

BiRoAD: Learning Shared and Role-Adaptive Representations for Bimanual Manipulation

Yan Shen^{1,4,*} Yuchen Liu^{1,*} Feng Jiang¹ Hangtian Hu¹
Xiaoqi Li^{1,4} Shu Chen³ Ruihai Wu² Hao Dong^{1,4,†}

¹CFCS, School of Computer Science, Peking University ²UC Berkeley

³Hong Kong University of Science and Technology (Guangzhou) ⁴PrimeBot Research Institute

*Equal contribution. †Corresponding author.

Abstract:

Bimanual manipulation requires policies that coordinate two arms while adapting their functional roles to scene geometry, object configuration, and task context. Learning such scene-conditioned role adaptation remains challenging, as demonstrations may contain uneven role distributions that limit generalization to under-represented arm–role configurations. In addition, many bimanual policies predict actions in fixed left- and right-arm action spaces. While this provides a natural parameterization for robot control, it does not explicitly specify how behaviors should transform when functional roles are exchanged across arms. Across different scene initializations, the two arms may follow a similar coordination pattern, but the role-specific behavior assigned to each arm should change with the scene. Therefore, we propose **BiRoAD**, a **Bimanual Role-Adaptive Decomposition** framework for learning shared and role-adaptive representations in bimanual policies. Given bimanual trajectory or action-token features, BiRoAD decomposes these features into swap-symmetric and swap-antisymmetric components: the former captures coordination structure invariant to arm exchange, and the latter captures role-specific distinctions that vary consistently with functional role assignment. The two components are then recomposed as residual updates to the original paired arm representations, allowing BiRoAD to serve as a modular feature transformation without changing the policy inputs, imitation-learning objective, or requiring manually defined role labels. Across multiple bimanual manipulation tasks with balanced and imbalanced role distributions, BiRoAD improves robustness across role configurations over corresponding base policies, with notable gains on underrepresented role configurations. Project Page: sxy7147.github.io/biroad-website/.

Keywords: Bimanual Manipulation, Imitation Learning

1 Introduction

Bimanual manipulation enables robots to coordinate two arms when interacting with objects and environments, extending their capabilities beyond single-arm operation [1, 2, 3, 4, 5, 6]. In many manipulation tasks, variations in object configuration and workspace constraints require the two arms to adapt their functional roles to the scene [7, 8]. Depending on the scene, either arm may need to take a leading, supporting, or operating role. However, demonstrations may exhibit uneven role distributions, where certain arm–role configurations appear more frequently than their mirrored or less common counterparts. Such imbalance can bias learned policies toward frequent arm-specific behaviors and hinder generalization to underrepresented role configurations. This creates a central challenge for policy learning: a bimanual policy should capture coordination structure shared across the arm pair while preserving role-specific distinctions that adapt to the scene.

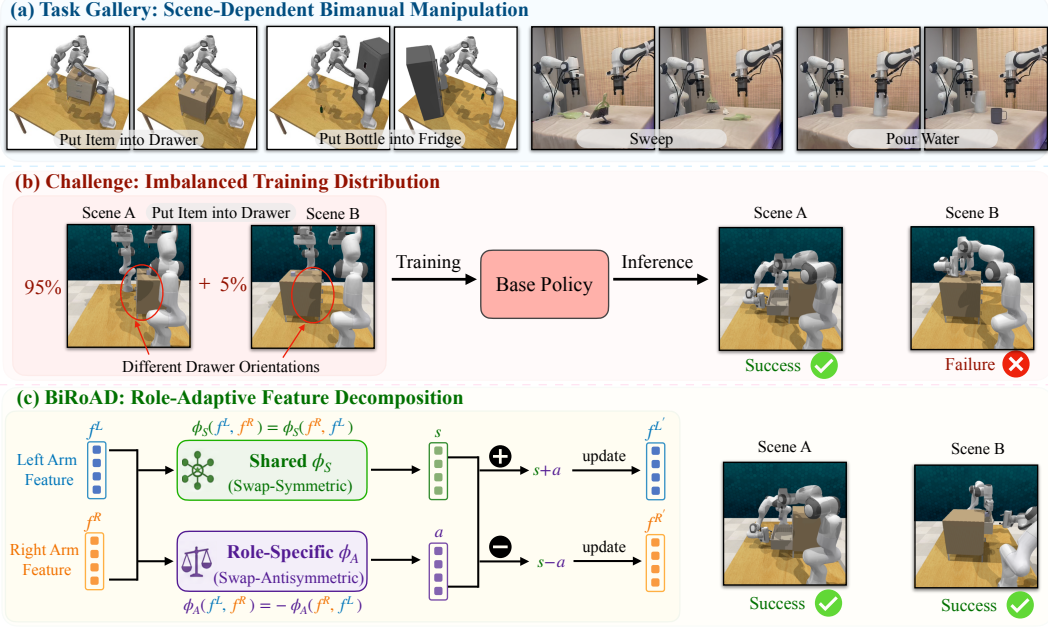


Figure 1: **Overview of BiRoAD.** (a) Bimanual tasks in which scene initializations induce different functional roles for two arms. (b) Under imbalanced training distributions, a base policy may favor frequent arm–role configurations and perform less reliably when the scene induces a less common role assignment. (c) BiRoAD decomposes paired arm features into swap-symmetric shared coordination and swap-antisymmetric role-specific components, then recomposes them as residual updates to encourage structured cross-arm sharing while preserving scene-dependent role adaptation.

