YOTO++: Learning Long-Horizon Closed-Loop Bimanual Manipulation from One-Shot Human Video Demonstrations

Huayi Zhou, *Member, IEEE,* Ruixiang Wang, Yunxin Tai, Yueci Deng, Guiliang Liu, *Member, IEEE,* Kui Jia, *Member, IEEE*

***Abstract*—Bimanual robotic manipulation remains a fundamental challenge due to the inherent complexity of dual-arm coordination and high-dimensional action spaces. This paper presents the entended YOTO++ (You Only Teach Once), which is a unified one-shot learning framework for teaching bimanual skills directly from third-person human video demonstrations. Our method extracts structured 3D hand motions using binocular vision and distills them into compact, keyframe-based trajectories for dual-arm execution. We develop a scalable demonstration proliferation strategy that synthetically augments one-shot demonstrations into diverse training samples, enabling effective learning of a customized bimanual diffusion policy. Extensive evaluations across a broad spectrum of long-horizon bimanual tasks, including asynchronous, synchronous, contact-rich, and non-prehensile scenarios, demonstrate strong generalization to novel skills and objects. We further introduce a visual alignment mechanism at the initial manipulation stage for closed-loop control, enabling the system to dynamically adapt to perturbations during execution. We validate the framework on a new dual-arm robotic platform to show seamless cross-embodiment transfer without additional retraining. YOTO++ achieves impressive performance in accuracy, robustness, and scalability, advancing the practical deployment of general-purpose bimanual manipulation systems. The project link is https://hnuzhy.github.io/projects/YOTOPlus.**

***Index Terms*—Bimanual robotic manipulation, one-shot imitation learning, human demonstration, hand movements.**

I. Introduction

BIMANUAL manipulation is an enduring topic in the robotics community [1]–[5]. 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 [6], [7]. 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 [6]–[8] and implicitly learning from demonstrations collected by teleoperation [9]–[11]. 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 [12]–[17]. 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. 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 [18]–[20]. 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 [21], [22], 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

H. Zhou, G. Liu and K. Jia (corresponding author) are with School of Data Science, The Chinese University of Hong Kong, Shenzhen (the corresponding e-mail: kuijia@cuhk.edu.cn). R. Wang is with Harbin Institute of Technology, Weihai. Y. Tai and Y. Deng are with DexForce, Shenzhen.

This work was supported by the Guangdong Provincial Key Field R&D Program (Project No. 20240104), and was also funded by the Shenzhen Science and Technology Major Project in 2024 (Project No. 202402002).

Manuscript received April 19, 2025; revised August 16, 2025.

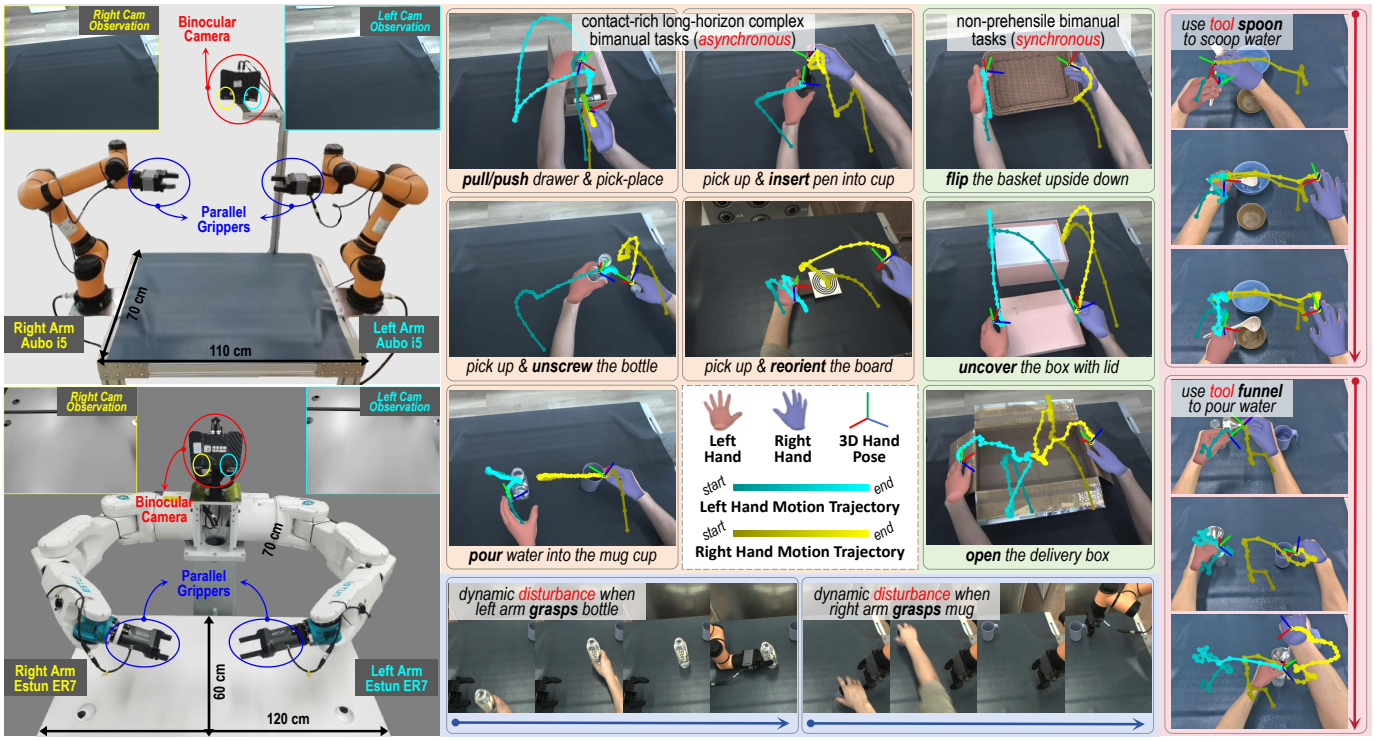


Fig. 1. Our proposed **YOTO++** (**Y**ou **O**nly **T**each **O**nce) enables cross-embodiment deployment (from the contralateral to humanoid dual-arm setups), and facilitates diverse bimanual tasks including asynchronous, synchronous and tool-using scenarios, with closed-loop control under dynamic disturbances during pre-grasping. Notably, it needs only the one-shot observation of a 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.

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 more stable and faster than teleoperation [9], [23]. For example, we can collect about 300 demonstrations in about 6 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 [24], [25] and binocular stereo matching [26], [27]. 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 [28]–[30] 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 replace observations (*e.g.*, 3D point clouds) from the entire scene to manipulated objects to accelerate training convergence and eliminate irrelevant terms [7], [31]. Then, instead of modeling continuous actions, we choose to predict essential keyposes [32]–[35], which can greatly reduce the diffusion space dimensionality. Third, we utilize the motion mask to determine 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 various challenging tasks. Overall, we have the following 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.

YOTO++ is an expansion of our previous conference work YOTO [36], in which we introduce several new components to enhance its functionality and comprehensiveness. Specifically, we have the following new contributions.

- YOTO++ incorporates a vision-based pre-grasping alignment module of each task, enabling closed-loop control that can robustly handle dynamic disturbances.
- YOTO++ can handle long-horizon contact-rich tasks, such as tool-use scenarios, by extracting and executing temporally consistent multi-stage actions through discretized yet semantically essential keyframes.
- YOTO++ supports a richer set of primitive skills, demonstrating its compatibility with up to 10 diverse bimanual manipulation tasks, covering both synchronous and asynchronous coordination patterns.
- The cross-embodiment adaptability of YOTO++ is validated by deploying it on a humanoid dual-arm robot, showcasing its platform-agnostic nature and real-world applicability across diverse robotic morphologies.

II. Related Works

Bimanual Robotic Manipulation. Many bimanual manipulation methods focus on specialized tasks or primitive skills, such as cloth-folding [37]–[39], bagging [40], [41], handover [42]–[44], untwisting [45] and dressing [46]. For general bimanual manipulation, typical research [1], [8], [47]–[50] 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 [6], [51] or stabilizer-actor [7], [52] pair. Most recently, the ALOHA series [9], [53]–[55] have revolutionized bimanual manipulation by dexterous teleoperating and upgrading low-cost hardwares of real-world robotics. These similar works [9], [11], [56]–[58] 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 [23], [59]–[62], mobile footplates [30], [53], [63], tactile feedbacks [59], [64], [65] or active cameras [66], [67]. 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.

Learn from Human Hand Videos. Human hand videos are valuable resources for learning complex manipulation behaviors [68]–[72]. Extensive research has leveraged human demonstrations to learn robot manipulation by extracting rich non-privileged features, such as keypoints [13], [73], [74], affordances [75]–[77], 3D hand poses [12], [15], motion trajectories [14], [15], [17], invariant correspondences [16], [78], [79] and hand-object interaction [80]–[82]. These features can be tailored to robot-specific variables to alleviate morphology gaps, such as manipulation plans, retargeted motions and precise actions. Two contemporary works [15], [17] also propose to use a single human demonstration to learn bimanual manipulation similar to us. RSRD [17] 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 [15] 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.

