines
 $v_z \leftarrow \text{CROSS_PRODUCT}(l_{\text{iw}}, l_{\text{rw}})$; // Z-axis direction
 $\bar{v}_z \leftarrow v_z / (\text{NORMALIZE}(v_z) + 1e-8)$; // vector normalization
 $v_y \leftarrow l_{\text{mid}} \leftarrow (l_{\text{iw}} + l_{\text{rw}}) / 2.0$; // middle line (Y-axis direction)
 $\bar{v}_y \leftarrow v_y / (\text{NORMALIZE}(v_y) + 1e-8)$; // vector normalization
 $\bar{v}_x \leftarrow \text{CROSS_PRODUCT}(\bar{v}_y, \bar{v}_z)$; // X-axis direction
 $v_{\text{rot}} \leftarrow \text{CONCATENATE}([\bar{v}_x, \bar{v}_y, \bar{v}_z])$; // final 3×3 rotation matrix
return v_{rot} ;

2) *High-Level Features Extraction*: Given a video demonstration (the left stream) of one specified bimanual task, we run our vision perception pipeline to obtain the 3D point trajectories and status of two hands.

3D point trajectories. We first use WiLoR [71] to detect bounding boxes of left and right hands in each frame and then estimate their 3D shapes $\mathcal{H}^L$ and $\mathcal{H}^R$ represented by MANO [74]. Then, we simply track the center point $h_j^{p, \diamond} = (x_j^\diamond, y_j^\diamond, z_j^\diamond)$ of each hand and obtain the 3D hands sequence $\mathbf{H} = \{(\mathcal{H}_j^\diamond, h_j^{p, \diamond})\}_{j=1}^J$, where $\diamond$ is the chirality and j is the index among all J frames. The $h_j^{p, \diamond}$ can be calculated by averaging several selected points (e.g., five finger tips) from 21 pre-defined joints of the 3D hand model $\mathcal{H}_j^\diamond$.

As of here, many similar works [42, 47, 14, 23] choose to retarget the produced continuous trajectories $\{h_j^{p, \diamond}\}_{j=1}^J$ to their end effectors through estimated 3D geometric transformations. However, considering the inherent errors of hand-related vision algorithms in left-right classification and 3D shape regression, we cannot fully trust trajectories directly derived from them. In particular, current state-of-the-art 3D hand mesh reconstruction methods, such as WiLoR [71] and HaMeR [67], still cannot achieve continuous and consistent prediction in a given camera space. This is also pointed out and verified by DexCap [86]. More examples can be found in Fig. 5. As an alternative, we propose to project all 3D points $\{h_j^{p, \diamond}\}_{j=1}^J$ onto the 2D image, and then lift these points to 3D by applying the stereo matching algorithm [92]. The final back-projected 3D point trajectories are $\{\bar{h}_j^{p, \diamond}\}_{j=1}^J$, which are guaranteed to be more stable in the given camera space.

States of two hands. In order to fully map hand movements to two-fingered grippers, we also need to determine the 3D orientations $h_j^{r, \diamond}$ and open/closed states $h_j^{g, \diamond}$ by further observing 3D hands $\mathcal{H}_j^\diamond$. Here, we can estimate the open/closed state by detecting if the hand is in contact with an object [78]. If there is contact, the hand is considered closed ($h_j^{g, \diamond} = 0$), otherwise open ($h_j^{g, \diamond} = 1$). This is more trustworthy than relying solely on hands to estimate status. For calculating 3D hand poses $h_j^{r, \diamond}$, we need to simplify the hand into a lower-dimensional gripper, which is analogous to the eigengrASPing [60, 18]. We summarize this process in Alg. 1. To this point, we have

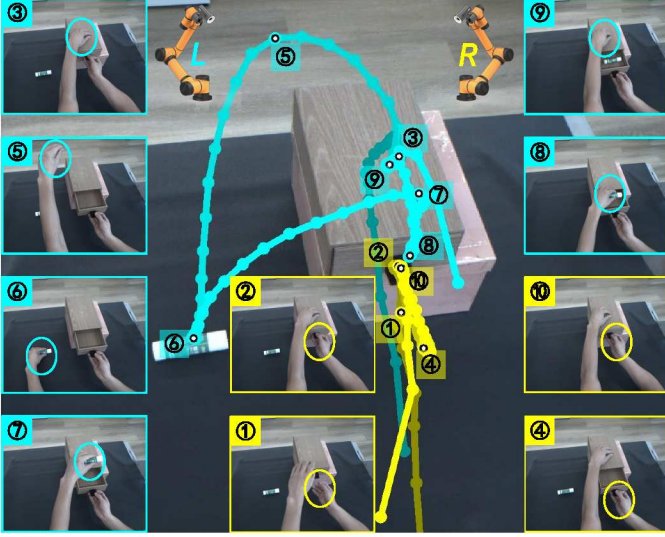


Fig. 3: A detailed example of extracted motion trajectories with corresponding keyframes of both left hand and right hand. It is best to zoom in to view the details.

obtained the rough motion trajectories purely based on human hand videos $\{(\hat{h}_j^{p,\diamond}, \hat{h}_j^{r,\diamond}, \hat{h}_j^{g,\diamond})\}_{j=1}^J$.

Additionally, we adopt cutting-edge vision algorithms (including the vision-language model Florence-2 [90] and SAM2 [73]) to extract segmented manipulated objects from the left initial image as our disturbance-free visual observations $\hat{O}$, which will be further lifted to 3D point clouds $\tilde{O}$ by applying stereo matching approaches [92, 93].

3) *Robot Actions Injection*: Although we have obtained robot-oriented motion trajectories, their validity and usability are still concerns. For example, some target poses may be unreachable for the failed inverse kinematics. Due to agnostic structures, two arms may collide at some point. An obvious approach is to replay and verify the rationality of each action step by step directly on real robots, but this choice is unsafe and inefficient, considering that the total number of frames J is usually about 100 to 200.

Keyframes-based motion actions. To this end, we turn to a more reasonable and safer post-processing, namely keyframes-based motion simplification and injection. Specifically, we inherit the abstraction of a consequent demonstration into discrete keyframes (*a.k.a.* keyposes) as in C2FARM [38] and PerAct [80]. Keyframes are important intermediate end-effector poses that summarize a demonstration and can be auto-extracted using simple heuristics, such as a change in the open/close end-effector state or local extrema of velocity/acceleration. This concept is widely used in long-horizon manipulation studies [55, 89, 41, 99]. Accordingly, we can just learn to predict the next best keyframe, and use a sampling-based motion planner to reach it during inference. We thus simplify trajectories $\{(\hat{h}_j^{p,\diamond}, \hat{h}_j^{r,\diamond}, \hat{h}_j^{g,\diamond})\}_{j=1}^J$ into a set of keyframes $\{(\tilde{h}_k^{p,\diamond}, \tilde{h}_k^{r,\diamond}, \tilde{h}_k^{g,\diamond})\}_{k=1}^K$, where k is the index of K keyframes. K is around 10 in our tasks ($K \ll J$), which makes it much more easier to quickly verify and correct errors.

To inject these keyposes into the dual-arm robot, we need to transform them from the camera coordinate to the robot coordinate using the pre-measured hand-eye calibration transformation matrix. Usually, a real-robot verification takes about three minutes. We finally update the verified trajectories into $\tilde{\mathbf{A}} = \{(\tilde{a}_k^{p,\diamond}, \tilde{a}_k^{r,\diamond}, \tilde{a}_k^{g,\diamond})\}_{k=1}^K$, which consists of the successfully injected K robot actions. An elaborate example of extracted keyframes is shown in Fig. 3.

Derivation of motion mask. Additionally, we should always care about the dual-arm spatial-temporal coordination, which is one of the core issues of bimanual manipulation. Fortunately, when we extract the hand motions, we already have a time record in every frame, which represents the refined keyframes-based set $\tilde{\mathbf{A}}$ naturally contains detailed timestamps. Based on it, we can thus derive the corresponding coordination strategy $\mathbf{C} = \{(\mathcal{C}_k^L, \mathcal{C}_k^R) | \mathcal{C}_k^\diamond \in \{0, 1\}\}_{k=1}^K$, where $\mathcal{C}_k^\diamond$ means the motion state of a robot arm at the k -th keyframe. The binary value 0 means holding on, 1 means moving on. Given this particularity, we name it *motion mask* to schedule robot motion. A specific illustration of $\mathbf{C}$ for the pull drawer task can be found in the down-left corner of Fig. 2. This example is broadly applicable to strictly asynchronous bimanual tasks (*e.g.*, $\mathcal{C}_k^L \neq \mathcal{C}_k^R$). While, for fully synchronous manipulation tasks, values of $\mathcal{C}_k^L$ and $\mathcal{C}_k^R$ in $\mathbf{C}$ keep the same. Currently, we do not consider those long-horizon tasks where synchronized and asynchronized keyframes are mixed.

In the following, we show that the extracted fine-grained keyframes-based motion actions $\tilde{\mathbf{A}}$ along with the corresponding *motion mask* $\mathbf{C}$ will continue to play a vital role.

C. Demonstration Proliferation from One Teaching

Based on the one-shot teaching, we propose two demonstration proliferation schemes, the automatic rollout verification of real robots and point cloud-level geometry augmentation of manipulated objects. This solution is an efficient and reliable route to quickly produce training data for imitation learning. An example is shown in Fig. 2 (b).

1) *Auto-Rollout Verification in Real-World*: Formally, our refined keyframes-based robot actions $\tilde{\mathbf{A}}$ are interpretable and editable. These properties assist us to conduct automated demonstration rollout verification and collection on real robots. First, we can easily split $\tilde{\mathbf{A}}$ into two distinctive trajectories $\tilde{\mathbf{A}}^L$ and $\tilde{\mathbf{A}}^R$ belonging to the left and right robotic arms based on the motion mask $\mathbf{C}$. Below is for decomposing strictly asynchronous tasks.

$$\begin{cases} \tilde{\mathbf{A}}^L &= \{(\tilde{a}_k^{p,L}, \tilde{a}_k^{r,L}, \tilde{a}_k^{g,L}) | \mathcal{C}_k^L = 1, \mathcal{C}_k^R = 0\}, \\ \tilde{\mathbf{A}}^R &= \{(\tilde{a}_k^{p,R}, \tilde{a}_k^{r,R}, \tilde{a}_k^{g,R}) | \mathcal{C}_k^L = 0, \mathcal{C}_k^R = 1\}, \\ K &= |\tilde{\mathbf{A}}^L| + |\tilde{\mathbf{A}}^R| = |\tilde{\mathbf{A}}|/2, \end{cases} \quad (2)$$

where we actually eliminate K redundant keyposes for unilateral arm waiting (holding on actions). For synchronous tasks ($|\tilde{\mathbf{A}}^L| = |\tilde{\mathbf{A}}^R| = K$), we always have to drive both arms, so there is no need to apply the motion mask.