Recent work has made substantial progress in bimanual manipulation [3, 9, 10, 11, 12, 13], with many policies predicting bimanual actions in fixed left- and right-arm action spaces as a natural and effective parameterization for robot control. Although modern architectures can support cross-arm information sharing through joint representations, they do not explicitly specify how behaviors should transform when functional roles are exchanged across arms. This ambiguity can be amplified under uneven role distributions, where frequently observed behaviors may become coupled with arm identity rather than grounded in the scene-conditioned functional role. These observations motivate representations that promote structured cross-arm sharing while retaining role-specific distinctions for scene-dependent adaptation.

In this work, as shown in Figure 1, we propose **BiRoAD**, a **Bimanual Role-Adaptive Decomposition** framework for learning shared and role-adaptive representations in bimanual policies. BiRoAD is motivated by the requirements of role-adaptive bimanual collaboration, which involves two complementary structures: shared coordination patterns that should remain consistent under arm exchange, and role-specific variation that should change consistently with the assigned functional roles. Given paired arm trajectory or action-token features, BiRoAD applies a symmetric–antisymmetric decomposition to organize bimanual interaction into these components. The swap–symmetric component models coordination structure that is invariant to arm exchange, while the swap–antisymmetric component captures role-specific distinctions that vary with the assigned roles. The outputs of the two branches are then recomposed as residual updates to the original paired arm representations, allowing BiRoAD to serve as a modular feature transformation without changing the policy inputs or imitation-learning objective. BiRoAD does not require additional annotations or manually defined role labels, which are often task-dependent and difficult to define consistently across diverse bimanual behaviors. Overall, BiRoAD provides a local symmetry-aware inductive bias that encourages both cross-arm sharing and role-specific differentiation within bimanual representations.

To evaluate role adaptation across varied scene initializations, we introduce a bimanual manipulation evaluation suite built on PerAct2 [4], where different initializations induce different functional

role assignments within the same task. The suite covers diverse bimanual coordination patterns and supports both balanced and imbalanced role distributions. Beyond average task success, we emphasize role-balanced robustness, which assesses whether a policy can adapt arm roles according to the scene rather than relying primarily on frequent arm-specific behaviors. Across these tasks, BiRoAD improves role-balanced robustness over corresponding base policies, with particularly clear gains on underrepresented and weaker-performing configurations. Together with real-world validation, these results suggest that bimanual policy learning benefits from explicitly structuring representations into shared coordination structure and role-specific distinctions. To summarize, in this paper:

- We study scene-conditioned role adaptation in bimanual manipulation, where varied scene initializations require policies to adjust functional role assignments across the two arms, especially under imbalanced role distributions.
- We propose BiRoAD, a modular symmetric–antisymmetric decomposition framework for learning shared and role-adaptive bimanual representations. BiRoAD can be integrated into existing policies as a residual feature transformation without requiring manual role labels.
- We introduce a bimanual manipulation evaluation suite with varied object initializations and balanced or imbalanced role distributions, and show that BiRoAD improves role-balanced robustness over corresponding base policies.

2 Related Work

2.1 Bimanual Manipulation

Bimanual manipulation expands robot capabilities by requiring spatial and temporal coordination between two arms [1, 3, 5, 14, 15, 16, 17, 18]. Recent progress has been driven by teleoperation systems, benchmarks, and learning-based policies. ACT [3] uses temporally chunked action prediction for fine-grained bimanual manipulation, while 3D FlowMatch Actor [9] predicts bimanual trajectories from 3D scene features using flow matching. Benchmarks [6, 19, 20, 21] such as PerAct2 [4] and RoboTwin [2] support evaluation across diverse bimanual tasks, scene configurations, and coordination requirements. Other studies have explored bimanual manipulation settings in which two arms assume complementary functional roles [22, 23, 24, 25, 26, 27, 28, 29, 10, 30]. VoxAct-B [7] considers scene-dependent acting and stabilizing roles, while EquiBim [31] enforces bilateral equivariance between symmetrically transformed observations and actions. Complementary to these approaches, BiRoAD does not require explicit role labels, predefined functional roles, or hard policy-level symmetry constraints. Instead, it studies how internal paired-arm representations can support scene-conditioned role adaptation across varying role configurations.

2.2 Imitation Learning for Robot Manipulation

Imitation learning enables robot policies to acquire manipulation skills from demonstrations, with recent progress in generative policy learning and vision-language-action policies [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]. Representative systems include ACT [3], which uses temporally chunked action prediction, and 3D FlowMatch Actor [9], which conditions flow-matching policies on 3D scene features. RDT-1B [12] scales diffusion-based imitation learning to bimanual manipulation with a large Robotics Diffusion Transformer and a unified action space across robot embodiments. Recent vision-language-action policies such as π_0 [11] and $\pi_{0.5}$ [13] further scale manipulation learning across diverse tasks and embodiments. BiRoAD further introduces a lightweight feature-level module that applies role-adaptive decomposition to paired arm features, supporting scene-conditioned role adaptation without changing the policy interface or imitation-learning objectives.

3 Problem Formulation

We consider imitation learning for bimanual manipulation from a dataset $\mathcal{D} = \{(\tau_i, l_i)\}_{i=1}^N$ of N expert demonstrations, where each demonstration consists of a trajectory τ_i and a task description

l_i . We represent each trajectory as $\tau_i = \{(o_t, s_t, a_t)\}_{t=1}^{T_i}$, where o_t denotes the scene observation, $s_t = (s_t^L, s_t^R)$ denotes the proprioceptive states of the left and right arms, and $a_t = (a_t^L, a_t^R)$ denotes their corresponding actions.