Visuomotor Imitation Learning. Visuomotor imitation learning aims to train action prediction policies based on visual observations by exploiting labeled demonstrations [21], [22], [83]–[85]. These learned policies can drive robots to complete various manipulation with just dozens of demonstrations, covering dexterous [23], [29] and bimanual [9], [60] tasks. Especially, ALOHA [9] introduced the action chunking transformers (ACT) to learn high-frequency controls with closed-loop feedback in an end-to-end manner. DP [28] adopted conditional denoising diffusion models [86]–[88] to represent visuomotor policies in robotics, exhibiting impressive training stability in modeling high-dimensional action distributions. DP3 [29] incorporated 3D conditioning into the original diffusion policy [28], rather than focusing on RGB images and states as conditions. EquiBot [30] combined SIM(3)-equivariance [89]–[91] with diffusion policy, acquiring a more generalizable and sample-efficient visuomotor policy than [28], [29]. 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 [12], [13], [15]–[17] or two ipsilateral arms [13] to build workstations. Some bimanual teleoperations also tend to be anthropopathic [59]–[61], [67] or ipsilateral [23], [64], [65]. 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 [9], [53]–[55] and its followers [11], [58], [66]. This heterolateral setup minimizes the overlap of accessible space and is thus compatible with a wider range of bimanual tasks. As shown in Fig. 1 left, we adopt the contralateral placement, where each arm (Aubo-i5[1]) has a span of 880 mm. In addition, we also arrange a humanoid dual-arm robot for cross-embodiment testing, where each arm (Estun ER7[2]) and has a span of 910 mm.

End Effector Selection: Although some methods utilize multi-fingered dexterous hands as end effectors [23], [60], [61], [67] and even add tactile sensors [59], [64], [65] to hands, we use two parallel-jaw grippers (with max opening distance 80 mm of each DH-Robotics[3] for Aubo i5, or 75 mm of each Jodell RG75[4] for Estun ER7), which are easier to control and interpret. We will show that it is sufficient to complete complex tasks that are inherently non-prehensile or synchronous.

Camera Observation: Many previous methods adopt the multi-view RGB observations [9], [11], [58], 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 [12], [13], [15], [23] is also acceptable. We use a binocular stereo camera (the DexSense 3D industrial camera[5]), similar to commercial RGB-D cameras, but providing raw left and right images to enable flexible post-processing.

IV. Method

In this part, we introduce in detail the proposed framework YOTO++, which contains three major modules that are 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

[1]https://www.aubo-cobot.com/public/i5product3
[2]https://en.estun.com/?list_110/1766.html
[3]https://en.dh-robotics.com/product/pgi
[4]https://www.jodell-robotics.com/product-detail?id=5
[5]https://dexforce-3dvision.com/productinfo/1022811.html

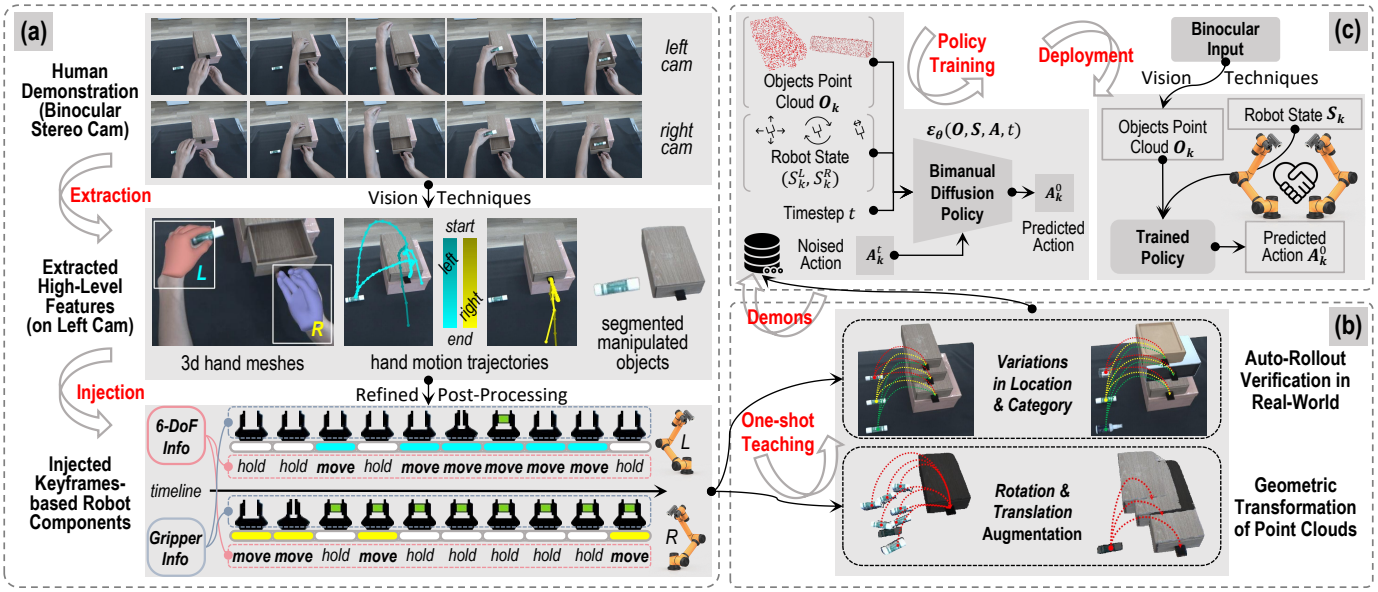


Fig. 2. The overview of our proposed YOTO++. It is a general framework consists of three main modules: **(a)** the human hand motion extraction and injection, **(b)** the training demonstration proliferation from one-shot teaching, and **(c)** the training and deployment of a customized bimanual diffusion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

policy (BiDP) method in Sec. IV-D. Finally, in Sec. IV-E, we explain how to incorporate a visual alignment mechanism for initial grasping to achieve closed-loop control of YOTO++.

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:

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

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 [26] by at least half.

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 [18] 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 [92]. 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 [15], [17], [61], [67] 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 [18] and

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 $\mathbf{P}_j^{\diamond} \leftarrow \textbf{MANO}(\mathcal{H}_j^{\diamond}, I_{\text{hand}})$; // 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
$\bar{v}_z \leftarrow v_z / (\text{NORMALIZE}(v_z) + 1\text{e-}8)$; // vector normalization
$v_y = l_{\text{mid}} \leftarrow (l_{\text{iw}} + l_{\text{rw}})/2.0$; // middle line (Y-axis direction)
$\bar{v}_y \leftarrow v_y / (\text{NORMALIZE}(v_y) + 1\text{e-}8)$; // vector normalization
$\bar{v}_x \leftarrow \text{CROSS\_PRODUCT}(\bar{v}_y, \bar{v}_z)$; // X-axis direction
$v_{rot} \leftarrow \text{CONCATENATE}([\bar{v}_x, \bar{v}_y, \bar{v}_z])$; // final 3×3 rotation matrix
return v_{rot};

HaMeR [20], still cannot achieve continuous and consistent prediction in a given camera space. This is also pointed out and verified by DexCap [23]. More examples can be found in Fig. 6. 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 [26]. The final back-projected 3D point trajectories are $\{\widehat{h}_j^{p,\diamond}\}_{j=1}^{J}$, 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 [19]. 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 [93], [94]. We summarize this process in Alg. 1. To this point, we have obtained the rough motion trajectories purely based on human hand videos $\{(\widehat{h}_j^{p,\diamond}, h_j^{r,\diamond}, h_j^{g,\diamond})\}_{j=1}^{J}$.

Additionally, we adopt cutting-edge vision algorithms (including the vision-language model Florence-2 [24] and SAM2 [25]) to extract segmented manipulated objects from the left initial image as our disturbance-free visual observations $\widehat{O}$, which will be further lifted to 3D point clouds $\widetilde{O}$ by applying stereo matching approaches [26], [27].