The above allows two arms to disengage smoothly. Then, we can precisely edit any keyframe in $\tilde{\mathbf{A}}^L$ or $\tilde{\mathbf{A}}^R$ closely related to the manipulated object to align with its changed

TABLE I: The time comparison of different data collection or expansion methods. We report the average completion time for 3 tasks, 10 valid trials in total for each task. The † means it can be achieved by directly modifying the script.

Methods	Operators	Arms	Long-Horizon Bimanual Tasks		
			pull drawer (s)	pour water (s)	unscrew bottle (s)
Master-Slave	2	2	204.8	226.2	247.9
Drag&Drop	2	1	100.7	115.4	123.6
Auto-Rollout	1	1	41.5	52.1	51.4
Geo-Trans †	1	0	1.5	1.5	1.0

keypose in real-world. We still take the pull drawer task (with 10 keyframes) as an example. When moving the object picked up by the left arm, we need to adjust the 6-th keypose $\tilde{a}_6^L = (\tilde{a}_6^{p,L}, \tilde{a}_6^{r,L}, \tilde{a}_6^{g,L})$. For example, if we move the object 5 cm along the X-axis positive direction, we then just add an offset $(0.05, 0.00, 0.00)$ to the position part $\tilde{a}_6^{p,L}$. Moreover, we can also replace objects with similar shapes in the same position to expand category diversity. Finally, we conduct the rollout to get a new demonstration. The same is true for adjusting the drawer manipulated by the right arm. Regardless of simplicity, we compared auto-rollout with two popular data collection methods, master-slave arm synchronization and drag-and-drop teaching, and found that it is more efficient. See Tab. I for the comparison. The other two ways are hampered by multi-operators and higher failure rates.

2) *Geometric Transformation of Point Clouds*: Regarding the above expansion of object positions and categories in real-world, we still have to verify them one by one. We thus expect to reliably augment visual observations of manipulated objects (the extracted 3D point clouds $\tilde{O}$) any number of times, so that theoretically infinite demonstrations can be obtained. In the auto-rollout stage, we have initially figured out the correspondence between manipulated objects and their relevant keyframes. Now, we can perform geometric transformations (mainly controlled rotations and translations) on the objects at the point cloud level, and update the 6-DoF values in the corresponding keyframes. In this way, matching pairs of visual observations $\tilde{O}$ and keyframes-based actions $\tilde{A}$ can be generated in batches, forming a series of new training data, which no longer need to be verified in real robots. It should be noted that the geometric transformation of $\tilde{O}$ is restricted, that is, it cannot exceed the reach of the robot arm. Fortunately, the rational moving range of manipulated objects can be measured during the auto-rollout phase incidentally. In Tab. I, we have added the time comparison of this data proliferation, which maintains the highest efficiency.

D. Bimanual Diffusion Policy Learning

In this part, we adapt popular visuomotor diffusion policies [15, 98, 95], and propose a customized bimanual diffusion policy (BiDP) to enable fast and robust imitation of long-horizon tasks. We firstly shrink the input observations into task-relevant object point clouds, allowing the policy model to converge quickly and resistant to interference. Additionally, we devise a motion mask to unify the action prediction and

address the dual-arm coordination problem.

Bimanual dataset composition. According to the definition in Sec. IV-A, we rewrite the training set as $\tilde{\mathcal{D}} = \{(\tilde{O}, \tilde{A}, C)_i\}_{i=1}^N$, where N is the number of demonstrations. C is the motion mask containing coordination strategies. $\tilde{\mathcal{D}}$ is generated by applying our proposed data proliferation solution to expand the seeding one-shot teaching to get a large dataset with hundreds or thousands of trajectories. Here, we update $\tilde{O} = \{\tilde{O}_k, S_k^L, S_k^R\}_{k=1}^K$ and $\tilde{A} = \{(\tilde{a}_k^{p,\diamond}, \tilde{a}_k^{r,\diamond}, \tilde{a}_k^{g,\diamond})\}_{k=1}^K$, where $\tilde{O}_k$ is the observation containing 3D point clouds of manipulated objects instead of the entire RGB image [15] or point clouds scene [98, 95]. S_k^L and S_k^R are robot proprioception states with similar formats as actions $S_k^\diamond = (\tilde{s}_k^{p,\diamond}, \tilde{s}_k^{r,\diamond}, \tilde{s}_k^{g,\diamond})$. $\tilde{A}$ have discrete keyposes, rather than continuous and dense robot states. Learning to predict keyposes is common in robotic manipulation [55, 89, 41, 99]. The policy needs to learn a mapping from the initial observation $\tilde{O}_1$ to all subsequent keyposes $\tilde{A}$ for two arms. The history horizon and prediction horizon is 1 and K , respectively. In evaluation, the policy predicts all actions to be executed conditioned only on an one-shot observation $\{\tilde{O}_1, S_1^L, S_1^R\}$ at first sight.

Diffusion-based policy representation. Similar to [15, 98], we utilize Denoising Diffusion Probabilistic Models (DDPMs) [36] to model the conditional distribution $p(\tilde{A}_k|\tilde{O}_k)$. Starting from the random Gaussian noise $\tilde{A}_k^T$, where T means diffusion steps, DDPM performs T iterations of denoising to predict actions with decreasing levels of noise, gradually from $\tilde{A}_k^{T-1}$ to $\tilde{A}_k^0$. This process follows:

$$\tilde{A}_k^{t-1} = \alpha(\tilde{A}_k^t - \gamma \varepsilon_\theta(\tilde{O}_k, \tilde{A}_k^t, t) + \mathcal{N}(0, \sigma^2, I)). \quad (3)$$

The policy finally outputs $\tilde{A}_k^0$. Because point clouds are used as the visual input instead of RGB images, we adopt more robust SIM(3)-equivariant architectures [96, 95], rather than policies based on CNNs [15] or transformers [98]. Formally, the noise prediction network ε_θ takes observation $\tilde{O}_k$, noisy action $\tilde{A}_k$ and diffusion timestep t as input, and predicts the gradient $\nabla \mathbf{E}(\tilde{A}_k)$ for denoising the noisy action input. It first uses a modified PointNet-based [72] encoder with SIM(3)-equivariance to encode visual observations. The encoded visual features and positional embeddings of t are passed to FiLM layers [70]. Then, the policy network applies a convolutional U-Net [75] to process $\tilde{A}_k, t$ and the conditioned observations to predict denoising gradients. Note that $\tilde{O}_k, \tilde{A}_k$ and $\tilde{A}_k^0$ are processed to be invariant to scale and position. Above-mentioned FiLM layers, convolutional U-net, and other connecting layers are also modulated to be $\mathbb{SO}(3)$ -equivariant. Please refer to [96, 95] for more details.

Customized bimanual diffusion policy. Since $\tilde{A}_k$ and $S_k^\diamond$ contain dual-arm actions in our task, it is important to preprocess them appropriately. A vanilla approach is to predict all actions in each keyframe, including $(\tilde{a}_k^{p,L}, \tilde{a}_k^{r,L}, \tilde{a}_k^{g,L})$ and $(\tilde{a}_k^{p,R}, \tilde{a}_k^{r,R}, \tilde{a}_k^{g,R})$. This not only needs to re-splice the position, rotation, and gripper data and modify the diffusion-based policy network accordingly, but also learns redundant actions for asynchronous tasks (as pointed out in Sec. IV-C),



Fig. 4: We collected a variety of manipulated objects in instance-level for each of five bimanual tasks to improve and verify the generalizability of trained policies. All of these objects are from everyday life, not intentionally customized.

which is inefficient and error-prone. To this end, we reorganize the action space into $\bar{\mathbf{A}} = \{\bar{\mathbf{A}}^L, \bar{\mathbf{A}}^R\}$ based on the motion mask $\mathbf{C}$ according to Eqn. 2. $\bar{\mathbf{A}}$ contains a series of time-ordered single-arm actions, which is a mixture of the left and right with removing potential redundancy. Taking the pull drawer task as an example, a demonstration consists of 10 keyframes $\{\bar{A}_1^R, \bar{A}_2^R, \bar{A}_3^L, \bar{A}_4^R, \bar{A}_5^L, \bar{A}_6^L, \bar{A}_7^L, \bar{A}_8^L, \bar{A}_9^L, \bar{A}_{10}^R\}$. For synchronous tasks, the left and right sides appear alternately. In this way, we unify the policy network form of bimanual tasks, which is also compatible with single-arm. More implementation details are in supplementary materials.

V. EXPERIMENTS

We aim to answer the following research questions. Q1: What is the quality of our extracted hand motions? Q2: Can the various strategies introduced in the YOTO framework enable it to better learn bimanual manipulation policies? Q3: Do trained BiDP models generalize outside of the in-distribution domain? Q4: Is the presented YOTO framework compatible with a variety of long-horizon complex tasks?

A. Experiment Setups

1) *Tasks*: We evaluate YOTO on five real-world bimanual tasks, including pull drawer, pour water, unscrew bottle, uncover lid and open box. These tasks collectively encompass two types of dual-arm collaborations: strictly asynchronous and synchronous. The manipulated objects in these tasks might be rigid, articulated, deformable or non-prehensile. They also involve many primitive skills such as pull/push, pick/place, re-orient, unscrew, revolve and lift up. Some skills must require both arms to complete. More importantly, all tasks are long-horizon, indicating that they are quite complex due to containing multiple substeps. In the following, we explain each task in brief:

- **pull drawer**: A drawer and a daily pocketed object. It consists of 6 substeps including stable the drawer (L), pull the drawer (R), pick up the object (L), place the object into the drawer (L), stable the drawer (L), and push the drawer (R).

TABLE II: Detailed statistics of five bimanual tasks. The $\dagger$ means we only count these auto-rollout demonstrations.

Task Names	pull drawer	pour water	unscrew bottle	uncover lid	open box
Is Synchronous?	$\times$	$\times$	$\times$	$\checkmark$	$\checkmark$
# Manipulated Objects	2	2	1	1	1
# Substeps	6	6	5	3	4
# Keyframes	10	11	12	12	16
Avg. Duration (s)	42	53	51	27	35
# Categories	9 3	6 3	6	5	4
# Demonstrations $\dagger$	243	162	54	45	36

- **pour water**: A capless bottle with water and an empty mug. It consists of 6 substeps including pick up the mug (R), pick up the bottle (L), bring the mug close to the bottle (R), pour water in bottle into the mug (L), put down the bottle (L), and put down the mug (R).

- **unscrew bottle**: A capped bottle with water. It consists of 5 substeps including pick up the bottle (L), bring the bottle close to the right arm (L), unscrew the cap (R), put down the cap (R), and put down the bottle (L).

- **uncover lid**: A rectangular box with a top covered lid and no handles. It consists of 3 substeps including go to the lower middle part of the lid (LR), lift up the lid (LR), and put down the lid to one side (LR).