In $\mathcal{D}$, different scene initializations may induce different functional role assignments between the two arms. For a given task family, let ρ_L and ρ_R denote the frequencies with which the left or right arm assumes a task-leading or operating role, respectively. In practice, these frequencies can be imbalanced, *i.e.*, $\rho_L \neq \rho_R$, when some arm–role configurations occur more often than their mirrored or less common counterparts in the demonstration data. In our setting, different initializations of the same task share the same language instruction l , which specifies the task goal, such as “placing an object into a target drawer,” but does not specify the scene-dependent role assignment, such as which arm should open the drawer and which should grasp the object. The policy must therefore infer the appropriate role assignment from the scene observation when predicting paired arm actions, without relying on manual role annotations or additional role-assignment supervision.

4 Method

4.1 Overview

As illustrated in Figure 2, we propose BiRoAD, a modular feature decomposition method for bi-manual policies. BiRoAD operates on paired left- and right-arm action representations within the policy’s action-generation pathway. Given paired arm features $F = (f^L, f^R)$, BiRoAD decomposes them into two complementary components: a swap-invariant component S , satisfying $S(f^L, f^R) = S(f^R, f^L)$, which captures coordination structure shared across the two arms; and a swap-antisymmetric component A , satisfying $A(f^L, f^R) = -A(f^R, f^L)$, which captures relative, role-specific distinctions between the arms. The concrete construction is described in Section 4.2. BiRoAD then recomposes these components as residual updates to the original arm representations, preserving the policy interface while encouraging the action representation to model both shared bi-manual coordination and scene-dependent role variation. Therefore, BiRoAD can be integrated into existing policy architectures without changing their input–output interfaces or requiring additional role annotations. Representative integrations are discussed in Section 4.3.

4.2 Symmetric–Antisymmetric Role Decomposition

BiRoAD is designed to separate shared bimanual coordination from role-specific left–right distinctions within paired arm representations. We instantiate this idea as a residual feature transformation in the action-generation pathway. Given the left- and right-arm trajectory or action-token features f^L and f^R at a selected insertion point in the action-generation module, BiRoAD first maps the paired features into exchange-symmetric and exchange-antisymmetric coordinates:

$$m = \frac{1}{2}(f^L + f^R), \quad d = \frac{1}{2}(f^L - f^R). \quad (1)$$

The mean component m is exchange-invariant and captures information shared across the two arms, while the relative component d changes sign when the order of the two arms is reversed and therefore encodes their ordered left–right difference. The factors $1/2$ keep the transformed features on a scale comparable to the original representations and yield the inverse relation $f^L = m + d$, $f^R = m - d$.

BiRoAD then processes these coordinates with two branches. The symmetric branch is defined as

$$z_S = \phi_S(m), \quad (2)$$

where ϕ_S is implemented as an MLP. Since m is exchange-invariant, ϕ_S is also invariant to arm exchange, providing a pathway for modeling coordination structure shared across the arm pair.

The antisymmetric branch models directional differences between the two arm representations by applying an odd transformation to the relative feature:

$$z_A = \frac{1}{2}(\phi_A(d) - \phi_A(-d)), \quad (3)$$

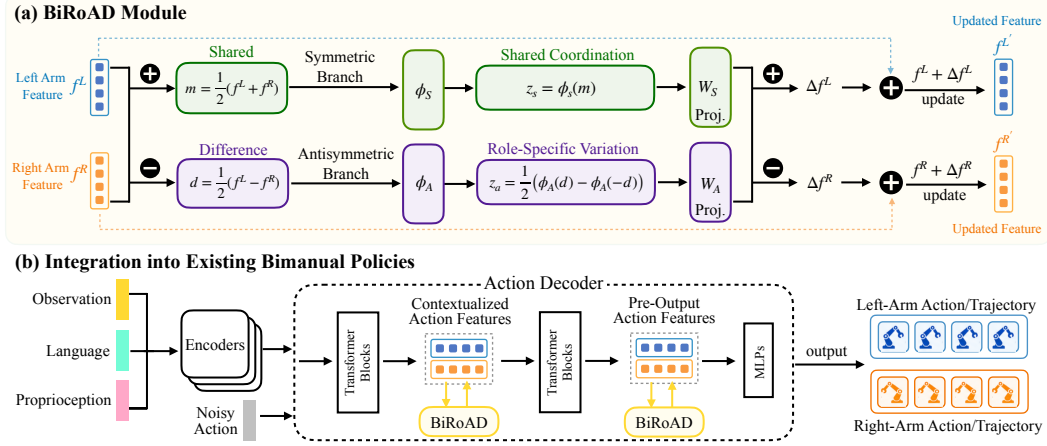


Figure 2: **BiRoAD module and policy integration.** (a) BiRoAD performs symmetric–antisymmetric feature decomposition and recomposes the resulting components as residual updates. (b) BiRoAD refines paired arm trajectory or action-token features at selected points in the action-generation module without changing the policy interface.

where ϕ_A is implemented as an MLP. This formulation does not require ϕ_A itself to satisfy any architectural symmetry constraint. Instead, antisymmetry is enforced by construction: reversing the ordered comparison maps d to $-d$, and therefore maps z_A to $-z_A$. The factor $1/2$ keeps the output scale comparable to the symmetric branch, although the antisymmetric property would hold without it. This branch provides a structured pathway for modeling role-specific differences that depend on the direction of the left–right comparison.