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 [21] and PerAct [22]. Keyframes are important intermediate end-effector poses that summarize a demonstration and can be

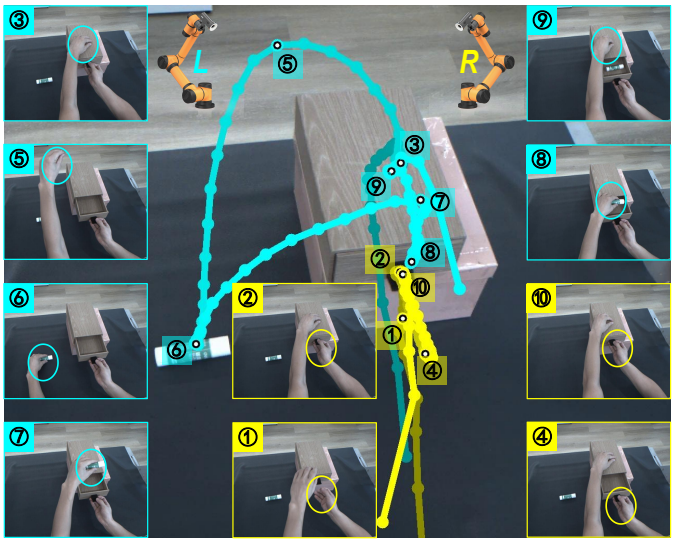


Fig. 3. Example of extracted trajectories with corresponding keyframes of both left hand and right hand. It is best to zoom in to view the details.

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 [32]–[35]. 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 $\{(\widehat{h}_j^{p,\diamond}, h_j^{r,\diamond}, h_j^{g,\diamond})\}_{j=1}^{J}$ into a set of keyframes $\{(\widetilde{h}_k^{p,\diamond}, \widetilde{h}_k^{r,\diamond}, \widetilde{h}_k^{g,\diamond})\}_{k=1}^{K}$, 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 two or three minutes. We finally update the verified trajectories into $\widetilde{\mathbf{A}} = \{(\widetilde{a}_k^{p,\diamond}, \widetilde{a}_k^{r,\diamond}, \widetilde{a}_k^{g,\diamond})\}_{k=1}^{K}$, 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 $\widetilde{\mathbf{A}}$ 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 $\widetilde{\mathbf{A}}$ 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 $\widetilde{\mathbf{A}}$ are interpretable and editable. These properties assist us to conduct automated demonstration rollout verification and collection on real robots. First, we can easily split $\widetilde{\mathbf{A}}$ into two distinctive trajectories $\widetilde{\mathbf{A}}^L$ and $\widetilde{\mathbf{A}}^R$ belonging to the left and right robotic arms based on the motion mask $\mathbf{C}$. Below is for decomposing strictly asynchronous tasks.

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

where we actually eliminate K redundant keyposes for unilateral arm waiting (holding on actions). For synchronous tasks ($|\widetilde{\mathbf{A}}^L| = |\widetilde{\mathbf{A}}^R| = K$), 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 $\widetilde{\mathbf{A}}^L$ or $\widetilde{\mathbf{A}}^R$ closely related to the manipulated object to align with its changed 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 $\widetilde{a}_6^L = (\widetilde{a}_6^{p,L}, \widetilde{a}_6^{r,L}, \widetilde{a}_6^{g,L})$. 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 $\widetilde{a}_6^{p,L}$. 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 $\widetilde{O}$) 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 $\widetilde{O}$ and keyframes-based actions $\widetilde{\mathbf{A}}$ 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 $\widetilde{O}$ 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.

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

D. Bimanual Diffusion Policy Learning

In this part, we adapt popular visuomotor diffusion policies [28]–[30], 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 $\widetilde{\mathcal{D}} = \{(\widetilde{\mathbf{O}}, \widetilde{\mathbf{A}}, \mathbf{C})_i\}_{i=1}^N$, where N is the number of demonstrations. $\mathbf{C}$ is the motion mask containing coordination strategies. $\widetilde{\mathcal{D}}$ 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 $\widetilde{\mathbf{O}} = \{\widetilde{O}_k, S_k^L, S_k^R\}_{k=1}^K$ and $\widetilde{\mathbf{A}} = \{(\widetilde{a}_k^{p,\diamond}, \widetilde{a}_k^{r,\diamond}, \widetilde{a}_k^{g,\diamond})\}_{k=1}^K$, where $\widetilde{O}_k$ is the observation containing 3D point clouds of manipulated objects instead of the entire RGB image [28] or point clouds scene [29], [30]. S_k^L and S_k^R are robot proprioception states with similar formats as actions $S_k^\diamond = (\widetilde{s}_k^{p,\diamond}, \widetilde{s}_k^{r,\diamond}, \widetilde{s}_k^{g,\diamond})$. $\widetilde{\mathbf{A}}$ have discrete keyposes, rather than continuous and dense robot states. Learning to predict keyposes is common in robotic manipulation [32]–[35]. The policy needs to learn a mapping from the initial observation $\widetilde{O}_1$ to all subsequent keyposes $\widetilde{\mathbf{A}}$ 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 $\{\widetilde{O}_1, S_1^L, S_1^R\}$ at first sight.

Diffusion-based policy representation. Similar to [28], [29], we utilize Denoising Diffusion Probabilistic Models (DDPMs) [86] to model the conditional distribution $p(\widetilde{\mathbf{A}}_k | \widetilde{\mathbf{O}}_k)$. Starting from the random Gaussian noise $\widetilde{\mathbf{A}}_k^T$, where T means diffusion steps, DDPM performs T iterations of denoising to predict actions with decreasing levels of noise, gradually from $\widetilde{\mathbf{A}}_k^{T-1}$ to $\widetilde{\mathbf{A}}_k^0$. This process follows:

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

The policy finally outputs $\widetilde{\mathbf{A}}_k^0$. Because point clouds are used as the visual input instead of RGB images, we adopt more robust SIM(3)-equivariant architectures [30], [89], rather than policies based on CNNs [28] or transformers [29]. Formally, the noise prediction network ε_θ takes observation $\widetilde{\mathbf{O}}_k$, noisy action $\widetilde{\mathbf{A}}_k$ and diffusion timestep t as input, and predicts the gradient $\triangledown\mathbf{E}(\widetilde{\mathbf{A}}_k)$ for denoising the noisy action input. It first uses a modified PointNet-based [95] encoder with SIM(3)-equivariance to encode visual observations. The encoded visual features and positional embeddings of t are passed to FiLM layers [96]. Then, the policy network applies a convolutional U-Net [97] to process $\widetilde{\mathbf{A}}_k$, t and the conditioned observations to predict denoising gradients. Note that $\widetilde{\mathbf{O}}_k$, $\widetilde{\mathbf{A}}_k$ and $\widetilde{\mathbf{A}}_k^0$ 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 [30], [89] for more details.

Customized bimanual diffusion policy. Since $\widetilde{\mathbf{A}}_k$ 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 $(\widetilde{a}_k^{p,L}, \widetilde{a}_k^{r,L}, \widetilde{a}_k^{g,L})$ and $(\widetilde{a}_k^{p,R}, \widetilde{a}_k^{r,R}, \widetilde{a}_k^{g,R})$. 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), which is inefficient and error-prone. To this end, we reorganize the action space into $\overline{\mathbf{A}} = \{\widetilde{\mathbf{A}}^L, \widetilde{\mathbf{A}}^R\}$ based on the motion mask $\mathbf{C}$ according to Eqn. 2. $\overline{\mathbf{A}}$ 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 $\{\widetilde{A}_1^R, \widetilde{A}_2^R, \widetilde{A}_3^L, \widetilde{A}_4^R, \widetilde{A}_5^L, \widetilde{A}_6^L, \widetilde{A}_7^L, \widetilde{A}_8^L, \widetilde{A}_9^L, \widetilde{A}_{10}^R\}$. 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.

E. Visual Alignment for Pre-Grasping

In the deployment stage, once the trained bimanual diffusion policy (BiDP) predicts a sequence of keyframe-based actions conditioned on the initial observation $\widetilde{O}_1$, the system is capable of completing a full manipulation task in an *open-loop* fashion. That is, the robot executes pre-computed action sequence $\widetilde{\mathbf{A}} = \{\widetilde{a}_k^\diamond\}_{k=1}^K$ without further feedback from the environment. While this paradigm is efficient, it lacks robustness in dynamic or perturbed settings, particularly during the initial grasping phase when object-robot interaction has not yet been physically established. In contrast, fully *closed-loop* control strategies (such as ACT [9] or Diffusion Policy [28]) employ recurrent observe-infer-act cycles at every time step. However, applying such feedback at all stages of long-horizon tasks can be redundant and computationally demanding. We observe that once the target object has been securely grasped, the relative pose between the end-effector and object becomes fixed, reducing the necessity for high-frequency visual feedback. In this case, it is both safe and efficient to rely on either the initial demonstration-aligned keyframes or model-inferred trajectories for the subsequent execution.

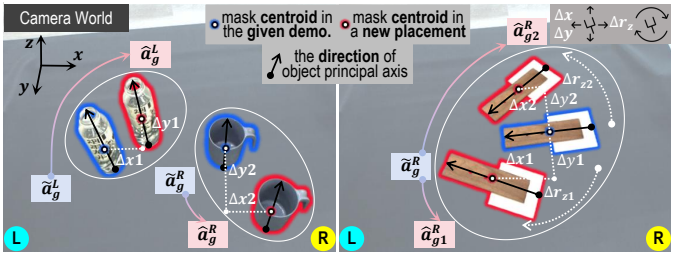


Fig. 4. Illustrations of the pre-grasping visual alignment applied to tasks `pouring water` (left) and `reorient board` (right). It shows that we can quickly adjust the grasp pose after the position and orientation of the manipulated object changes. It is best to zoom in to view the details.