- **open box**: A delivery box with four handleable wings. It consists of 4 substeps including go close to the two vertical wings (LR), flick open two wings (LR), go close to the two horizontal wings (LR), and flick open two wings (LR).

The statistics of these tasks are in Tab. II, where the number of keyframes is counted based on the one-shot teaching. Examples of each task are shown in Fig. 1 and Fig. 7.

2) *Demonstrations*: Imitation learning requires sufficient training data, including diverse verified task trajectories, to learn a closed-loop action prediction policy. To this end, as described in Sec. IV-C, we start from a single-shot teaching of every task and collect a considerable number of demonstrations via the proposed rapid proliferation solution. Moreover, to improve and evaluate the generalization of learned policies,

we have collected multiple objects within each task. All related assets are shown in Fig. 4.

Specifically, we first implement the auto-rollout strategy to collect real robot data. We set 3 (for tasks `pull drawer` and `pour water`) or 9 (for the other three tasks) position variations for each manipulated object, and replace all alternatives from the assets in each position. In this way, we get training data with diverse positions and categories. The demonstration number of every task is in the last row of Tab. II, where we added statistics on their average duration. We then processed these data into the form suitable for BiDP, including extracting 3D point clouds of manipulated objects and saving the corresponding multi-step end-effector keyposes. Note that we also recorded the complete binocular video observation and continuous robot actions during each auto-rollout, so that we can reproduce mainstream policy learning methods [104, 15, 98, 95] for comparison. Next, we applied 3D geometric transformations to each demonstration, acting only on task-relevant object point clouds. These synthetically augmented data are only applicable to our proposed BiDP algorithm. After formulating the script, we finally expanded the data volume by 100 times, which results in 5K~24K trajectories per task. This magnitude is comparable to existing large-scale bimanual teleoperation methods such as RDT [53] (6K+ self-created episodes) and π_0 [7] (5~100 hours post-training data), but our cost is extremely low.

3) *Baselines*: We compare our method to four strong baselines. (1) *Action Chunking Transformers (ACT)* [104]. It is proposed by ALOHA and uses a well-designed transformer structure as the visual encoder. (2) *Diffusion Policy (DP)* [15]. The vanilla diffusion policy uses RGB images as inputs and ResNet [34] as the visual encoder. We modified it by using point cloud scenes as observations and a PointNet++ encoder [72]. (3) *3D Diffusion Policy (DP3)* [98]. It is a variant of diffusion policy with a simpler point cloud encoder. It also designs a two-layer MLP to encode robot proprioceptive states before concatenating with the observation representation. (4) *EquiBot* [95]. It takes the point cloud scene as observation, and learns to predict continuous undecomposed 7-DoF actions of dual arms. Note that these baselines, including our BiDP, are designed to learn task-independent policies, and do not consider the multi-task model currently.

4) *Metrics*: We train all methods for 500 or 1,000 epochs and only save the last checkpoint for testing. We evaluate each model with 5 trials for each single object (last three tasks with **30**, **25** and **20** trials, respectively) or 2 trials for paired objects (first two tasks with **54** and **36** trials, respectively) in every task. These objects have randomized initial placements. For a more detailed comparison, we report the **average length** (following CLAVIN [59]) in each substep for a sequenced long-horizon task, where the last substep indicates the final **success rate**. Although above tests have new variations in object placements, we choose two tasks `pull drawer` and `uncover lid` to perform more challenging out-of-distribution (OOD) evaluations on novel objects. We omit the last object or paired objects from the training set

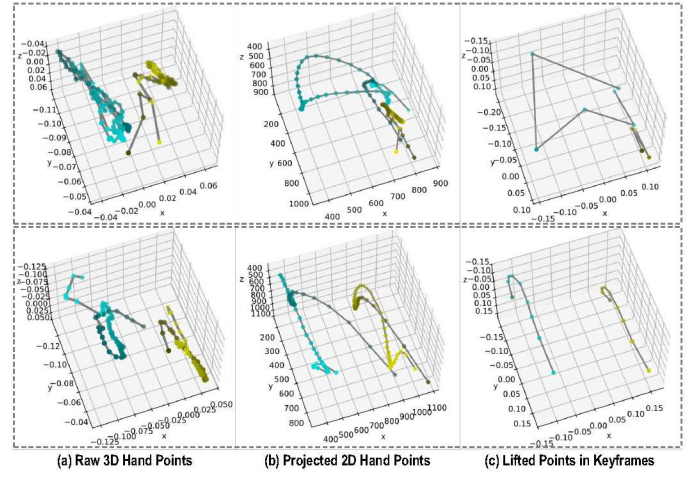


Fig. 5: Illustrations of extracted hand motion trajectories by using (a) unhandled raw 3D hand center points, (b) projected hand center points on the 2D image, and (c) lifted 3D points in simplified keyframes. The first and second line represents the task `pull drawer` and `uncover lid`, respectively.

TABLE III: Ablation studies of proposed strategies in YOTO and the bimanual diffusion policy (BiDP). The task `pull drawer` with 243 episodes is used to train all models.

Ids	purely object observation	using sparse keyframes	reorganize action space	using geometric transforms	Success Rate	Avg. Len.
1	X	X	X	X	13/54 (24.1%)	3.54
2	✓	X	X	X	26/54 (48.1%)	3.80
3	X	✓	X	X	28/54 (51.9%)	4.15
4	✓	✓	X	X	31/54 (57.4%)	4.31
5	✓	✓	✓	X	33/54 (61.1%)	4.48
6	✓	✓	X	✓	42/54 (77.8%)	5.15
7	✓	✓	✓	✓	43/54 (79.6%)	5.31

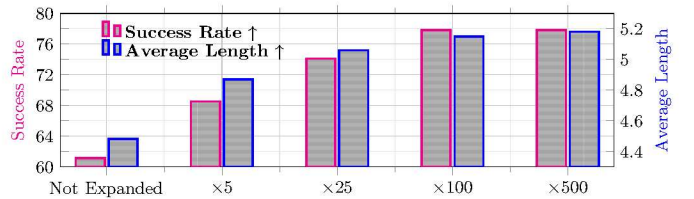


Fig. 6: Ablation studies on expanded training data at different scales using geometric transformations. The task `pull drawer` with 243 episodes is treated as the not expanded version.

and treat them as unseen objects to evaluate the final trained model. The number of all OOD trials is quadrupled.

B. Results Comparison

Here, we answer the questions raised at the beginning one by one, including basic in-distribution results and generalizations to out-of-distribution settings.

(Q1) **Our extracted hand motions have good continuity and consistency.** We first discuss the quality of the extracted motion trajectories, which is the core concept of this paper and extremely important for the various strategies developed next. As shown in Fig. 5, we compared the general effect of 3D hand

TABLE IV: Quantitative results of detailed long-horizon performance comparisons (**in-distribution evaluations**). The step-wise success rates and average length of completed task sequences are reported. We use different colors such as teal, olive and purple to indicate that each substep corresponds to the left arm, right arm and both arms, respectively.

Methods	pull drawer (243 episodes)							pour water (162 episodes)						
	stable drawer	pull drawer	pick object	place object	stable drawer	push drawer	Avg. Len.	pick mug	pick bottle	close to bottle	pour water	place bottle	place mug	Avg. Len.
ACT	42/54	26/54	18/54	15/54	09/54	05/54	2.13	28/36	24/36	23/36	03/36	03/36	03/36	2.33
DP	43/54	26/54	15/54	11/54	10/54	06/54	2.06	30/36	29/36	29/36	06/36	06/36	06/36	2.94
DP3	52/54	36/54	28/54	15/54	11/54	09/54	2.80	33/36	31/36	31/36	08/36	08/36	07/36	3.28
EquiBot	53/54	44/54	36/54	24/54	21/54	13/54	3.54	32/36	30/36	30/36	11/36	10/36	09/36	3.39
BiDP (Ours)	54/54	52/54	48/54	45/54	45/54	43/54	5.31	35/36	34/36	34/36	29/36	28/36	28/36	5.22

Methods	unscrew bottle (54 episodes)						uncover lid (45 episodes)				open box (36 episodes)				
	pick bottle	close to right	unscrew cap	place cap	place bottle	Avg. Len.	close to lid	lift up lid	place lid	Avg. Len.	close to wings	open wings	close to wings	open wings	Avg. Len.
ACT	24/30	22/30	02/30	02/30	02/30	1.73	23/25	08/25	01/25	1.28	15/20	05/20	05/20	00/20	1.25
DP	26/30	26/30	06/30	06/30	06/30	2.33	23/25	16/25	04/25	1.72	19/20	07/20	06/20	03/20	1.75
DP3	27/30	27/30	06/30	06/30	05/30	2.37	24/25	19/25	06/25	1.96	20/20	08/20	08/20	04/20	2.00
EquiBot	28/30	28/30	08/30	07/30	06/30	2.57	24/25	18/25	07/25	1.96	20/20	10/20	09/20	04/20	2.35
BiDP (Ours)	30/30	30/30	24/30	24/30	23/30	4.37	25/25	24/25	20/25	2.76	20/20	19/20	19/20	14/20	3.60

TABLE V: Comparison of the average success rate of various methods on all five tasks (**in-distribution evaluations**).

Methods	ACT	DP	DP3	EquiBot	BiDP (Ours)
Average Success Rate	5.7%	15.8%	19.4%	23.4%	76.8%

TABLE VI: Quantitative results of detailed long-horizon performance comparisons (**out-of-distribution evaluations**). The substeps are abbreviated as sequential numbers.

Methods	pull drawer (144 episodes)							uncover lid (36 episodes)				Average Success Rate
	S1	S2	S3	S4	S5	S6	Avg. Len.	S1	S2	S3	Avg. Len.	
ACT	2/8	0/8	0/8	0/8	0/8	0/8	0.25	12/20	00/20	00/20	0.60	0.0%
DP	5/8	1/8	0/8	0/8	0/8	0/8	0.75	14/20	01/20	00/20	0.75	0.0%
DP3	5/8	1/8	1/8	0/8	0/8	0/8	0.88	15/20	02/20	00/20	0.85	0.0%
EquiBot	5/8	3/8	3/8	3/8	3/8	1/8	2.25	17/20	09/20	01/20	1.25	8.8%
BiDP (Ours)	8/8	6/8	6/8	5/8	5/8	4/8	4.25	18/20	12/20	04/20	1.70	35.0%

motion trajectories extracted using different methods in two different long-horizon bimanual tasks. Firstly, when directly applying advanced 3D hand mesh reconstruction methods (either HaMeR [67] or WiLoR [71]), the resulting hand trajectory is always unstable and difficult to parse (see Fig. 5 (a)). This is mainly because most of these methods are based on monocular images, and the preset camera parameters such as focus and focal length are directly calculated using the center and size of each image. This makes the estimation results for consecutive frames in the video not in a unified and invariant camera space, and therefore unreliable and ambiguous in depth. Nevertheless, this intuitive but sub-optimal approach is still widely used by mainstream methods for learning from human videos [42, 47, 23]. In comparison, after projecting these 3D points onto a 2D image plane (with the Z-axis set to 0 for ease of visualization), it is clear that the trajectory trends and estimated motion flow are improved (see Fig. 5 (b)). This conclusion is generally applicable, for tasks like ours where the camera is stationary and its intrinsic and extrinsic parameters are known. Finally, as described in Sec IV-B, we filter out sparse keyframes from these continuous points and lift the corresponding position components into 3D points to obtain

the keyposes suitable for the end-effector (see Fig. 5 (c)). We thus claim that our extracted hand motion trajectory based on an one-shot human teaching has a more guaranteed quality. And we expect that this motion extraction technology will be used for retargeting to other more dexterous end-effectors, such as multi-fingered hands.