The two components are recomposed into arm-specific residual updates, with the symmetric component shared across arms and the antisymmetric component added with opposite signs:

$$\Delta f^L = W_S z_S + W_A z_A, \quad \Delta f^R = W_S z_S - W_A z_A, \quad (4)$$

where W_S and W_A are learnable projections. The sign pattern mirrors the symmetric–antisymmetric coordinates: $W_S z_S$ contributes a common residual to the arm pair, while $W_A z_A$ modulates their relative left–right variation and therefore enters the two residuals with opposite signs. The updated features are obtained by residual addition:

$$f^{L'} = f^L + \Delta f^L, \quad f^{R'} = f^R + \Delta f^R. \quad (5)$$

This residual formulation preserves the policy interface while incorporating both shared coordination structure and role-specific differences.

Overall, BiRoAD introduces a local exchange-consistent inductive bias within paired bimanual representations. It does not require the full policy to be strictly equivariant to arm exchange, since the architecture may still include ordered observations, fixed arm embeddings, or separate left- and right-arm action heads. Instead, BiRoAD applies this structure only to selected features in the action-generation pathway, making it a lightweight residual transformation that can be integrated into existing policies without changing their input–output interface or requiring role annotations.

4.3 Integration into Existing Bimanual Policies

We next describe how BiRoAD is integrated into the action-generation pathway of existing bimanual policy backbones. We instantiate the module in two representative policy families: a 3D flow-matching actor [9] and a π_0 -style vision-language-action policy [11].

3D flow-matching actor. The 3D flow-matching actor predicts bimanual trajectories by iteratively refining noisy action trajectories conditioned on language instructions and 3D scene representations. Its trajectory decoder progressively transforms task intent, spatial context, and denoising information into arm-specific trajectory features, making it a suitable location for BiRoAD. We therefore apply

BiRoAD directly to the corresponding left- and right-arm trajectory features at selected decoder stages, while leaving the surrounding decoder structure unchanged.

Concretely, we insert BiRoAD at three stages of the trajectory decoder: after the initial language and observation conditioning, after trajectory–scene feature interaction, and immediately before the final action prediction heads. These insertion points expose trajectory features at different levels of contextualization, enabling the residual symmetric–antisymmetric update to refine both shared coordination structure and role-specific distinctions during trajectory generation. Other components of the policy, including the encoders, action heads, and flow-matching objective, are kept unchanged.

π_0 -style VLA policy. The π_0 -style VLA policy factorizes action generation into a vision-language prefix and an action-generation suffix. The prefix encodes image and language observations into contextual representations, while the suffix combines this context with proprioceptive state, noisy action inputs, and denoising-time information to generate actions. We attach BiRoAD to the suffix rather than the prefix, since the suffix is where action representations are constructed and refined.

Within the suffix, we insert BiRoAD at two stages. First, we apply BiRoAD to the suffix hidden states after they have incorporated the vision-language context, proprioceptive state, timestep, and noisy action information. These hidden states serve as task- and observation-conditioned intermediate action representations, enabling the symmetric–antisymmetric decomposition to operate after the main conditioning signals have been fused. Second, in the output head, we apply BiRoAD to the per-arm latent features derived from the action-horizon hidden states before projecting them to the final action prediction. The vision-language pathway, action normalization, output convention, and training loss are kept unchanged.

Training. BiRoAD is trained jointly with the base policy using the original imitation-learning or flow-matching objective of the corresponding backbone. We do not introduce additional role labels, role-classification losses, or auxiliary supervision. The parameters of BiRoAD are optimized end-to-end through the same action prediction loss used by the underlying policy.

5 Experiments

5.1 Experimental Settings

Simulation Setup and Tasks. We build our simulation suite on RLBench [44], following the bimanual setup of PerAct2 [4] with two Franka Emika Panda arms. To evaluate scene-conditioned role adaptation, we adapt a set of PerAct2 bimanual task families into role-configurable settings, where different object layouts and interaction regions induce different functional role assignments between the two arms. We use eight task families and construct two configurations for each: a base-role configuration and a role-reversed configuration, yielding sixteen task variants in total. We further increase task randomization by broadening object initialization ranges. The full task list and implementation details are provided in the supplementary material. All visual observations are resized to 128×128 during training and evaluation. For each task family, we evaluate on 50+50 held-out episodes from the base-role and role-reversed configurations respectively. For training, we consider two demonstration compositions: a balanced 50+50 setting and a highly imbalanced 95+5 setting, to evaluate role-balanced learning and generalization to underrepresented role configurations.

Evaluation metrics. To assess whether a policy can adapt to different arm-role configurations, we first report the configuration-level average success rates under the base-role and role-reversed configurations. We further summarize performance with 4 metrics computed at the task level. For a task t , let s_t^{base} and s_t^{rev} denote its success rates under the two configurations. We compute *Mean*, *HM*, *Worst*, and *Gap* for this task pair, and report the average of each metric over all tasks. Here, *Mean* captures overall success across configurations, *HM* denotes their harmonic mean and emphasizes balanced performance, *Worst* measures the lower success rate; and *Gap* measures the absolute performance disparity between them. Since a small *Gap* may also result from uniformly low success rates, we interpret *Gap* together with the other metrics rather than as a standalone indicator of performance. Detailed metric definitions and formulas are provided in the supplementary material.

Table 1: Main simulation results under balanced and imbalanced training distributions.