The critical phase, therefore, lies in ensuring robust alignment and correction during the pre-grasping stage, where disturbances in object can significantly impact the manipulation success. To address this, we propose a lightweight visual alignment algorithm (refer illustrations in Fig. 4) that enables closed-loop pre-grasping by aligning the current object pose with the initial demonstrated configuration. Concretely, during the one-shot demonstration, we record the 6-DoF grasp pose $\widetilde{a}_g^\diamond = (\widetilde{a}_g^{p,\diamond}, \widetilde{a}_g^{r,\diamond})$ for the object original placement. At deployment time, when the same object is re-placed in a new position, we estimate the relative transformation with respect to the demonstrated pose. Given our tabletop workspace assumption, this transformation is mostly constrained to planar translation and in-plane rotation. To estimate them, we utilize a moment-based alignment method [98], [99] derived from 2D binary masks, which are segmented by employing VFMs [24], [25]. The geometric centroid of the mask provides the 2D projection of object center, and its displacement from the demonstrated mask yields $(\Delta x, \Delta y)$ in pixel space. Through the known hand-eye calibration matrix $\mathbf{T}_{\text{cam}\to\text{ee}}$, these pixel shifts are accurately mapped to real-world distances in the robot end-effector frame. For rotational alignment, we compute the second-order image moments of the 2D mask and estimate the principal axis direction. The difference in principal axis orientation before and after perturbation provides $\Delta\theta$, which is also converted into a rotation in the robot base frame. The final grasping pose is then updated as:

$$\widehat{a}_g^\diamond = \widetilde{a}_g^\diamond \oplus (\mathbf{T}_{\text{cam}\to\text{ee}}^{-1} \cdot T_\Delta \cdot \mathbf{T}_{\text{cam}\to\text{ee}}), \tag{4}$$

where $T_\Delta = (\Delta x, \Delta y, \Delta\theta)$ means the estimated planar transformation, and $\oplus$ denotes pose composition in $\mathbb{SE}(3)$.

This proposed alignment strategy does not require CAD-based 6D object pose estimators [100], nor does it rely on generic 6-DoF grasp pose detectors [101] which are often task-agnostic. Instead, it leverages task-specific grasp poses derived from demonstration, which are semantically meaningful and geometrically grounded. Furthermore, since no new deep models are introduced, the additional computational and memory overhead is minimal, enabling fast perception-action loops suitable for dynamic feedback. In experiments, we integrate this pre-grasping visual alignment module into our framework on both contralateral and humanoid dual-arm robots. Results demonstrate robust closed-loop grasp correction in the presence of dynamic perturbations on objects, further enhancing the reliability and transferability of YOTO++.

TABLE II
DETAILED STATISTICS OF TEN BIMANUAL TASKS. THE † MEANS WE ONLY COUNT THESE AUTO-ROLLOUT DEMONSTRATIONS.

Task Names	pull drawer	unscrew bottle	pour water	insert pen	reorient board	flip basket	uncover lid	open box	tool spoon	tool funnel
Is Synchronous?	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
# Manipulated Objects	2	1	2	3	1	1	1	1	3	3
# Substeps	6	5	6	6	5	3	3	4	6	8
# Keyframes	10	12	11	11	10	8	12	16	13	19
Avg. Duration (s)	42	51	53	59	40	23	27	35	59	82
# Categories	9 | 3	6	6 | 3	3 | 3	5	3	5	4	1 | 1 | 1	1 | 1 | 1
# Demonstrations †	243	54	162	243	45	27	45	36	27	27
# Testing Trials	54	30	36	36	25	15	25	20	10	10

Fig. 5. We collected a variety of manipulated objects in instance-level for each of **ten bimanual tasks** to improve and verify the generalizability of trained policies. All of these objects are from everyday life, not intentionally customized.

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 YOTO++ enable it to better learn bimanual manipulation policies? Q3: Do trained BiDP models generalize outside of the in-distribution domain? Q4: Is the presented framework YOTO++ compatible with a variety of long-horizon complex tasks? Q5: Does YOTO++ have good closed-loop control capabilities that can resist disturbance? Q6: Can YOTO++ be easily transferred to other dual-arm robots with different structures?

A. Experiment Setups

1) Tasks: We evaluate YOTO++ on ten real-world bimanual tasks. They collectively encompass two types of dual-arm collaborations: strictly asynchronous and synchronous. The manipulated objects in these tasks might be rigid, articulated 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: ① `pull drawer`: A drawer and a daily pocketed object. It consists of 6 substeps including stable drawer (L), pull drawer (R), pick up object (L), place object into drawer (L), stable drawer (L), and push drawer (R). ② `unscrew bottle`: A capped bottle with water. It consists of 5 substeps including pick up bottle (L), bring bottle close to right arm (L), unscrew cap (R), put down cap (R), and put down bottle (L). ③ `pour water`: A capless bottle with water and an empty mug. It consists of 6 substeps including pick up mug (R), pick up bottle (L), bring mug close to bottle (R), pour bottle's water into mug (L), put down bottle (L), and put down mug (R). ④ `insert pen`: A handleless cup and two differently oriented pens/spoons/forks. It consists of 6 substeps including pick up pen (R), pick up cup (L), insert pen into cup (R), pick up another pen (R), insert another pen into cup (R), and put down cup (L). ⑤ `reorient board`: An inverted board/spoon/shovel. It consists of 5 substeps including pick up board (R), reorient board (R), grasp board (L), loosen board (R), and reorient board to put down it (L). ⑥ `flip basket`: An inverted basket/pillow. It consists of 3 substeps including go to the bottom part of basket (LR), lift up basket (LR), and put down basket (LR). ⑦ `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 lid (LR), lift up lid (LR), and put down lid to one side (LR). ⑧ `open box`: A delivery box with four handleable wings. It consists of 4 substeps including go close to two vertical wings (LR), flick open two wings (LR), go close to two horizontal wings (LR), and flick open two wings (LR). ⑨ `tool spoon`: An inverted spoon with two bowls of different sizes. It consists of 6 substeps including pick up and reorient spoon (R), grasp spoon (L), loosen spoon (R), reorient spoon to scoop water from big bowl (L), reorient spoon to carry water to small bowl (L), and reorient spoon to pour water into small bowl (L). ⑩ `tool funnel`:An inverted funnel, a bottle and a mug. It consists of 8 substeps including pick up and reorient funnel (R), grasp funnel (L), loosen funnel (R), reorient funnel to insert it into bottle (L), reorient arm to pick up bottle (L), pick up mug (R), bring bottle close to mug (L), and pour mug's water into bottle via funnel (R).

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. 8.

2) Demonstrations: Current 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. 5.

Specifically, we first implement the auto-rollout strategy to collect real robot data. We set 3 (for tasks with multiple objects) or 9 (for tasks with only one object) 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 second to 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 [9], [28]–[30] 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 [11] (6K+ self-created episodes) and π_0 [58] (5∼100 hours post-training data), but our cost is extremely low.

3) Baselines: We compare our BiDP to four strong baselines. (1) *Action Chunking Transformers (ACT)* [9]. It is proposed by ALOHA and uses a well-designed transformer structure as the visual encoder. (2) *Diffusion Policy (DP)* [28]. The vanilla diffusion policy uses RGB images as inputs and ResNet [102] as the visual encoder. We modified it by using point cloud scenes as observations and a PointNet++ encoder [95]. (3) *3D Diffusion Policy (DP3)* [29]. 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* [30]. 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 or 2 trials for paired objects in every task (refer the last row of Tab. II). Trails for two tool-using tasks are 10. These objects have randomized initial placements. For a more detailed comparison, we report the **average length** (following CLAVIN [103]) 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 and treat them as unseen objects to evaluate the final trained model. The number of all OOD trials is quadrupled.