(Q2) **The various strategies we propose in YOTO are effective.** After extracting primary keyposes that could be successfully injected into the robot, we continue to explore YOTO including other strategies, which are closely related to the visuomotor policy learning. As shown in Tab. III, we quantitatively illustrate the effectiveness of each strategy one by one through many ablation studies. We experimented with task *pull drawer* which has 243 training trajectories. First, the method (*id-1*) without any proposed strategy can be regarded as the vanilla EquiBot [95], which takes the entire point cloud scene as observation, learns to predict continuous actions, models paired end-effector poses and leverages non-augmented training demonstrations. Despite being a solid baseline, it performed the worst on this challenging long-horizon task. Next, we replaced the input with point clouds containing only manipulated objects (*id-2*) or predicted simplified sparse keyposes (*id-3*), and the success rate and average execution length of the task were improved. These results suggest that reducing unnecessary distractions in the input and learning fewer simplified actions are the right direction. When both are used together (*id-4*), better performance can be achieved. Based on these two strategies, we decoupled the output action space and reconstructed it into a single-arm format (*id-5*), the policy could also be superior, indicating the importance of eliminating redundant actions. Alternately, if 3D geometric transformations were applied to further expand training demonstrations (*id-6*), the resulting model effect was much better, with the most prominent growth. This proves that our developed demonstration proliferation is simple yet efficient. We accordingly show in Fig. 6 the typical trend that using more extended training data leads to better performance, which is consistent with our consensus. Finally, combining the above strategies together (*id-7*), our BiDP takes full advantage

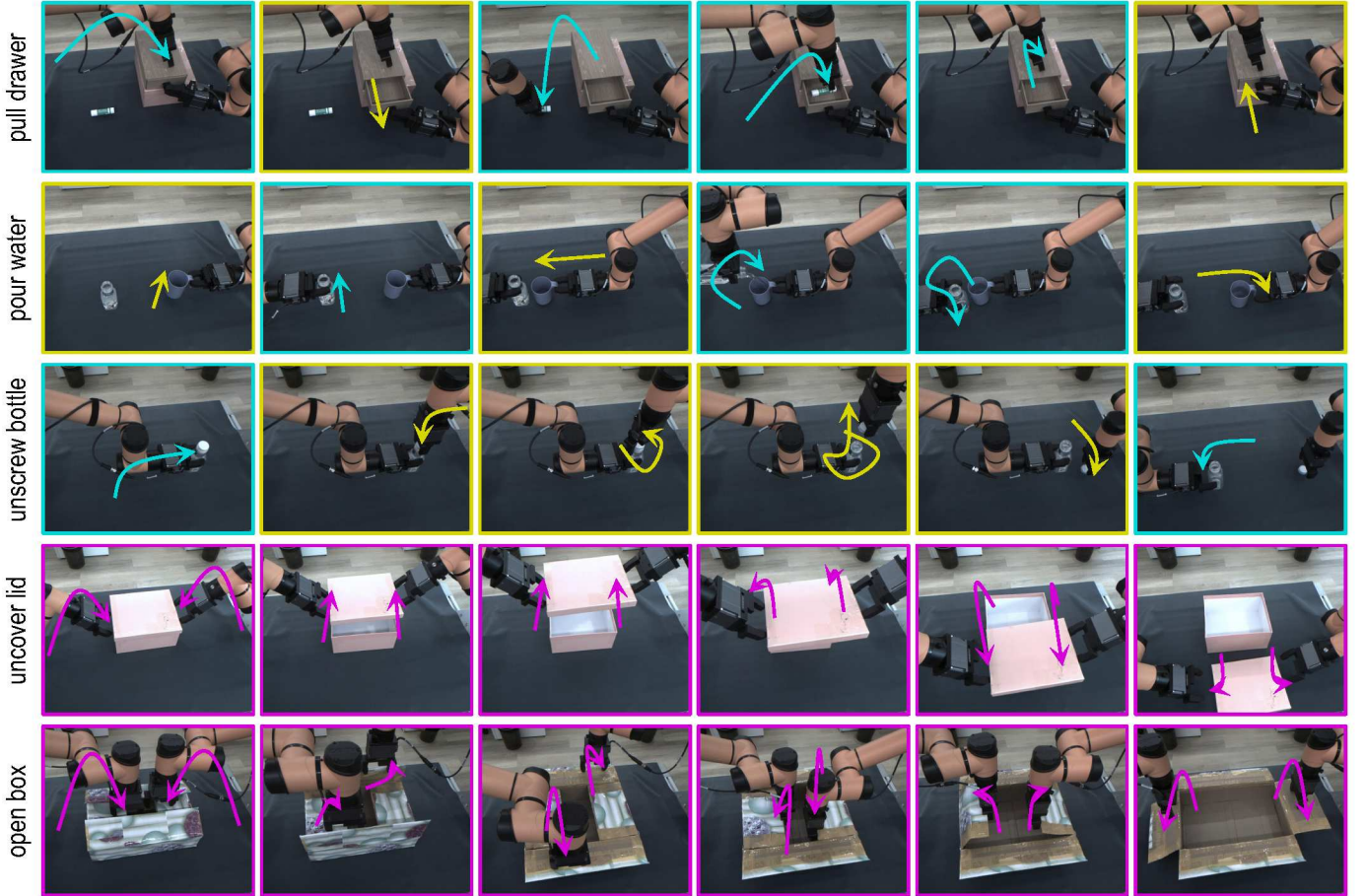


Fig. 7: Visualization of five bimanual tasks performed on real robots. We use different colors such as teal, olive and purple to distinguish frames of left arm, right arm and both arms, respectively. Arrows are artificially added to show movement trends.

of all the strengths and has achieved the best results.

On the other hand, we need to compare and explain whether BiDP is better than other visuomotor imitation methods [104, 15, 98, 95] on more bimanual tasks. As shown in Tab. IV, following the mainstream in-distribution setting, we performed extensive policies training and real robot evaluations on five long-horizon tasks, and reported a detailed performance comparison of various methods. Generally speaking, we can draw three conclusions from these quantitative data. (1) First, the diffusion-based strategy always performed better than the transformer-based ACT. This is mainly because the diffusion model can model a higher-dimensional action space and is highly malleable, while transformer architectures usually do not have these characteristics and require a large amount of data to achieve scale effects and gain advantages. In addition, ACT utilizes 2D images as observations instead of 3D input, which also makes it achieve inferior results. (2) Second, a more advanced and sophisticated 3D observation perception architecture can lead to higher policy performance. For example, compared to the modified DP that directly uses PointNet++ to process 3D point cloud input, DP3 and EquiBot adopt a self-designed lightweight MLP encoder and SIM(3)-equivariant backbone to extract point cloud features, respectively, and always achieved better results. (3) Finally, for more complex

long-horizon bimanual manipulation tasks, the existing state-of-the-art methods still have a lot of room for improvement, such as the gradually decaying effect over multiple substeps and less exploration of efficient utilization of training data. Thanks to the proposed multiple strategies, our BiDP can better cope with bimanual tasks, significantly better than all compared policies. We summarized the average success rate of each method on all five tasks in Tab V, where our method BiDP achieved a success rate of nearly 60%, demonstrating good potential for practical robotic applications. To sum up, it can be concluded that the various strategies we proposed in YOTO are quite effective.

(Q3) **BiDP has satisfactory out-of-domain generalization ability.** To further illustrate the superiority of BiDP, we designed tests under out-of-distribution (OOD) settings. Results are shown in Tab VI. From it, we can see that, except for our method and EquiBot, the performance of the other three methods has dropped significantly when it comes to OOD setups, showing poor generalization to unseen objects. Comparing to EquiBot, our BiDP still has a clear advantage, thanks to the fact that we use explicit 3D geometric transformations for expanding the training demonstrations instead of SIM(3)-equivariant augmentation of the entire point cloud input in EquiBot. In addition, using pure object point clouds as input

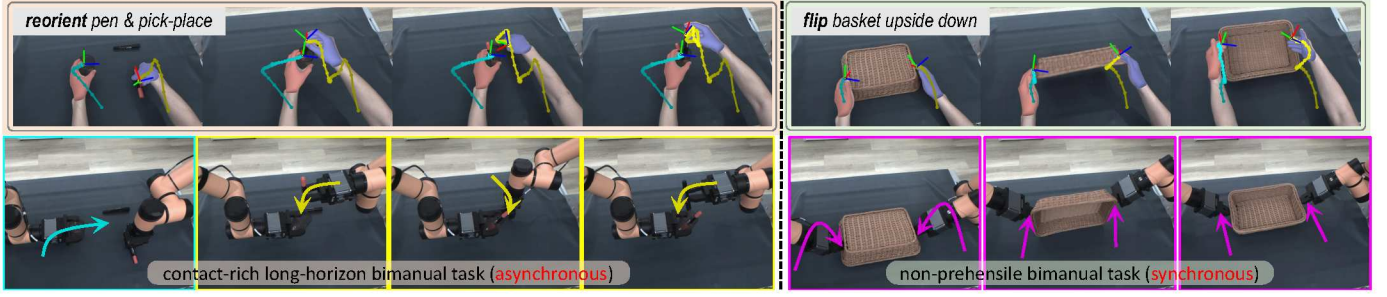


Fig. 8: Illustrations of another two bimanual tasks. **Top**: the visualization of hand motions extraction. **Bottom**: the corresponding rollout examples by injecting actions on real robots. Refer to Fig. 1 and Fig. 7 for notes on different colors and curves.

also makes our model more robust compared to all baselines. The core idea here is to rely on the still rapidly developing capabilities of vision foundation models, such as the open vocabulary detection [90] and segmentation [73], to more reliably perceive various unseen scenes and objects. In summary, these results verify that our BiDP indeed outperforms prior methods with the least amount of performance degradation in OOD generalization.

(Q4) **YOTO is widely applicable to diverse bimanual tasks.** Our proposed framework YOTO is compatible with most bimanual tasks, such as the selected five representative long-horizon tasks, covering a variety of skills, multi-object perception, dual-arm coordinated processing, intricate motion trajectories, and varying execution substeps. In addition to the above-mentioned quantitative results, we also qualitatively demonstrate the visual effects of real robot execution on five tasks in Fig. 7, mainly showing the sparse keyframes contained in them. We can see that the two robot arms have learned the movements demonstrated by human hands and complete these complex tasks in an orderly manner.