Method	50:50					95:5				
	Config.-level Avg. $\uparrow$	Mean $\uparrow$	HM $\uparrow$	Worst $\uparrow$	Gap $\downarrow$	Config.-level Avg. $\uparrow$	Mean $\uparrow$	HM $\uparrow$	Worst $\uparrow$	Gap $\downarrow$
DP3	11.75 / 12.25	12.00	11.46	10.25	3.50	12.00 / 6.00	9.00	5.80	5.25	7.50
ACT	20.25 / 19.25	19.75	16.89	13.50	12.50	32.75 / 6.25	19.50	7.36	6.25	26.50
PerAct ²	22.00 / 21.25	21.62	18.78	16.00	11.25	13.25 / 19.75	16.50	13.11	12.00	9.00
π_0	18.75 / 24.00	21.38	20.64	18.25	6.25	23.25 / 14.00	18.62	15.69	13.00	11.25
π_0 + BiRoAD	26.00 / 27.25	26.62	26.37	24.75	3.75	36.75 / 17.75	27.25	21.23	17.75	19.00
3DFA	74.50 / 67.50	71.00	63.42	56.50	29.00	81.25 / 50.00	65.62	56.56	46.75	37.75
3DFA + BiRoAD	83.00 / 75.25	79.12	77.41	71.50	15.25	90.50 / 63.25	76.88	71.21	63.25	27.25

Table 2: Ablation study of BiRoAD on 3DFA under the 95:5 training distribution.

Method	Config.-level Avg. $\uparrow$	Mean $\uparrow$	HM $\uparrow$	Worst $\uparrow$	Gap $\downarrow$
3DFA (base)	81.25 / 50.00	65.62	56.56	46.75	37.75
+ BiRoAD (direct)	82.25 / 49.50	65.88	57.90	49.50	32.75
+ BiRoAD (early)	85.75 / 51.25	68.50	59.40	51.00	35.00
+ BiRoAD (middle)	85.00 / 51.75	68.38	60.75	51.75	33.25
+ BiRoAD (late)	82.25 / <u>59.75</u>	<u>71.00</u>	<u>65.45</u>	<u>58.00</u>	26.00
+ BiRoAD (middle+late)	86.50 / 55.25	70.88	63.29	55.25	31.25
+ Param.-matched MLP (full)	86.25 / 50.50	68.38	57.84	50.50	35.75
+ BiRoAD (full)	90.50 / 63.25	76.88	71.21	63.25	<u>27.25</u>

5.2 Main Results

Baselines. We select representative bimanual manipulation policies as baselines. For a fair comparison, all methods use the same observation resolution, action representation, and language setting. (1) **3DFA** [9], a strong 3D manipulation policy that predicts bimanual trajectories from 3D scene features using flow matching; (2) π_0 [11], a vision-language-action flow policy that generates robot actions through conditional denoising with visual, language, and proprioceptive context; (3) **ACT** [3], a transformer-based imitation-learning policy with temporally chunked action prediction; (4) **DP3** [38], a diffusion-based visuomotor policy that predicts action trajectories from 3D observations; and (5) **PerAct2** [4], a voxel-based bimanual manipulation policy built on PerAct [45]. Following its original protocol, PerAct2 is trained separately for each task (using both the base-role and role-reversed configurations), whereas the other baselines are trained in a multitask setting.

Analysis. Table 1 compares the policies under balanced and highly skewed training distributions. Across both settings, BiRoAD improves the two tested backbones in mean success rate, harmonic mean, and worst-group performance. Under the balanced 50:50 setting, BiRoAD increases the mean success rate for both the π_0 -style and 3DFA baselines, suggesting that the symmetric–antisymmetric decomposition can benefit policy learning when both role configurations are equally represented. The gains are more evident under the 95:5 setting, where policies must generalize to a less frequent role configuration. For 3DFA, BiRoAD improves the configuration-level averages from 81.25/50.00 to 90.50/63.25, with corresponding gains in harmonic mean and worst-group performance. BiRoAD also improves the π_0 -style backbone in absolute success rates. Although its Gap increases, both configuration-level averages improve, highlighting that Gap should be interpreted together with absolute success rates rather than in isolation. Overall, these trends suggest that BiRoAD improves role adaptation, especially on weaker or underrepresented configurations. Per-task success rates are provided in the supplementary material.

5.3 Ablation Study

To better understand the design choices in BiRoAD, we conduct ablations under the challenging 95:5 training distribution, where one role configuration is heavily underrepresented. All variants use the same 3DFA backbone, the best-performing backbone in the main comparison. We examine three design factors. First, *direct* replaces the original paired arm features with the BiRoAD features, whereas our default design uses BiRoAD as a residual update. Second, *early*, *middle*, and *late* insert BiRoAD at different trajectory-decoder stages: after initial language and observation conditioning, after trajectory–scene feature interaction, and before the final action prediction heads, respectively.

Table 3: Real-world success rates under balanced and imbalanced training distributions.

Method	Balanced Setting						Imbalanced Setting					
	Bouquet Handover	Banana to Drawer	Open Bottle Cap	Sweep to Dustpan	Pour Water	Overall	Bouquet Handover	Banana to Drawer	Open Bottle Cap	Sweep to Dustpan	Pour Water	Overall
3DFA	0.6 / 0.6	0.2 / 0.2	0.4 / 0.2	0.4 / 0.2	0.2 / 0.2	0.36 / 0.28	0.6 / 0.0	0.0 / 0.0	0.6 / 0.0	0.2 / 0.2	0.6 / 0.0	0.40 / 0.04
3DFA + BiRoAD	0.6 / 0.6	0.4 / 0.2	0.4 / 0.4	0.2 / 0.2	0.4 / 0.2	0.40 / 0.32	0.6 / 0.2	0.4 / 0.0	0.4 / 0.2	0.2 / 0.0	0.2 / 0.2	0.36 / 0.12
π_0	0.6 / 0.4	0.0 / 0.0	0.0 / 0.0	0.0 / 0.4	0.0 / 0.2	0.12 / 0.20	0.2 / 0.4	0.0 / 0.0	0.0 / 0.0	0.0 / 0.2	0.0 / 0.2	0.04 / 0.16
π_0 + BiRoAD	0.6 / 0.4	0.8 / 0.0	0.6 / 0.2	0.4 / 0.2	0.2 / 0.0	0.52 / 0.16	0.2 / 0.6	0.0 / 0.4	0.2 / 0.2	0.2 / 0.2	0.0 / 0.2	0.12 / 0.32