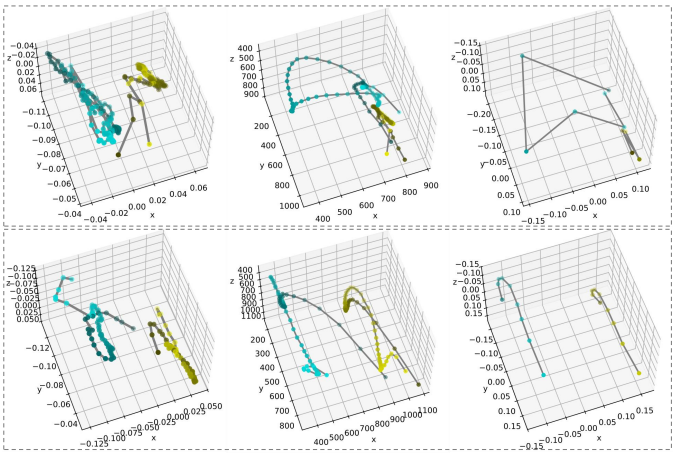


Fig. 6. Illustrations of extracted hand motion trajectories by using **(a)** unprocessed raw 3D hand center points, **(b)** projected hand center points on the 2D image, and **(c)** lifted 3D points in simplified keyframes. The first and second line represents the task *pull drawer* and *uncover lid*, 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	✗	✗	✗	✗	13/54 (24.1%)	3.54
2	✓	✗	✗	✗	26/54 (48.1%)	3.80
3	✗	✓	✗	✗	28/54 (51.9%)	4.15
4	✓	✓	✗	✗	31/54 (57.4%)	4.31
5	✓	✓	✓	✗	33/54 (61.1%)	4.48
6	✓	✓	✗	✓	42/54 (77.8%)	5.15
7	✓	✓	✓	✓	43/54 (79.6%)	5.31

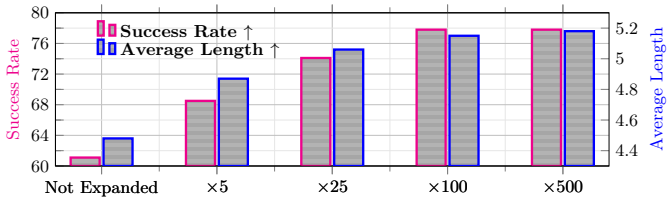


Fig. 7. Ablation studies on expanded training data at different scales using geometric transformations. The task *pull drawer* with 243 episodes is treated as the not expanded version.

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. 6, 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)							unscrew bottle (54 episodes)						pour water (162 episodes)						
	stable drawer	pull drawer	pick object	place object	stable drawer	push drawer	**Avg. Len.**	pick bottle	close to right	unscrew cap	place cap	place bottle	**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	24/30	22/30	02/30	02/30	02/30	1.73	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	26/30	26/30	06/30	06/30	06/30	2.33	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	27/30	27/30	06/30	06/30	05/30	2.37	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	28/30	28/30	08/30	07/30	06/30	2.57	32/36	30/36	30/36	11/36	10/36	09/36	3.39
BiDP	54/54	52/54	48/54	45/54	45/54	43/54	5.31	30/30	30/30	24/30	24/30	23/30	4.37	35/36	34/36	34/36	29/36	28/36	28/36	5.22

Methods	insert pen (243 episodes)							reorient board (45 episodes)						flip basket (27 episodes)				uncover lid (45 episodes)			
	pick pen	pick cup	insert pen	pick pen+	insert pen+	place cup	**Avg. Len.**	pick board	reorient board	grasp board	loosen board	place board	**Avg. Len.**	close to basket	lift up basket	place basket	**Avg. Len.**	close to lid	lift up lid	place lid	**Avg. Len.**
	29/36	15/36	05/36	02/36	01/36	01/36	1.47	14/25	10/25	05/25	04/25	03/25	1.44	09/15	04/15	01/15	0.56	23/25	08/25	01/25	1.28
	33/36	20/36	08/36	05/36	02/36	02/36	1.94	15/25	10/25	06/25	05/25	05/25	1.64	09/15	03/15	01/15	0.52	23/25	16/25	04/25	1.72
	35/36	25/36	13/36	09/36	05/36	05/36	2.56	19/25	16/25	08/25	07/25	07/25	2.28	13/15	07/15	02/15	0.88	24/25	19/25	06/25	1.96
	34/36	27/36	15/36	12/36	07/36	06/36	2.81	22/25	20/25	12/25	10/25	09/25	2.92	14/15	10/15	05/15	1.16	24/25	18/25	07/25	1.96
	36/36	33/36	31/36	29/36	28/36	28/36	5.14	24/25	22/25	21/25	20/25	20/25	4.28	15/15	13/15	10/15	2.53	25/25	24/25	20/25	2.76

Methods	open box (36 episodes)					tool spoon (27 episodes)							tool funnel (27 episodes)								
	close to wings	open wings	close to wings	open wings	**Avg. Len.**	pick spoon	grasp spoon	loosen spoon	scoop water	carry water	pour water	**Avg. Len.**	pick funnel	grasp funnel	loosen funnel	insert funnel	pick bottle	pick mug	close to mug	pour water	**Avg. Len.**
	15/20	05/20	05/20	00/20	1.25	03/10	02/10	01/10	00/10	00/10	00/10	0.60	04/10	02/10	00/10	00/10	00/10	00/10	00/10	00/10	0.60
	19/20	07/20	06/20	03/20	1.75	04/10	02/10	02/10	00/10	00/10	00/10	0.80	05/10	02/10	02/10	00/10	00/10	00/10	00/10	00/10	0.90
	20/20	08/20	08/20	04/20	2.00	07/10	04/10	04/10	01/10	01/10	00/10	1.70	08/10	05/10	04/10	01/10	01/10	00/10	00/10	00/10	1.90
	20/20	10/20	09/20	04/20	2.35	08/10	05/10	05/10	02/10	02/10	01/10	2.30	08/10	06/10	05/10	02/10	01/10	00/10	00/10	00/10	2.20
	20/20	19/20	19/20	14/20	3.60	10/10	08/10	08/10	06/10	06/10	05/10	4.30	09/10	08/10	08/10	05/10	04/10	04/10	04/10	03/10	4.50

TABLE V

COMPARISON OF THE AVERAGE SUCCESS RATE OF VARIOUS METHODS ON ALL TEN TASKS (**IN-DISTRIBUTION EVALUATIONS**).

Methods	ACT	DP	DP3	EquiBot	**BiDP**
Average Success Rate	4.97%	11.10%	15.20%	21.31%	65.85%

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	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 [20] or WiLoR [18]), the resulting hand trajectory is always unstable and difficult to parse (see Fig. 6 **(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 [15], [17], [61]. 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. 6 **(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. 6 **(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 [30], 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

Fig. 8. Visualization of ten 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. It is best to zoom in to view the details.

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. 7 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 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 [9], [28]–[30] 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 ten 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 (such as flipping and tool-using), the existing state-of-the-art methods still have a lot of room for improvement, such as the gradually decaying

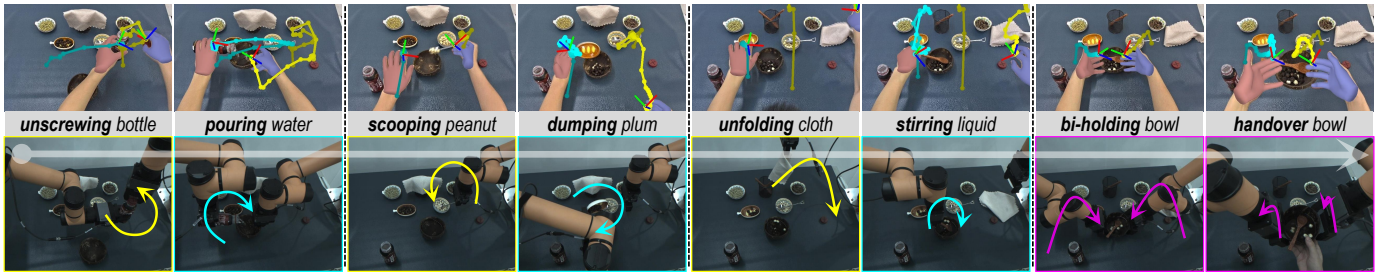


Fig. 9. Illustrations of another super long-horizon bimanual task containing various atomic skills. **Top:** the visualization of hand motions extraction. **Bottom:** the corresponding rollout examples by injecting actions on real robots. Refer to Fig. 1 and Fig. 8 for notes on different colors and curves.

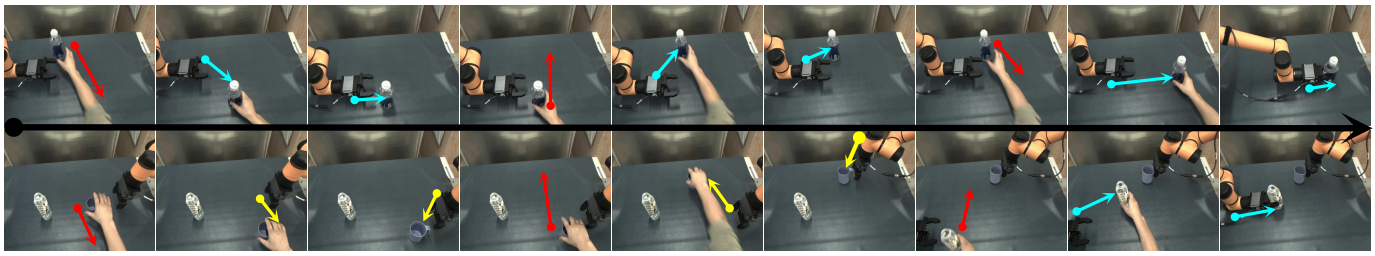


Fig. 10. Example of dynamic interferences during the pre-grasping stage for tasks `unscrew bottle` (top row) and `pour water` (bottom row), where each object is manually disturbed with one, two or three times. The red arrow indicates the direction of the manually moved object (interfering). The teal arrow and olive arrow indicate the movement direction of the left and right robotic arms (chasing) respectively.