Moreover, we selected another two typical bimanual manipulation tasks and enabled the dual-arm robot to learn new given tasks quickly and easily through one-shot human teaching. Due to space limitations, we did not continue the demonstration proliferation and policy training. The illustrations of extracted actions that can be injected into real robots are shown in Fig. 8. These results further reveal the simplicity, versatility and scalability of YOTO. In the future, we will explore using YOTO to handle more intricate, valuable, but less researched bimanual manipulation tasks.

VI. CONCLUSION AND LIMITATION

In this paper, we propose a novel framework named YOTO to address the challenge of efficient and robust bimanual manipulation. Our approach learns from one-shot human video demonstrations, using vision techniques to extract fine-grained and consecutive hand features like pose, joints, and states. To ensure stable and precise manipulation, we simplify noisy hand motion trajectories into discrete keyframes and introduce a motion mask for better dual-arm coordination. Based on the refined one-shot teaching, we develop a scalable data proliferation solution using auto-rollout verification and 3D geometric transformations to rapidly create diverse training

examples. With this enriched dataset, we design a dedicated bimanual diffusion policy (BiDP) that simplifies observations, predicts keyposes, and reorganizes action spaces for efficient training. Validated on five complex bimanual tasks, our framework demonstrates superior performance in both synchronous and asynchronous scenarios. These contributions provide a standardized method for transferring human motions to robots, a scalable approach for data generation, and an effective algorithm for mastering intricate dual-arm tasks, advancing the field of bimanual manipulation.

Limitation: Although YOTO has achieved impressive performance on various long-horizon bimanual manipulation tasks, we conclude that it has at least the following limitations. (1) Our vision-based hand trajectory extraction schemes have inherent errors. This means that we have to check carefully and verify on the real robot whether the extracted position and posture information is reliable, which still requires additional manpower. (2) The primary version of YOTO adopts a fixed workbench, which limits its flexibility and accessibility. In the future, we may consider using mobile bases, such as wheeled carts or multi-legged robots. (3) The equipped parallel gripper is not flexible enough and has limited functionality. Upgrading the end-effector to a multi-fingered dexterous hand or equipping it with force-tactile sensors can make the robot more versatile and powerful. (4) More ultra-difficult bimanual tasks are still under-explored, such as the specialized tool-based manipulation (e.g., picking up a hammer to pound a nail or twisting a screwdriver to tighten a screw), highly dynamic non-quasi-stationary tasks, and friendly interactive collaboration with people. In short, these limitations highlight the need for further innovations to enhance robustness, generalization, and scalability in bimanual robot manipulation.

ACKNOWLEDGMENTS

This work was supported by the Guangdong Provincial Key Field R&D Program (No. 20240104, the project name: Research and Application of Common Key Technologies of Robot Embodied Intelligence Based on AI Large Model), and also received funding from the 2024 Shenzhen Science and Technology Major Project (No. 202402002, the project name: Research and Development of Multimodal Database for Robots to Learn Human Skills).

REFERENCES

- [1] Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debiddatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.
- [2] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken 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.
- [3] Arpit Bahety, Shreeya Jain, Huy Ha, Nathalie Hager, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Bag all you need: Learning a generalizable bagging strategy for heterogeneous objects. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 960–967. IEEE, 2023.
- [4] Arpit Bahety, Priyanka Mandikal, Ben Abbatematteo, and Roberto Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [5] Ravin Balakrishnan and Gordon Kurtenbach. Exploring bimanual camera control and object manipulation in 3d graphics interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 56–62, 1999.
- [6] Christian Bersch, Benjamin Pitzer, and Sören Kammel. Bimanual robotic cloth manipulation for laundry folding. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1413–1419. IEEE, 2011.
- [7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [8] Johann Brehmer, Joey Bose, Pim De Haan, and Taco S Cohen. Edgi: Equivariant diffusion for planning with embodied agents. *Advances in Neural Information Processing Systems*, 36, 2024.
- [9] Alper Canberk, Cheng Chi, Huy Ha, Benjamin Burchfiel, Eric Cousineau, Siyuan Feng, and Shuran Song. Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5872–5879. IEEE, 2023.
- [10] Lawrence Yunliang Chen, Baiyu Shi, Roy Lin, Daniel Seita, Ayah Ahmad, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. Bagging by learning to singulate layers using interactive perception. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3176–3183. IEEE, 2023.
- [11] Lawrence Yunliang Chen, Baiyu Shi, Daniel Seita, Richard Cheng, Thomas Kollar, David Held, and Ken Goldberg. Autobag: Learning to open plastic bags and insert objects. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3918–3925. IEEE, 2023.
- [12] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C Karen Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. *arXiv preprint arXiv:2410.08464*, 2024.
- [13] Yuanpei Chen, Chen Wang, Yaodong Yang, and Karen Liu. Object-centric dexterous manipulation from human motion data. In *8th Annual Conference on Robot Learning*, 2024.
- [14] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. In *8th Annual Conference on Robot Learning*, 2024.
- [15] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.
- [16] Rohan Chitnis, Shubham Tulsiani, Saurabh Gupta, and Abhinav Gupta. Efficient bimanual manipulation using learned task schemas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1149–1155. IEEE, 2020.
- [17] Ian Chuang, Andrew Lee, Dechen Gao, and Iman Soltani. Active vision might be all you need: Exploring active vision in bimanual robotic manipulation. *arXiv preprint arXiv:2409.17435*, 2024.
- [18] Matei Ciocarlie, Corey Goldfeder, and Peter Allen. Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem. In *Proceedings of Robotics: Science and Systems (RSS)*, 2007.
- [19] Adria Colomé and Carme Torras. Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes. *IEEE Transactions on Robotics*, 34(3):602–615, 2018.
- [20] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.
- [21] Michael Drolet, Simon Stepputtis, Siva Kailas, Ajinkya Jain, Jan Peters, Stefan Schaal, and Heni Ben Amor. A comparison of imitation learning algorithms for bimanual manipulation. *IEEE Robotics and Automation Letters*, 2024.
- [22] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer*

- Vision and Pattern Recognition*, pages 12943–12954, 2023.
- [23] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.
 - [24] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.
 - [25] Jianfeng Gao, Xiaoshu Jin, Franziska Krebs, Noémie Jaquier, and Tamim Asfour. Bi-kvil: Keypoints-based visual imitation learning of bimanual manipulation tasks. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16850–16857. IEEE, 2024.
 - [26] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
 - [27] Jennifer Grannen, Priya Sundaresan, Brijen Thananjeyan, Jeffrey Ichnowski, Ashwin Balakrishna, Vainavi Viswanath, Michael Laskey, Joseph Gonzalez, and Ken Goldberg. Untangling dense knots by learning task-relevant keypoints. In *Conference on Robot Learning*, pages 782–800. PMLR, 2021.
 - [28] Jennifer Grannen, Yilin Wu, Suneel Belkhale, and Dorsa Sadigh. Learning bimanual scooping policies for food acquisition. In *Conference on Robot Learning*, pages 1510–1519. PMLR, 2023.
 - [29] Jennifer Grannen, Yilin Wu, Brandon Vu, and Dorsa Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Conference on Robot Learning*, pages 563–576. PMLR, 2023.
 - [30] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
 - [31] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.
 - [32] Markus Grotz, Mohit Shridhar, Tamim Asfour, and Dieter Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks. *arXiv preprint arXiv:2407.00278*, 2024.
 - [33] Valentin N Hartmann, Andreas Orthey, Danny Driess, Ozgur S Oguz, and Marc Toussaint. Long-horizon multi-robot rearrangement planning for construction assembly. *IEEE Transactions on Robotics*, 39(1):239–252, 2022.
 - [34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
 - [35] Paul Hebert, Nicolas Hudson, Jeremy Ma, and Joel W Burdick. Dual arm estimation for coordinated bimanual manipulation. In *2013 IEEE International Conference on Robotics and Automation*, pages 120–125. IEEE, 2013.
 - [36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
 - [37] Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. Dynamic handover: Throw and catch with bimanual hands. In *Conference on Robot Learning*, pages 1887–1902. PMLR, 2023.
 - [38] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.
 - [39] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.
 - [40] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.
 - [41] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.
 - [42] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *8th Annual Conference on Robot Learning*, 2024.
 - [43] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
 - [44] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024.
 - [45] Franziska Krebs and Tamim Asfour. A bimanual manipulation taxonomy. *IEEE Robotics and Automation Letters*, 7(4):11031–11038, 2022.

- [46] Gen Li, Nikolaos Tsagkas, Jifei Song, Ruairidh Mon-Williams, Sethu Vijayakumar, Kun Shao, and Laura Sevilla-Lara. Learning precise affordances from ego-centric videos for robotic manipulation. *arXiv preprint arXiv:2408.10123*, 2024.
- [47] Jinhan Li, Yifeng Zhu, Yuqi Xie, Zhenyu Jiang, Mingyo Seo, Georgios Pavlakos, and Yuke Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation. In *8th Annual Conference on Robot Learning*, 2024.
- [48] Yunfei Li, Chaoyi Pan, Huazhe Xu, Xiaolong Wang, and Yi 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.
- [49] Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.
- [50] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024.
- [51] I-Chun Arthur Liu, Sicheng He, Daniel Seita, and Gaurav S Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation. In *8th Annual Conference on Robot Learning*, 2024.
- [52] Junjia Liu, Yiting Chen, Zhipeng Dong, Shixiong Wang, Sylvain Calinon, Miao Li, and Fei Chen. Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166, 2022.
- [53] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.
- [54] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21740–21751, 2024.
- [55] Xiao Ma, Sumit Patidar, Iain Haughton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18081–18090, 2024.
- [56] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315. IEEE, 2010.
- [57] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.
- [58] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pages 1678–1690. PMLR, 2022.
- [59] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.
- [60] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.
- [61] Seyed Sina Mirrazavi Salehian, Nadia Barbara Figueroa Fernandez, and Aude Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Proceedings of Robotics: Science and Systems (RSS)*, 2016.
- [62] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024.
- [63] Soroush Nasiriany, Sean Kirmani, Tianli Ding, Laura Smith, Yuke Zhu, Danny Driess, Dorsa Sadigh, and Ted Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation. *arXiv preprint arXiv:2411.02704*, 2024.
- [64] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.
- [65] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.
- [66] Georgios Papagiannis, Norman Di Palo, Pietro Vitiello, and Edward Johns. R+x: Retrieval and execution from everyday human videos. *arXiv preprint arXiv:2407.12957*, 2024.
- [67] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2024.
- [68] Angelika Peer, Yuta Komoguchi, and Martin Buss. Towards a mobile haptic interface for bimanual manipulations. In *2007 IEEE/RSJ International Conference on*