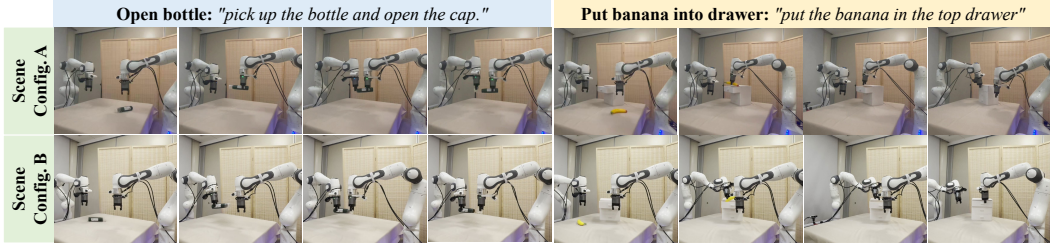


Figure 3: **Visualization of real-world experiments.** Different scene configurations induce different arm-role assignments, requiring the policy to adapt its bimanual behavior to the current scene.

The *middle+late* variant combines the latter two stages, while *full* applies BiRoAD at all three stages. Third, *Param.-matched MLP* uses the same multi-stage insertion pattern as *full*, but replaces the symmetric–antisymmetric decomposition with pairwise MLPs.

Table 2 reports the results. The relatively lower performance of *direct* suggests that BiRoAD is more effective when used as a residual refinement, which preserves backbone representations while introducing role-adaptive structure. Among single-stage variants, *late* achieves the strongest minority-configuration performance, indicating that role-aware coupling is especially useful near action prediction, where arm-specific pose outputs are produced. The *full* model achieves the strongest overall performance, suggesting that early and middle insertions help propagate role-aware features through the decoder, while late insertion shapes the final arm-specific actions. The *Param.-matched MLP* improves the frequent-configuration average but provides smaller gains on the minority configuration, suggesting that the exchange-structured decomposition is important beyond parameter count.

5.4 Real-World Experiments

We further evaluate BiRoAD in a real-world bimanual setup with two Franka Research 3 (FR3) arms equipped with DH-Robotics PGC-series parallel grippers. Observations are captured by one third-person and two wrist-mounted RealSense D405 cameras, and expert demonstrations are collected via human teleoperation. We consider five tasks with two scene configurations that induce different arm-role assignments. In the balanced setting, each method uses 25 demonstrations per configuration. In the imbalanced setting, we use opposite configuration ratios for the two backbone families, 25:5 for 3DFA and 5:25 for π_0 , to cover both directions of configuration imbalance. This design is motivated by the fact that the two configurations are not fully equivalent in the real setup due to the off-center third-person camera. We therefore focus on within-backbone comparisons between each base policy and its BiRoAD-enhanced counterpart. Each task is evaluated over 5 trials per configuration (10 trials per task), using task success rate as the metric. Figure 3 shows the real-world rollouts, and Table 3 reports the results. Overall, BiRoAD improves both base policies on real-world tasks that require scene-conditioned role adaptation. These trends are consistent with the simulation results.

6 Conclusion

In this work, we studied scene-conditioned role adaptation in bimanual manipulation, where different scene configurations induce different functional roles for the two arms. We proposed BiRoAD, a modular framework that decomposes paired arm representations into swap-symmetric components for shared coordination and swap-antisymmetric components for role-specific variation. Imple-

mented as a residual feature transformation, BiRoAD can be integrated into existing policies without modifying their interfaces, training objectives, or requiring manual role labels. Experimental results in simulation and the real world show that BiRoAD improves role-balanced robustness, particularly on weaker or underrepresented configurations, suggesting that structured shared and role-adaptive representations can benefit bimanual policy learning under scene-dependent role assignments.

Limitations. BiRoAD is a modular representation transformation that complements existing bimanual policies. Its benefits therefore depend on the representational and action-generation capacity of the underlying policy. This method does not directly address broader challenges in manipulation such as perception uncertainty, contact dynamics, or execution variability. Our evaluation focuses on two-arm tasks where scene configurations induce different functional roles between arms. Extending BiRoAD to multi-arm systems and dexterous hands would further test its generality.

Acknowledgments

This paper is supported by Beijing Natural Science Foundation (26L080330).