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 ten tasks in Tab. V, where our method BiDP achieved a success rate of nearly 66%, 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 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 [24] and segmentation [25], 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 YOTO++ is compatible with most bimanual tasks, such as the selected ten 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 above-mentioned quantitative results (Tab. IV and Tab. V), we also qualitatively demonstrate the visual effects of real robot execution on ten tasks in Fig. 8, mainly showing 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.

TABLE VII
THE SUCCESS RATE OF **PRE-GRASPING (P-G)** AND **FULLY COMPLETING (F-C)** TASK UNDER DIFFERENT TIMES OF DYNAMIC DISTURBANCE.

Tasks	Success Rate	Dynamic Disturbance Times				
		#1	#2	#3	#4	#5
unscrew bottle	P-G	28/30	27/30	27/30	24/30	21/30
	F-C	23/30	23/30	22/30	20/30	17/30
pour water	P-G	34/36	34/36	33/36	30/36	26/36
	F-C	28/36	27/36	26/36	23/36	20/36
reorient board	P-G	23/25	23/25	22/25	20/25	16/25
	F-C	20/25	19/25	19/25	17/25	14/25
tool spoon	P-G	08/10	08/10	08/10	06/10	04/10
	F-C	05/10	05/10	04/10	02/10	02/10
Avg. SR.	P-G	89.9%	89.1%	87.4%	75.8%	61.6%
	F-C	71.1%	69.4%	65.4%	54.6%	47.1%

Moreover, we selected a typical super long-horizon bimanual task (`snack making`) and enabled the dual-arm robot to learn new given goals 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. 9. 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 tasks.

(Q5) **YOTO++ can achieve good anti-interference effect in the pre-grasping stage**. To evaluate the effectiveness of our vision-guided pre-grasping algorithm, we conducted controlled tests on four bimanual tasks under varying levels of external disturbances, as summarized in Tab. VII. The

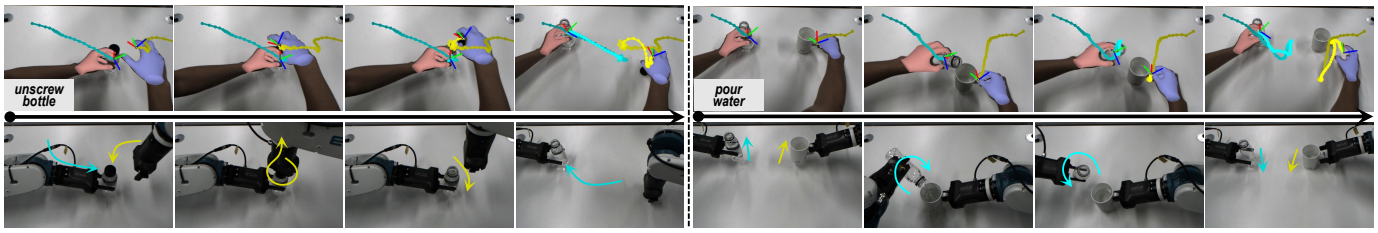


Fig. 11. Illustrations of two selected bimanual tasks transferred to the humanoid robot. **Top Row:** the visualization of hand motions extraction. **Bottom Row:** the corresponding rollout examples by injecting actions on real robots. Refer to Fig. 1 and Fig. 8 for notes on different colors and curves.

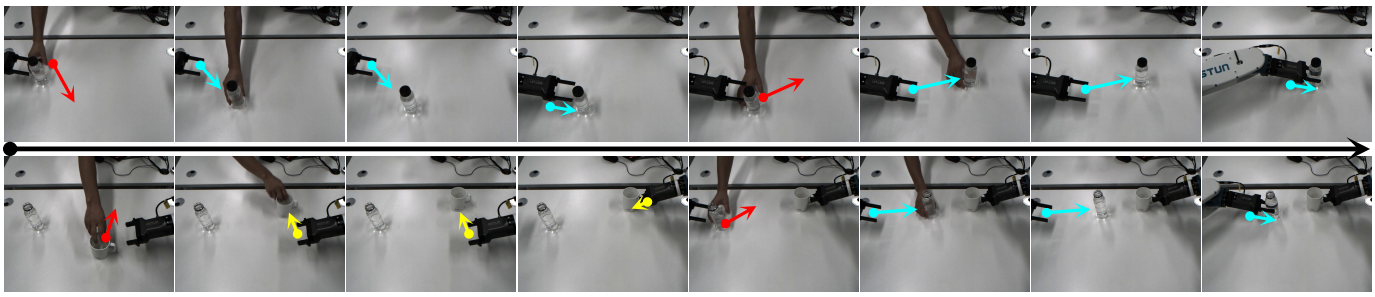


Fig. 12. Example of dynamic interferences during the pre-grasping stage for tasks `unscrew bottle` (top row) and `pour water` (bottom row) on the new **humanoid dual-arm robot**. Here, each object is manually disturbed with one or two times. Refer to Fig. 10 for notes on different colors of arrows.

evaluation criteria remain consistent with those used in the in-distribution setting (Tab. IV), except that we adopt a hybrid control strategy: the closed-loop pre-grasping alignment, while the remainder of the task follows open-loop execution. This hybrid scheme is designed to balance robustness and efficiency, offering an advantage over full closed-loop alternatives, as further evidenced by the quantitative trends.

Overall, as the number of injected perturbations increases from 1 to 5, both the success rate of pre-grasping and overall task completion show a gradual decline, which is expected. However, the gap between successful grasping and final task success remains consistently small across all conditions, highlighting the critical role of the pre-grasping phase. These results indicate that once a stable grasp is established (effectively treating the gripper and object as a rigid composite), the rest of the task proceeds reliably under open-loop control. Comparing with the full closed-loop baseline BiDP from Tab. IV, we observe that our hybrid approach often achieves higher task success rates under light-to-moderate interference (1$\sim$3 disturbances). This is largely due to the fact that, during disturbance injection, the robot arm has already partially approached the object, reducing the likelihood of collision and enabling more accurate alignment. As disturbance frequency increases, however, error accumulation and partial occlusion by the approaching arm degrade performance slightly. Additionally, qualitative results in Fig. 10 illustrate the pre-grasp alignment process under ongoing perturbations, further validating the robustness of our method. Together, these findings demonstrate that the designed closed-loop pre-grasping module effectively mitigates one of the most fragile phases in bimanual manipulation (*e.g.*, the initial object contact) thereby contributing significantly to overall task robustness.

(Q6) **YOTO++ can be seamlessly transferred to a new humanoid dual-arm robot**. Our YOTO++ is inherently hardware-agnostic by design. Since human-demonstrated dual-hand trajectories are extracted and encoded in a robot-agnostic space, they can be injected into any dual-arm robotic system as long as the actions remain within its reachable workspace. To validate this, we deploy YOTO++ on a structurally different humanoid dual-arm robot (see Fig. 1, lower-left), which features an anthropomorphic layout more common in general-purpose platforms (discussed in Sec. III).

In this setup, we retain the original hardware assumptions: parallel-jaw grippers and a third-view binocular stereo camera. Without retraining, we directly reuse the motion extraction and injection module to transfer human-demonstrated actions onto the new platform. We evaluate this setup on two representative tasks `unscrew bottle` and `pour water`, both requiring precise coordination and long-horizon planning. As shown in Fig. 11, the system continues to robustly extract dual-hand trajectories and execute keyframe-based actions on the new robot, confirming the successful transfer of core capabilities. Furthermore, in scenarios where object variation is limited (*i.e.*, intra-instance consistency), we observe that the visual alignment module introduced in Sec. IV-E remains effective. As shown in Fig. 12, YOTO++ can still achieve the train-free closed-loop pre-grasping, followed by direct replay of the demonstrated action sequence, completing the task without additional adaptation. These results provide strong empirical evidence for its cross-embodiment generality and practical deployability across diverse dual-arm robotic systems.