- Intelligent Robots and Systems*, pages 384–391. IEEE, 2007.
- [69] Weikun Peng, Jun Lv, Yuwei Zeng, Haonan Chen, Siheng Zhao, Jichen Sun, Cewu Lu, and Lin Shao. Tiebot: Learning to knot a tie from visual demonstration through a real-to-sim-to-real approach. In *8th Annual Conference on Robot Learning*, 2024.
 - [70] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
 - [71] Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. *arXiv preprint arXiv:2409.12259*, 2024.
 - [72] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017.
 - [73] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
 - [74] Javier Romero, Dimitris Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 2017.
 - [75] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, pages 234–241. Springer, 2015.
 - [76] Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chae-won Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18007–18018, 2024.
 - [77] Gautam Salhotra, I-Chun Arthur Liu, and Gaurav S Sukhatme. Learning robot manipulation from cross-morphology demonstration. In *Conference on Robot Learning*, pages 2257–2277. PMLR, 2023.
 - [78] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9869–9878, 2020.
 - [79] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. In *8th Annual Conference on Robot Learning*, 2024.
 - [80] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.
 - [81] Doganay Sirintuna, Idil Ozdamar, and Arash Ajoudani. Carrying the uncarriable: a deformation-agnostic and human-cooperative framework for unwieldy objects using multiple robots. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7497–7503. IEEE, 2023.
 - [82] Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V Dimarogonas, and Danica Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous systems*, 60(10):1340–1353, 2012.
 - [83] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.
 - [84] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.
 - [85] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning*, pages 201–221. PMLR, 2023.
 - [86] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
 - [87] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.
 - [88] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bi-manual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022.
 - [89] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023.
 - [90] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.
 - [91] Fan Xie, Alexander Chowdhury, M De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep imitation learning for bimanual robotic manipulation.

Advances in Neural Information Processing Systems, 33:2327–2337, 2020.

- [92] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023.
- [93] Gangwei Xu, Xianqi Wang, Zhaoxing Zhang, Junda Cheng, Chunyuan Liao, and Xin Yang. Igev++: Iterative multi-range geometry encoding volumes for stereo matching. *arXiv preprint arXiv:2409.00638*, 2024.
- [94] Lei Yan, Theodoros Stouraitis, João Moura, Wenfu Xu, Michael Gienger, and Sethu Vijayakumar. Impact-aware bimanual catching of large-momentum objects. *IEEE Transactions on Robotics*, 2024.
- [95] Jingyun Yang, Ziang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. In *8th Annual Conference on Robot Learning*, 2024.
- [96] Jingyun Yang, Congyue Deng, Jimmy Wu, Rika Antonova, Leonidas Guibas, and Jeannette Bohg. Equivact: Sim (3)-equivariant visuomotor policies beyond rigid object manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9249–9255. IEEE, 2024.
- [97] Zhaodong Yang, Yunhai Han, and Harish Ravichandar. Asymdex: Leveraging asymmetry and relative motion in learning bimanual dexterity. *arXiv preprint arXiv:2411.13020*, 2024.
- [98] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [99] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [100] Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 445–456, 2024.
- [101] Fan Zhang and Michael Gienger. Affordance-based robot manipulation with flow matching. *arXiv preprint arXiv:2409.01083*, 2024.
- [102] Tianle Zhang, Dongjiang Li, Yihang Li, Zecui Zeng, Lin Zhao, Lei Sun, Yue Chen, Xuelong Wei, Yibing Zhan, Lusong Li, et al. Empowering embodied manipulation: A bimanual-mobile robot manipulation dataset for household tasks. *arXiv preprint arXiv:2405.18860*, 2024.
- [103] Xinyu Zhang and Abdeslam Boularias. One-shot im-
- itation learning with invariance matching for robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [104] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- [105] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Seyed Kamyar Seyed Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*, 2024.
- [106] Yan Zhao, Ruihai Wu, Zhehuan Chen, Yourong Zhang, Qingnan Fan, Kaichun Mo, and Hao Dong. Dualafford: Learning collaborative visual affordance for dual-gripper manipulation. In *The Eleventh International Conference on Learning Representations*, 2023.
- [107] Bohan Zhou, Haoqi Yuan, Yuhui Fu, and Zongqing Lu. Learning diverse bimanual dexterous manipulation skills from human demonstrations. *arXiv preprint arXiv:2410.02477*, 2024.
- [108] Jihong Zhu, Michael Gienger, Giovanni Franzese, and Jens Kober. Do you need a hand?—a bimanual robotic dressing assistance scheme. *IEEE Transactions on Robotics*, 40:1906–1919, 2024.

APPENDIX

A. Implementation Details of Our BiDP

In this section, we describe in detail the architecture and implementation of our proposed method BiDP.

1) *Spaces of observation and action*: We adopt a 13-dimensional proprioception vector and a 7-dimensional action space for each robot arm, respectively. The proprioception data for each arm consists of the following information: a 3-dimensional end-effector position, a 6-dimensional vector denoting end-effector orientation (represented by two columns of the end-effector rotation matrix), a 3-dimensional vector indicating the direction of gravity, and a scalar that represents the degree to which the gripper is opened. The action space for each arm consists of the following information: a 3-dimensional vector for the end-effector position offset, a 3-dimensional vector for the end-effector angular velocity in axis-angle format, and a scalar denoting the gripper action.

For all our bimanual tasks, the observation horizon is set to 1, so we only use the initial state observation of the left arm as one of the network inputs. And the initial state of the right arm is always fixed in each task. For the number of the action steps, which is also the length of the predicted horizon, we simplify it and set the prediction length of the three strictly asynchronous tasks to the number of keyframes K , and the prediction length of the two synchronous tasks to $2K$, which is not reducible. This is slightly different from the setup used in the mainstream methods ACT [104], Diffusion Policy [15] and EquiBot [95], where the action horizon is always smaller than the prediction horizon with redundant steps.

2) *Network architecture*: In all tasks, we use a SIM(3)-equivariant PointNet++ [96, 95] with 4 layers and hidden dimensionality 128 as the feature encoder. For the noise prediction network, we inherit hyperparameters from the original Diffusion Policy [15]. Specifically, to optimize for inference speed in all experiments, we use the DDIM scheduler [83] with 8 denoising steps, instead of the DDPM scheduler [36] which performs up to 100 denoising steps.

3) *Sampling of point cloud*: As we all known, setting the number of points to sample in the point cloud observation is a key hyperparameter to consider when designing an architecture that takes point cloud inputs. In our experiments, we found out that using 1024 points is sufficient for all tasks. In particular, we have tried increasing the number of point clouds to 2048 or more, but the evaluation improvement in each task is minimal, and this will also cause the storage occupied by the training observation data to be too large and the training time cost to increase. Therefore, reducing the number of points to 1024 can make training faster without hurting performance. And all our policy models can be trained on a GeForce RTX 3090 Ti with 24 GB of memory.

4) *Training and evaluation*: When using fully expanded training demonstrations (including $\times 100$ and $\times 500$), we train all methods of the first two tasks and the last three tasks for 500 and 1,000 epochs, respectively. The batch-size is set to 64. Otherwise, when using these under-expanded training

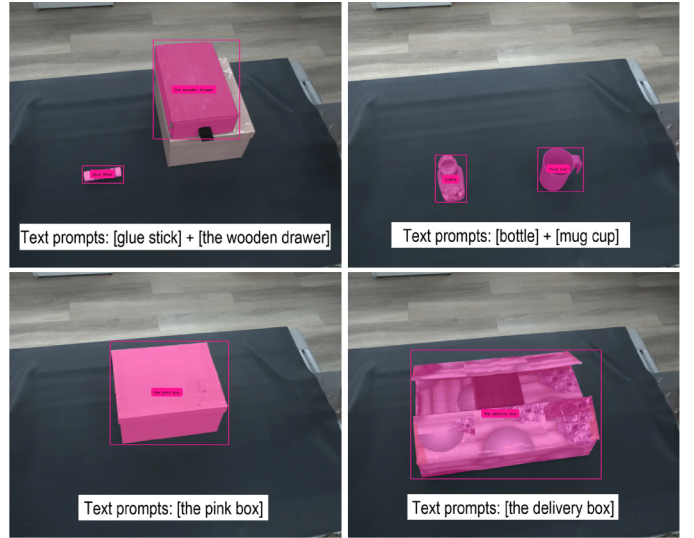


Fig. 9: Examples of using vision foundation models (VFMs) to detect and segment manipulated objects.

data (including $\times 25$ and $\times 5$ and not expanded), we train all methods of the first two tasks and the last three tasks for 2,000 and 4,000 epochs, respectively. For all experiments, we only evaluate the last one checkpoint saved at the end of training. For every evaluation in the real world, we run the policy in a randomly initialized placement of objects for dozens of episodes (please refer the metrics part in the main content for more details), and record the mean average length and success rate achieved by the policy.

In addition, we have explained and demonstrated the importance and advantages of object-centric point cloud input in our main paper. At inference time, we also need to preprocess the binocular RGB observations to obtain the point cloud of manipulated objects. This core design relies on the still rapidly developing capabilities of vision foundation models (VFMs). Here we leverage the state-of-the-art open vocabulary detection method Florence-2 [90] and segmentation method SAM2 [73] to automatically extract object masks and then filter out corresponding point clouds. Examples are shown in Fig. 9. Despite this, occasionally we may fail to segment desired objects accurately, and in these special cases we will manually correct the masks. These cases are not counted as failed evaluation trails due to not involving significant elements of bimanual robot manipulation. Because we believe that the next generation of VFMs can alleviate these problems, or we can directly address them through domain adaptation, test-time adaptation, or adjusting input prompts.

B. Details of Our Selected Bimanual Tasks

We here summarize details related to all manipulation tasks, including object size, hand trajectory visualization, number of keyframes and the valid manipulation area.

1) *Real size of manipulated objects*: In order to give readers a more intuitive cognition of the size of all the objects used in our experiment, we have summarized the centimeter-level shape information of the objects involved in detail in Fig. 10.



Fig. 10: We collected a variety of manipulated objects in instance-level for each of five bimanual tasks to improve and verify the generalizability of trained policies. All of these objects are from everyday life, not intentionally customized. We also collect the detailed size information in centimeter-level for all related objects. Best to view after zooming in.