References

- [1] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Conference on robot learning*, pages 563–576. PMLR, 2023.
- [2] Y. Mu, T. Chen, Z. Chen, S. Peng, Z. Lan, Z. Gao, Z. Liang, Q. Yu, Y. Zou, M. Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the computer vision and pattern recognition conference*, pages 27649–27660, 2025.
- [3] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [4] M. Grotz, M. Shridhar, Y.-W. Chao, T. Asfour, and D. Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks. In *CoRL 2024 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2024.
- [5] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.
- [6] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.
- [7] I. Liu, C. Arthur, S. He, D. Seita, and G. Sukhatme. Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation. *arXiv preprint arXiv:2407.04152*, 2024.
- [8] J. Gao, X. Jin, F. Krebs, N. Jaquier, and T. Asfour. Bi-kvil: Keypoints-based visual imitation learning of bimanual manipulation tasks. In *2024 IEEE international conference on robotics and automation (ICRA)*, pages 16850–16857. IEEE, 2024.
- [9] N. Gkanatsios, J. Xu, M. Bronars, A. Mousavian, T.-W. Ke, and K. Fragkiadaki. 3d flowmatch actor: Unified 3d policy for single-and dual-arm manipulation. *arXiv preprint arXiv:2508.11002*, 2025.
- [10] G. Lu, T. Yu, H. Deng, S. S. Chen, Y. Tang, and Z. Wang. Anybimanual: Transferring unimanual policy for general bimanual manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13662–13672, 2025.
- [11] 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. *arXiv preprint arXiv:2410.24164*, 2024.
- [12] 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 *International Conference on Learning Representations*, volume 2025, pages 29982–30009, 2025.
- [13] Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- [14] 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.
- [15] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh. Learning bimanual scooping policies for food acquisition. *arXiv preprint arXiv:2211.14652*, 2022.
- [16] Y. Ren, Z. Zhou, Z. Xu, Y. Yang, G. Zhai, M. Leibold, F. Ni, Z. Zhang, M. Buss, and Y. Zheng. Enabling versatility and dexterity of the dual-arm manipulators: A general framework toward universal cooperative manipulation. *IEEE Transactions on Robotics*, 40, 2024.

- [17] Y. Zhao, R. Wu, Z. Chen, Y. Zhang, Q. Fan, K. Mo, and H. Dong. Dualafford: Learning collaborative visual affordance for dual-gripper manipulation. *arXiv preprint arXiv:2207.01971*, 2022.
- [18] Y. Shen, R. Wu, Y. Ke, X. Song, Z. Li, X. Li, H. Fan, H. Lu, et al. Biassemble: Learning collaborative affordance for bimanual geometric assembly. *arXiv preprint arXiv:2506.06221*, 2025.
- [19] Y. R. Wang, C. Ung, C. Tan, G. Tannert, J. Duan, J. Li, A. Le, R. Oswal, M. Grotz, W. Pumacay, et al. Roboeval: Where robotic manipulation meets structured and scalable evaluation. *arXiv preprint arXiv:2507.00435*, 2025.
- [20] X. Peng, C. Gao, L. Jin, A. Li, and S. Liu. Bicoord: A bimanual manipulation benchmark towards long-horizon spatial-temporal coordination. *arXiv preprint arXiv:2604.05831*, 2026.
- [21] F. Jiang, Y. Chen, K. Xu, Y. Liu, H. Wang, Z. Shen, J. Lu, S. Huang, Y. Wang, C. Xie, et al. Robowm-bench: A benchmark for evaluating world models in robotic manipulation. *arXiv preprint arXiv:2604.19092*, 2026.
- [22] Y. Shen, F. Jiang, Z. He, X. Li, Y. Liu, Z. Li, R. Wu, and H. Dong. Bipremanip: Learning affordance-based bimanual preparatory manipulation through anticipatory collaboration. *arXiv preprint arXiv:2603.21679*, 2026.
- [23] 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. *arXiv preprint arXiv:2501.14208*, 2025.
- [24] H. Zhou and K. Jia. Vlbiman: Vision-language anchored one-shot demonstration enables generalizable bimanual robotic manipulation. *arXiv preprint arXiv:2509.21723*, 2025.
- [25] A. Palma, I. Spinelli, V. Prasad, L. Scofano, Y. Jin, G. Chalvatzaki, and F. Galasso. Bimanual robot manipulation via multi-agent in-context learning. *arXiv preprint arXiv:2604.20348*, 2026.
- [26] J.-J. Jiang, X.-M. Wu, Y.-X. He, L.-A. Zeng, Y.-L. Wei, D. Zhang, and W.-S. Zheng. Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12427–12437, 2025.
- [27] J. J. Liu, Y. Li, K. Shaw, T. Tao, R. Salakhutdinov, and D. Pathak. Factr: Force-attending curriculum training for contact-rich policy learning. *arXiv preprint arXiv:2502.17432*, 2025.
- [28] A. Lee, I. Chuang, L.-Y. Chen, and I. Soltani. Interact: Inter-dependency aware action chunking with hierarchical attention transformers for bimanual manipulation. *arXiv preprint arXiv:2409.07914*, 2024.
- [29] J. Liang, Y. Chen, Q. Yu, Y. Shen, H. Zhang, H. Dong, and R. Wu. A3d: Adaptive affordance assembly with dual-arm manipulation. *arXiv preprint arXiv:2601.11076*, 2026.
- [30] T. Yu, G. Lu, Z. Yang, H. Deng, S. S. Chen, J. Lu, W. Ding, G. Hu, Y. Tang, and Z. Wang. Manigaussian++: General robotic bimanual manipulation with hierarchical gaussian world model. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12232–12239. IEEE, 2025.
- [31] Z. Zhang, A. Mohan, S. Han, W. Shou, D. Wang, and Y. She. Equibim: Learning symmetry-equivariant policy for bimanual manipulation. *arXiv preprint arXiv:2603.08541*, 2026.
- [32] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