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 such as pose, joints, and contact states. To ensure stable and precise execution, we simplify noisy hand trajectories into discrete keyframes and introduce a motion mask to regulate dual-arm coordination. On top of this, we develop a scalable demonstration proliferation

strategy that combines real-world auto-rollout and geometric transformation to generate diverse training data efficiently. With this enriched dataset, we train a dedicated bimanual diffusion policy (BiDP) that simplifies visual inputs, predicts task-relevant keyposes, and reorganizes action spaces for more tractable learning. In this extended version, we further demonstrate the broad generalization of YOTO++ by introducing more tasks involving new atomic skills and tool usage, revealing its strong spatial-temporal consistency in multi-stage manipulation. We also integrate a lightweight visual alignment module for closed-loop pre-grasping correction, enabling robustness against dynamic disturbances. Finally, we validate the cross-embodiment applicability of YOTO++ by transferring it to a humanoid dual-arm platform without retraining. These contributions together form a unified and practical solution for scalable, robust, and generalizable bimanual manipulation, advancing the frontier of imitation learning in real robots.

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 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 robotic manipulation.

REFERENCES

[1] P. Hebert, N. Hudson, J. Ma, and J. W. Burdick, "Dual arm estimation for coordinated bimanual manipulation," in *ICRA*. IEEE, 2013, pp. 120–125.

[2] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[3] F. Xie, A. Chowdhury, M. De Paolis Kaluza, L. Zhao, L. Wong, and R. Yu, "Deep imitation learning for bimanual robotic manipulation," *NeurIPS*, vol. 33, pp. 2327–2337, 2020.

[4] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta, "Efficient bimanual manipulation using learned task schemas," in *ICRA*. IEEE, 2020, pp. 1149–1155.

[5] M. Drolet, S. Stepputtis, S. Kailas, A. Jain, J. Peters, S. Schaal, and H. B. Amor, "A comparison of imitation learning algorithms for bimanual manipulation," *RAL*, 2024.

[6] M. Grotz, M. Shridhar, T. Asfour, and D. Fox, "Peract2: Benchmarking and learning for robotic bimanual manipulation tasks," *arXiv preprint arXiv:2407.00278*, 2024.

[7] I.-C. A. Liu, S. He, D. Seita, and G. S. Sukhatme, "Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation," in *CoRL*, 2024.

[8] F. Krebs and T. Asfour, "A bimanual manipulation taxonomy," *RAL*, vol. 7, no. 4, pp. 11 031–11 038, 2022.

[9] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *RSS*, 2023.

[10] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *ICRA*. IEEE, 2024, pp. 6892–6903.

[11] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," in *ICLR*, 2025.

[12] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín, "Screwmimic: Bimanual imitation from human videos with screw space projection," in *RSS*, 2024.

[13] J. Gao, X. Jin, F. Krebs, N. Jaquier, and T. Asfour, "Bi-kvil: Keypoints-based visual imitation learning of bimanual manipulation tasks," in *ICRA*. IEEE, 2024, pp. 16 850–16 857.

[14] Y. Chen, C. Wang, Y. Yang, and K. Liu, "Object-centric dexterous manipulation from human motion data," in *CoRL*, 2024.

[15] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "Okami: Teaching humanoid robots manipulation skills through single video imitation," in *CoRL*, 2024.

[16] W. Peng, J. Lv, Y. Zeng, H. Chen, S. Zhao, J. Sun, C. Lu, and L. Shao, "Tiebot: Learning to knot a tie from visual demonstration through a real-to-sim-to-real approach," in *CoRL*, 2024.

[17] J. Kerr, C. M. Kim, M. Wu, B. Yi, Q. Wang, K. Goldberg, and A. Kanazawa, "Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction," in *CoRL*, 2024.

[18] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou, "Wilor: End-to-end 3d hand localization and reconstruction in-the-wild," in *CVPR*, 2025.

[19] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, "Understanding human hands in contact at internet scale," in *CVPR*, 2020, pp. 9869–9878.

[20] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, "Reconstructing hands in 3d with transformers," in *CVPR*, 2024, pp. 9826–9836.

[21] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in *CVPR*, 2022, pp. 13 739–13 748.

[22] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *CoRL*. PMLR, 2023, pp. 785–799.

[23] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," in *RSS*, 2024.

[24] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," in *CVPR*, 2024, pp. 4818–4829.

[25] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," in *ICLR*, 2025.

[26] G. Xu, X. Wang, X. Ding, and X. Yang, "Iterative geometry encoding volume for stereo matching," in *CVPR*, 2023, pp. 21 919–21 928.

[27] G. Xu, X. Wang, Z. Zhang, J. Cheng, C. Liao, and X. Yang, "Igev++: Iterative multi-range geometry encoding volumes for stereo matching," *TPAMI*, 2025.

[28] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *IJRR*, p. 02783649241273668, 2023.

[29] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *RSS*, 2024.

[30] J. Yang, Z. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg, "Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning," in *CoRL*, 2024.

[31] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "Rvt-2: Learning precise manipulation from few demonstrations," in *RSS*, 2024.

[32] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *CVPR*, 2024, pp. 18 081–18 090.

[33] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki, "Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation," in *CoRL*, 2023.

[34] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," in *CoRL*, 2024.

[35] J. Zeng, Q. Bu, B. Wang, W. Xia, L. Chen, H. Dong, H. Song, D. Wang, D. Hu, P. Luo *et al.*, "Learning manipulation by predicting interaction," in *RSS*, 2024.

[36] H. Zhou, R. Wang, Y. Tai, Y. Deng, G. Liu, and K. Jia, "You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations," in *RSS*, 2025.

[37] A. Colomé and C. Torras, "Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes," *TRO*, vol. 34, no. 3, pp. 602–615, 2018.

[38] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *CoRL*. PMLR, 2022, pp. 192–202.

[39] A. Canberk, C. Chi, H. Ha, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Cloth funnels: Canonicalized-alignment for multi-purpose garment manipulation," in *ICRA*. IEEE, 2023, pp. 5872–5879.

[40] L. Y. Chen, B. Shi, D. Seita, R. Cheng, T. Kollar, D. Held, and K. Goldberg, "Autobag: Learning to open plastic bags and insert objects," in *ICRA*. IEEE, 2023, pp. 3918–3925.

[41] A. Bahety, S. Jain, H. Ha, N. Hager, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Bag all you need: Learning a generalizable bagging strategy for heterogeneous objects," in *IROS*. IEEE, 2023, pp. 960–967.

[42] B. Huang, Y. Chen, T. Wang, Y. Qin, Y. Yang, N. Atanasov, and X. Wang, "Dynamic handover: Throw and catch with bimanual hands," in *CoRL*. PMLR, 2023, pp. 1887–1902.

[43] L. Yan, T. Stouraitis, J. Moura, W. Xu, M. Gienger, and S. Vijayakumar, "Impact-aware bimanual catching of large-momentum objects," *TRO*, 2024.

[44] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu, "Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning," in *ICRA*. IEEE, 2023, pp. 3867–3874.

[45] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik, "Twisting lids off with two hands," in *CoRL*, 2024.

[46] J. Zhu, M. Gienger, G. Franzese, and J. Kober, "Do you need a hand?–a bimanual robotic dressing assistance scheme," *TRO*, vol. 40, pp. 1906–1919, 2024.

[47] S. S. Mirrazavi Salehian, N. B. Figueroa Fernandez, and A. Billard, "Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty," in *RSS*, 2016.

[48] V. N. Hartmann, A. Orthey, D. Driess, O. S. Oguz, and M. Toussaint, "Long-horizon multi-robot rearrangement planning for construction assembly," *TRO*, vol. 39, no. 1, pp. 239–252, 2022.

[49] Y. Zhao, R. Wu, Z. Chen, Y. Zhang, Q. Fan, K. Mo, and H. Dong, "Dualafford: Learning collaborative visual affordance for dual-gripper manipulation," in *ICLR*, 2023.

[50] H. Razali and Y. Demiris, "Keystate-driven long-term generation of bimanual object manipulation sequences," *TPAMI*, 2025.

[51] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, "Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects," *RAL*, vol. 7, no. 2, pp. 5159–5166, 2022.

[52] J. Grannen, Y. Wu, B. Vu, and D. Sadigh, "Stabilize to act: Learning to coordinate for bimanual manipulation," in *CoRL*. PMLR, 2023, pp. 563–576.

[53] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation," in *CoRL*, 2024.

[54] J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, K. Draper, D. Dwibedi, C. Finn, P. Florence, S. Goodrich *et al.*, "Aloha 2: An enhanced low-cost hardware for bimanual teleoperation," *arXiv preprint arXiv:2405.02292*, 2024.

[55] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," in *CoRL*, 2024.

[56] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," in *RSS*, 2024.

[57] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *CoRL*, 2024.

[58] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," in *RSS*, 2025.

[59] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik, "Learning visuotactile skills with two multifingered hands," *arXiv preprint arXiv:2404.16823*, 2024.

[60] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak, "Bimanual dexterity for complex tasks," in *CoRL*, 2024.

[61] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," in *CoRL*, 2024.