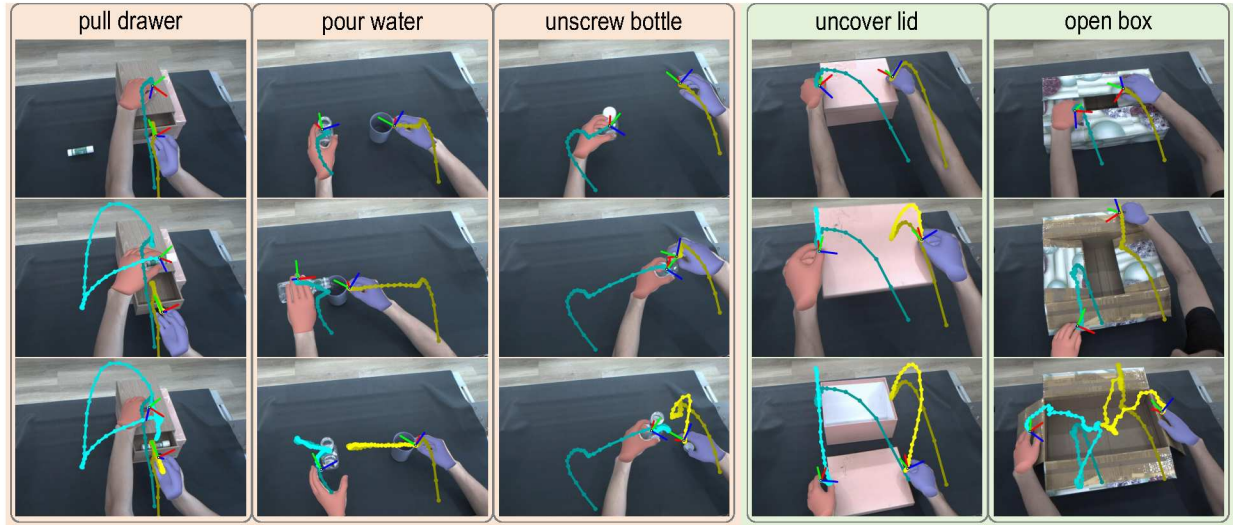


Fig. 11: Visualization of extracted hand trajectories for five long-horizon bimanual tasks. Best to view after zooming in.

Without loss of generality, we roughly abstract all objects into two typical geometric shapes, namely **cuboids** (represented by length, width and height) and **cylinders** (represented by height and diameter). The cuboids include 9 everyday objects and 3 drawers in the pull drawer task, 5 covered boxes in the uncover lid task, and 4 express boxes in the open box task. The cylinders include the 6 bottles and 3 mugs in the pour water task, and the 6 caps in the unscrew bottle task. In particular, we also counted the height of the lid in task uncover lid and the length of the flippable wings on both

sides in task open box. These objects are all within the size range that the gripper can grasp and the operating table can place. We expect that these detailed statistics can help better understand and reproduce each task.

2) *Hand motion extraction results:* The movement mode of all bimanual tasks we designed mainly comes from a single-shot teaching of both human hands. In Fig. 11, we show the detailed visualization results of extracted hand motion trajectories so that we can better understand the entire process of manipulation, including which objects each arm contacts

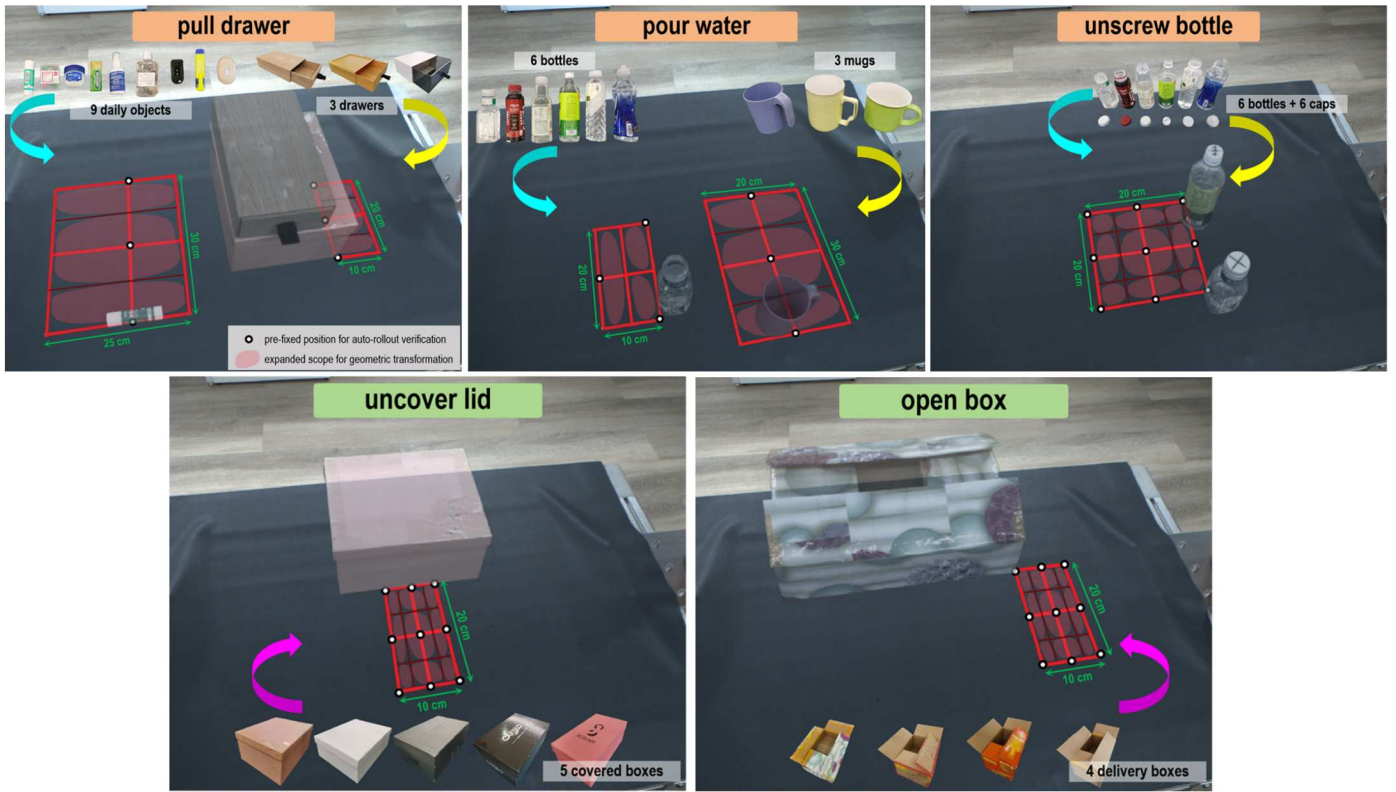


Fig. 12: Illustration of the effective placement range of manipulated objects on the workbench for each task during training and testing. We mainly show the predefined position points of the objects in the automatic verification phase (indicating by black-white dots), as well as the approximate distribution of the object positions after using the geometry transformation (indicating by spread translucent red scopes). Best to view after zooming in.

and the order of motion. Our subsequent demonstration proliferation and learned action prediction policies will follow the same motion pattern.

3) *Determination of keyframes in each task:* As described in the main text, we use discrete keyframes (*a.k.a.* keyposes) to simplify and represent each long-horizon task as in C2FARM [38] and PerAct [80]. Keyframes can be auto-extracted using simple heuristics, such as a change in the open/close end-effector state or local extrema of velocity/acceleration. This abstraction way is extremely effective for the first three strictly asynchronous tasks, but the latter two tasks that require the synchronization of both arms do not perform well due to very few remaining keyframes. Therefore, we artificially increase sampling frames between keyframes with larger step spans to make the manipulation action more stable and smooth. This constraint can also be added to heuristic rules to complete automatic keyframes extraction.

4) *Effective area on the workbench:* Since we used two fixed manipulators, the accessible space is limited. Specifically, as shown in Fig. 12, we marked the distribution of the effective areas where the objects were located in each task on the table platform. In the training demonstrations we collected and the real robot evaluation phase, we would not place objects outside these areas to avoid exceeding the reach limit of two arms. Note that this does not mean that the policies we trained do not generalize to different locations. Even in a restricted

area, the position and orientation of the manipulated objects during testing may be completely new, so it is still a non-trivial out-of-distribution (OOD) situation.

In Fig. 12, we use a series of black-white dots to represent each pre-fixed point, which corresponds to the position of the manipulated object in the auto-rollout phase. These points are defined relative to the area where the object touches the table, and can be at its geometric center (`open box`), the midpoint of a side (`uncover lid`), or a corner in a fixed direction (`open box`, `pour water`, `unscrew bottle` and `open box`). We show some examples of semi-transparent manipulated objects in each sub-image. The effective range drawn on the table also refers to these proxy points. Therefore, considering that the object itself may be quite large (refer Fig. 10), the area it actually covers will be much larger. On the other hand, we utilize the semi-transparent red area to indicate the approximate distribution of objects after their positions have been modified using the point cloud-based geometric transformation. We apply measured parameters to control the augmented objects to basically cover the entire valid area without overlapping, so that the trained model has robust generalization of position variations. Moreover, this design ensures that the absolute displacement of the object will not be too large, thus avoiding obvious violations of the imaging rules under perspective projection.