[62] Y. Chen, Y. Geng, F. Zhong, J. Ji, J. Jiang, Z. Lu, H. Dong, and Y. Yang, "Bi-dexhands: Towards human-level bimanual dexterous manipulation," *TPAMI*, vol. 46, no. 5, pp. 2804–2818, 2023.

[63] S. Yan, Z. Zhang, M. Han, Z. Wang, Q. Xie, Z. Li, Z. Li, H. Liu, X. Wang, and S.-C. Zhu, "M 2 diffuser: Diffusion-based trajectory optimization for mobile manipulation in 3d scenes," *TPAMI*, 2025.

[64] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang, "Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning," *arXiv preprint arXiv:2407.03162*, 2024.

[65] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu, "Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback," in *ICRA*, 2024.

[66] I. Chuang, A. Lee, D. Gao, and I. Soltani, "Active vision might be all you need: Exploring active vision in bimanual robotic manipulation," *arXiv preprint arXiv:2409.17435*, 2024.

[67] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," in *CoRL*, 2024.

[68] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *CVPR*, 2022, pp. 18 995–19 012.

[69] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges, "Arctic: A dataset for dexterous bimanual hand-object manipulation," in *CVPR*, 2023, pp. 12 943–12 954.

[70] X. Zhan, L. Yang, Y. Zhao, K. Mao, H. Xu, Z. Lin, K. Li, and C. Lu, "Oakink2: A dataset of bimanual hands-object manipulation in complex task completion," in *CVPR*, 2024, pp. 445–456.

[71] Y. Liu, H. Yang, X. Si, L. Liu, Z. Li, Y. Zhang, Y. Liu, and L. Yi, "Taco: Benchmarking generalizable bimanual tool-action-object understanding," in *CVPR*, 2024, pp. 21 740–21 751.

[72] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote *et al.*, "Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives," in *CVPR*, 2024, pp. 19 383–19 400.

[73] G. Papagiannis, N. Di Palo, P. Vitiello, and E. Johns, "R+x: Retrieval and execution from everyday human videos," *arXiv preprint arXiv:2407.12957*, 2024.

[74] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," in *RSS*, 2024.

[75] G. Li, N. Tsagkas, J. Song, R. Mon-Williams, S. Vijayakumar, K. Shao, and L. Sevilla-Lara, "Learning precise affordances from egocentric videos for robotic manipulation," *arXiv preprint arXiv:2408.10123*, 2024.

[76] S. Nasiriany, S. Kirmani, T. Ding, L. Smith, Y. Zhu, D. Driess, D. Sadigh, and T. Xiao, "Rt-affordance: Affordances are versatile intermediate representations for robot manipulation," *arXiv preprint arXiv:2411.02704*, 2024.

[77] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu, "Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation," in *ECCV*. Springer, 2024, pp. 222–239.

[78] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to act from actionless videos through dense correspondences," in *ICLR*, 2024.

[79] X. Zhang and A. Boularias, "One-shot imitation learning with invariance matching for robotic manipulation," in *RSS*, 2024.

[80] A. Bandini and J. Zariffa, "Analysis of the hands in egocentric vision: A survey," *TPAMI*, vol. 45, no. 6, pp. 6846–6866, 2020.

[81] T. Zhu, R. Wu, J. Hang, X. Lin, and Y. Sun, "Toward human-like grasp: Functional grasp by dexterous robotic hand via object-hand semantic representation," *TPAMI*, vol. 45, no. 10, pp. 12 521–12 534, 2023.

[82] L. Yang, X. Zhan, K. Li, W. Xu, J. Zhang, J. Li, and C. Lu, "Learning a contact potential field for modeling the hand-object interaction," *TPAMI*, 2024.

[83] E. Johns, "Coarse-to-fine imitation learning: Robot manipulation from a single demonstration," in *ICRA*. IEEE, 2021, pp. 4613–4619.

[84] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *CoRL*. PMLR, 2022, pp. 1678–1690.

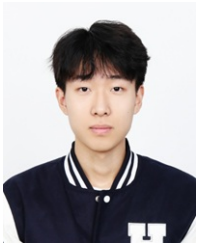
[85] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *CoRL*. PMLR, 2022, pp. 991–1002.

[86] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[87] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICLR*, 2021.

[88] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*. PMLR, 2021, pp. 8162–8171.
[89] J. Yang, C. Deng, J. Wu, R. Antonova, L. Guibas, and J. Bohg, "Equivact: Sim (3)-equivariant visuomotor policies beyond rigid object manipulation," in *ICRA*. IEEE, 2024, pp. 9249–9255.
[90] H. Ryu, J. Kim, H. An, J. Chang, J. Seo, T. Kim, Y. Kim, C. Hwang, J. Choi, and R. Horowitz, "Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation," in *CVPR*, 2024, pp. 18 007–18 018.
[91] J. Brehmer, J. Bose, P. De Haan, and T. S. Cohen, "Edgi: Equivariant diffusion for planning with embodied agents," *NeurIPS*, vol. 36, 2024.
[92] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ToG*, vol. 36, no. 6, 2017.
[93] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *RAM*, vol. 11, no. 4, pp. 110–122, 2004.
[94] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," in *RSS*, 2007.
[95] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS*, vol. 30, 2017.
[96] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *AAAI*, vol. 32, no. 1, 2018.
[97] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.
[98] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *TRO*, vol. 20, no. 4, pp. 713–723, 2004.
[99] L. Kotoulas and I. Andreadis, "Accurate calculation of image moments," *TIP*, vol. 16, no. 8, pp. 2028–2037, 2007.
[100] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *CVPR*, 2024, pp. 17 868–17 879.
[101] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *TRO*, vol. 39, no. 5, pp. 3929–3945, 2023.
[102] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
[103] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *RAL*, vol. 7, no. 3, pp. 7327–7334, 2022.

Huayi Zhou is now a postdoctoral researcher at School of Data Science, The Chinese University of Hong Kong, Shenzhen. Before that, he received his B.S degree at Dept. of Computer Science from Hunan University in 2017, and both M.S. degree and Ph.D. degree at Dept. of Computer Science and Engineering from Shanghai Jiao Tong University in 2020 and 2024, respectively. His research interests lie in robot manipulation, multi-modal perception, and computer vision (e.g., object detection, pose estimation, and monocular 3D reconstruction), combined with enduring machine learning techniques such as multi-task learning, domain adaptation, domain generalization and semi-supervised learning.



Ruixiang Wang is currently a final-year undergraduate student at Harbin Institute of Technology, Weihai, expected to graduate with a B.E. degree in Automation in 2025. He will join The Chinese University of Hong Kong, Shenzhen as a Ph.D. candidate under the supervision of Prof. Kui Jia, with research interests in embodied intelligence.

Yunxin Tai is currently an engineer with DexForce Technology Corporation. He received the B.E. degree from South China University of Technology in 2018. His research interests include sim2real object detection, 6d pose estimation, and domain adaptation.

Yueci Deng received his B.Eng. degree in Electronic Information Engineering from the University of Electronic Science and Technology of China (joint program with the University of Glasgow), and his M.Sc. degree in Signal Processing from Nanyang Technological University, Singapore. He is currently the head of the Fundamental Engine team at DexForce Technology Co., Ltd., where he leads research and development in Sim2Real Embodied AI, simulation engines, and large-scale synthetic data generation. He serves as a collaborator on the Open3D open-source project and is the principal investigator of a multimillion-RMB research grant on multimodal databases for robot learning. His research interests include 3D computer vision/graphics, robotics, machine learning. He has authored several patents and publications in top-tier AI and robotics conferences.

Guiliang Liu is currently working as an Assistant Professor at the School of Data Science at The Chinese University of Hong Kong, Shenzhen. He obtained his undergraduate degree from South China University of Technology. He then earned his Ph.D. in Computer Science from Simon Fraser University in Canada and completed postdoctoral research at the University of Waterloo and the Vector Institute in Canada. His research primarily focuses on reinforcement learning and embodied decision-making. In the field of safe reinforcement learning, he leverages inverse constraint inference methods to enhance the safety of reinforcement learning systems. Additionally, he specializes in embodied robotic manipulation skills, developing efficient data engines to improve robotic operation in complex tasks, and designing robust control algorithms to ensure the safety and stability of humanoid robots in challenging environments. His research collaborator includes Baidu Research, Huawei Noah's Ark Lab and DexForce Research.

Kui Jia is currently a Professor with School of Data Science, The Chinese University of Hong Kong, Shenzhen. He received the B.Eng. degree in marine engineering from Northwestern Polytechnical University, China, in 2001, the M.Eng. degree in electrical and computer engineering from the National University of Singapore in 2003, and the Ph.D. degree in computer science from Queen Mary University of London, London, U.K., in 2007. His research interests are in computer vision and machine learning. His recent research focuses on theoretical deep learning and its applications to 3D vision. He has been serving as an Associate Editor for IEEE Trans. on Image Processing and Trans. on Machine Learning Research.