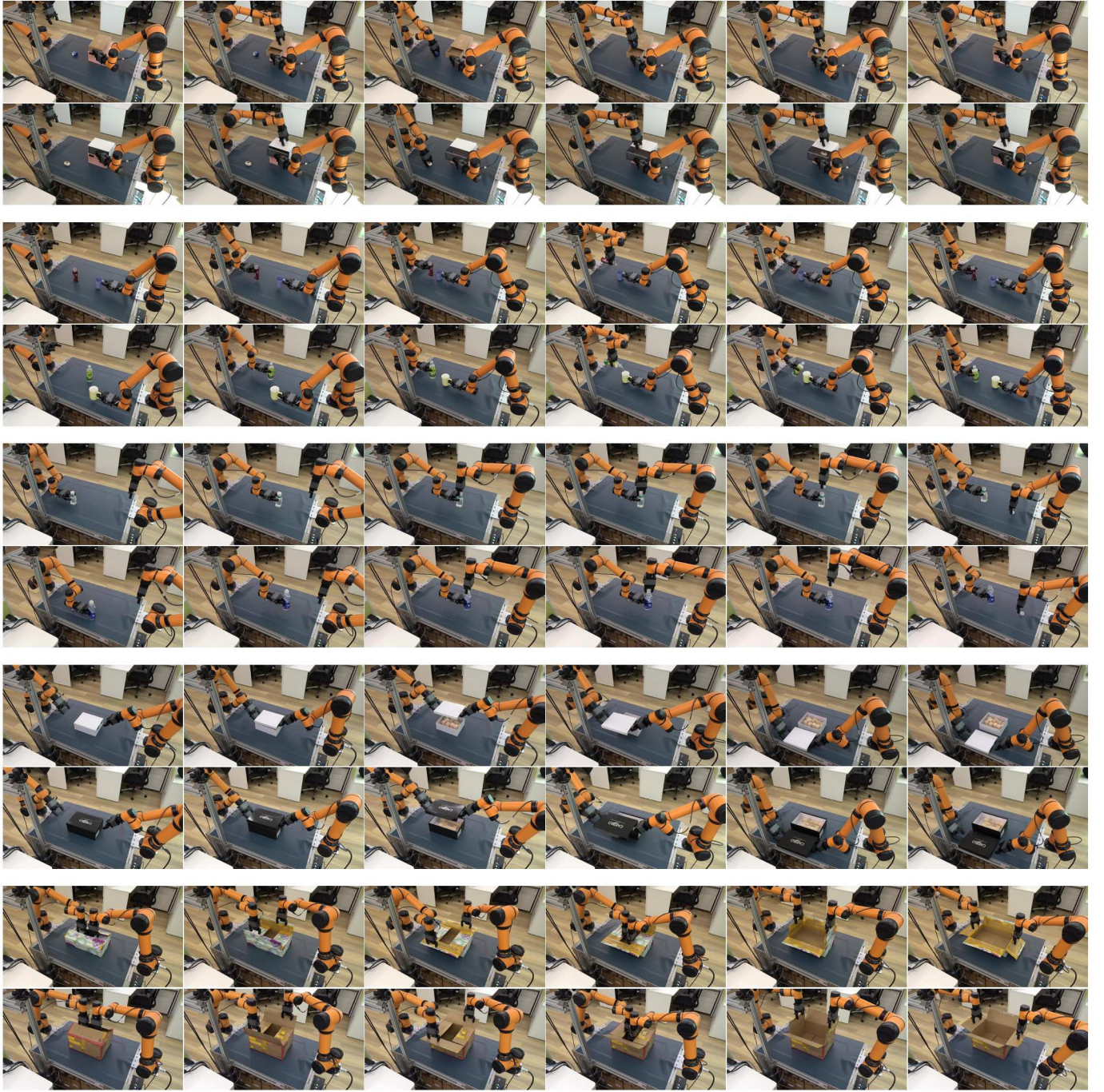


Fig. 13: Qualitative rollout samples from the third-person perspective for all real robot evaluation scenarios. From top to bottom, they are the five long-horizon bimanual tasks mentioned in the main paper. Best to view after zooming in.

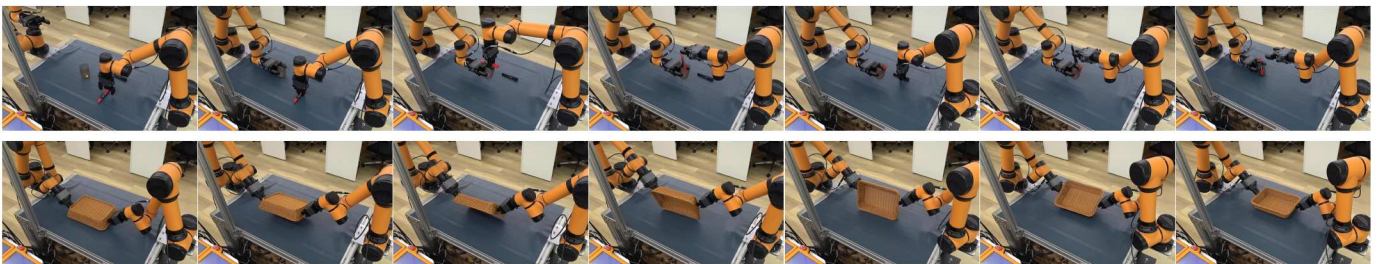


Fig. 14: Rollout examples of another two new tasks including `reorient pen` (top row) and `flip basket` (bottom row).

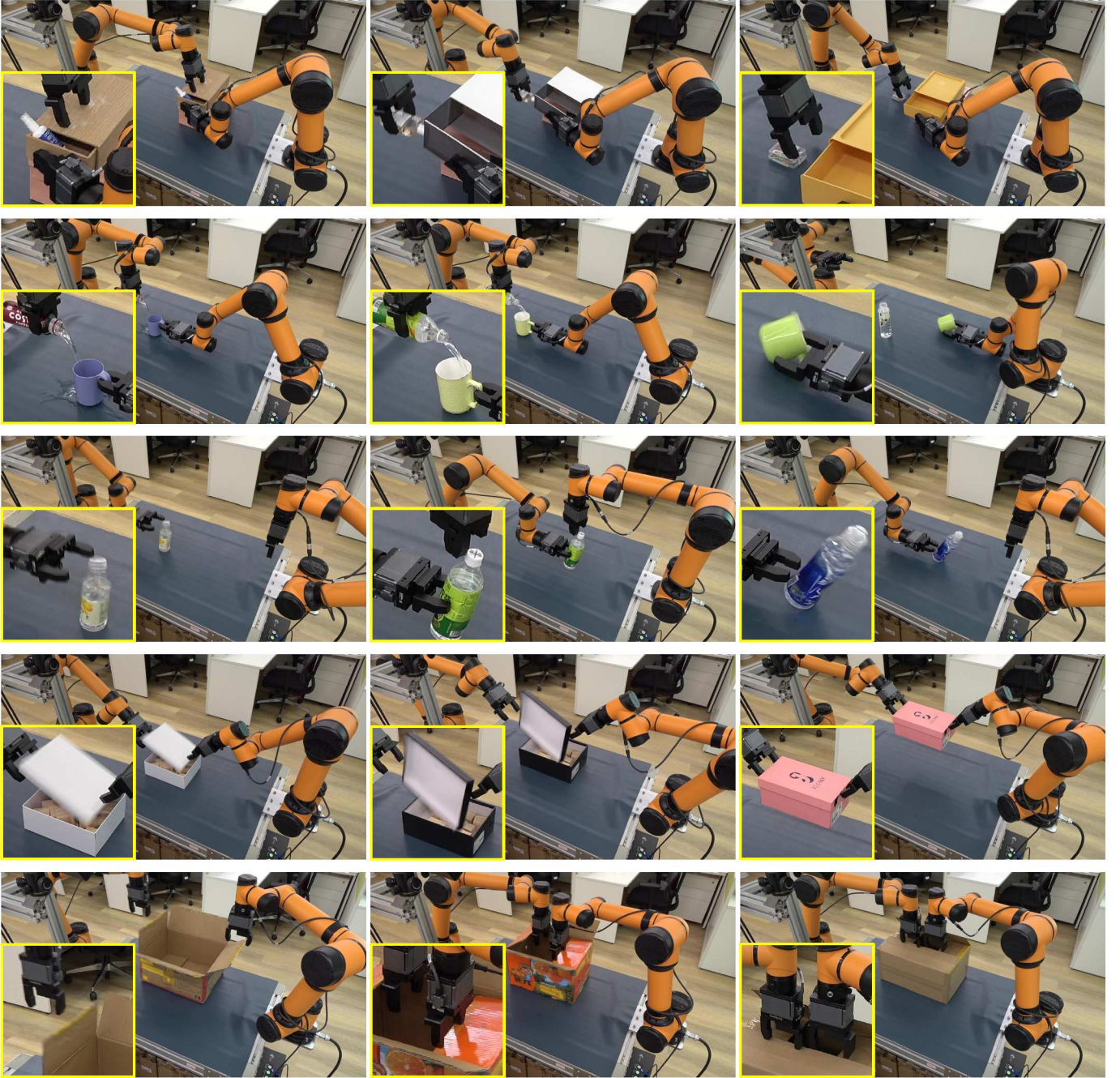


Fig. 15: From top to bottom, we have examples of failed cases in all five tasks during evaluation. We have outlined and magnified the areas where the failures occurred so that we can quickly examine them. Best to view after zooming in.

C. Evaluation Results and Performance Analysis

1) *More qualitative examples:* In Fig. 13, we show qualitative rollout samples from the third-person perspective for all evaluation tasks we mentioned in the main paper. We still choose images near the keyframes to illustrate more effectively, e.g., the model’s generalization to object *category* and *location* variations. These examples show more complete scenes and the motion of two robot arms, and can be considered as a supplement to the limited field of view of the binocular observation camera. Note that these third-person

video recordings do not participate in any training and testing.

Specifically, from top to bottom in Fig. 13, we have: (1.1) open the *gray wooden drawer*, pick up the *vaseline box* and put it into the drawer, and close the drawer; (1.2) open the *white plastic drawer*, pick up the *headphone case* and put it into the drawer, and close the drawer; (2.1)&(2.2) grasp and pick up the *mug cup*, grasp and pick up the *drink bottle*, pour the water contained in the bottle into the mug cup, and place back the mug cup and bottle; (3.1)&(3.2) grasp and pick up the *drink bottle*, unscrew the *circular cap* of the bottle, and

place back the bottle and cap; (4.1)&(4.2) use two arms to close to the lid of the *covered box*, lift up the box lid, and place down the box lid; (5.1)&(5.2) use two arms to close to the two upper longer wings of the *delivery box*, open the wings, close to the another two lower shorter wings, and open the wings. For more intuitive qualitative results, please refer to the recorded videos.

2) *Rollouts of two new tasks*: We show in the main paper two additional tasks that can be injected into dual-arm manipulators after a single-shot teaching following the proposed YOTO paradigm. Fig. 14 shows a series of third-person recordings of real rollouts. They involve two important atomic skills: *reorientation* and *rearrangement*. The two tasks are: (1) *reorient pen*. Pick up two pens placed in different directions and place them in a cup. (2) *flip basket*. Flip the non-prehensile woven basket with the mouth facing down 180 degrees so that it faces upwards.

3) *Analysis of failure cases*: Although our method BiDP outperforms many strong baselines [104, 15, 98, 95] for addressing long-horizon bimanual manipulation tasks, it still presents various failure cases during evaluation. Below, we focus our analysis on execution failure of our BiDP in real-world experiments. In Fig. 15, we show some representative failure examples of all real robot executions we have performed with our method.

Specifically, from top to bottom, we have collected failed cases like: (1.1) The object was wrongly placed in the drawer so that the drawer could not be closed successfully; (1.2) The object collided with the drawer during the transfer process, causing the drawer to move out of position and affecting its closing operation; (1.3) A grasping error occurred during the object picking process, causing the object to fail to be picked up; (2.1) The mouth of the bottle is not aligned with the mouth of the mug cup, causing more water to spill out; (2.2) The contact point was biased when grasping the mug handler, making it easy for water to spill out; (2.3) A grabbing error occurred while picking up the mug causing it to fall over; (3.1) When picking up the bottle, the gripper squeezed the bottle cap, causing the grip to fail; (3.2) The gripper fails to clamp the center of the bottle cap, causing the cap to fail to be twisted off; (3.3) When picking up the bottle, the gripper collided with the bottle body, causing the bottle to fall and the gripping failed; (4.1) The box lid fell off due to lack of coordination when opening it; (4.2) The box lid fell off due to lack of coordination when transferring it. (4.3) Inappropriate distance between arms caused that the entire covered box was lifted and moved without opening the box lid; (5.1) After the delivery box was fully opened, due to a defect in the pressing action (such as not long enough or deep enough), a wing rebounded back due to the tension at the hinges; (5.2) Due to the delivery box displacement or inaccurate action prediction, a lower shorter wing could not be successfully opened. (5.3) Due to inaccurate action prediction, a upper longer wing failed to open successfully. These failure cases point out the direction that needs further exploration in the future. For more intuitive

qualitative results, please refer to our recorded videos.