- [33] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.
- [34] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.
- [35] Z. Li, J. Liu, D. Li, T. Teng, M. Li, S. Calinon, D. Caldwell, and F. Chen. Manidp: Manipulability-aware diffusion policy for posture-dependent bimanual manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9956–9962. IEEE, 2025.
- [36] G. Yan, J. Zhu, Y. Deng, S. Yang, R.-Z. Qiu, X. Cheng, M. Memmel, R. Krishna, A. Goyal, X. Wang, et al. Manifold: A general robot manipulation policy via consistency flow training. *arXiv preprint arXiv:2509.01819*, 2025.
- [37] 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. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.
- [38] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.
- [39] Y. Feng, H. Tan, X. Mao, C. Xiang, G. Liu, S. Huang, H. Su, and J. Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025.
- [40] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- [41] X. Li, L. Heng, J. Liu, Y. Shen, C. Gu, Z. Liu, H. Chen, N. Han, R. Zhang, H. Tang, et al. 3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation. In *9th Annual Conference on Robot Learning*, 2025.
- [42] X. Li, J. Xu, M. Zhang, J. Liu, Y. Shen, I. Ponomarenko, J. Xu, L. Heng, S. Huang, S. Zhang, et al. Object-centric prompt-driven vision-language-action model for robotic manipulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27638–27648. IEEE, 2025.
- [43] X. Li, M. Zhang, Y. Geng, H. Geng, Y. Long, Y. Shen, R. Zhang, J. Liu, and H. Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18061–18070. IEEE, 2024.
- [44] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.
- [45] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

A Real-World Experiments

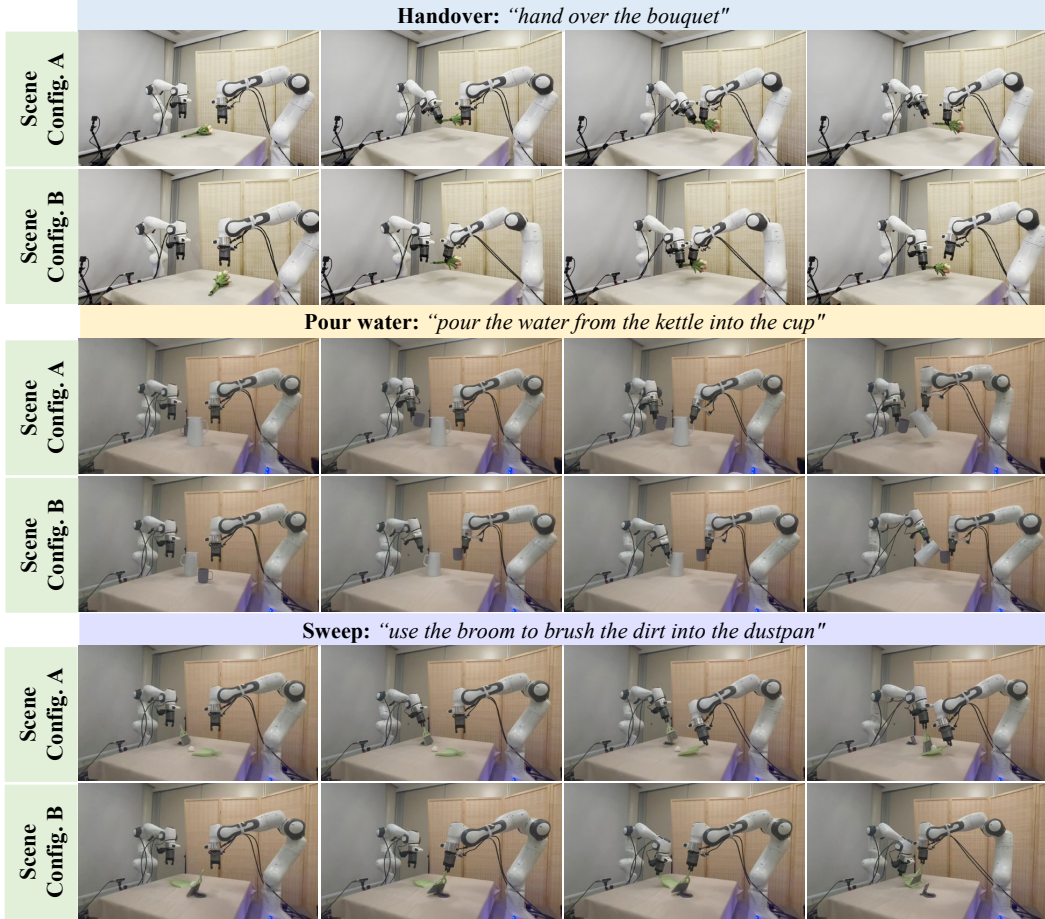


Figure 4: **Additional real-world experiments.** We visualize real-world task variants under different scene configurations, which induce different arm-role assignments and require scene-conditioned bimanual behavior.

We provide additional descriptions of the real-world tasks used in our experiments. The benchmark contains five bimanual tasks, each with two scene configurations, yielding a total of ten real-world task variants. Similar to the simulation benchmark, the two configurations of each task share the same task goal, while variations in object placement, tool placement, or workspace accessibility induce different functional role assignments between the two arms. Figure 4 shows the exact language instructions used for training and evaluation.

The following paragraphs provide additional clarification on the task procedure and role variations.

1. **Bouquet Handover.** One arm grasps the bouquet from the table, while the other arm receives it to complete the handover. The bouquet’s location differs between configurations, determining which arm is better suited for the initial grasping role.
2. **Put Banana into Drawer.** One arm opens the drawer, while the other grasps the banana and places it inside. Differences in banana and drawer layouts across configurations require the policy to select which arm should open the drawer and which should manipulate the banana.
3. **Open Bottle Cap.** One arm picks up the bottle, while the other grasps and removes the cap. Variations in bottle placement across configurations affect which arm is better positioned for picking up versus operating roles.

4. **Sweep to Dustpan.** One arm uses the broom to sweep the dirt, while the other holds the dustpan to collect it. The relative positions of the broom, dustpan, and dirt vary between configurations, requiring the policy to assign tool-use roles based on the current scene.
5. **Pour Water.** One arm grasps and tilts the kettle, while the other stabilizes or positions the cup. Differences in kettle and cup placement across configurations induce distinct role assignments for pouring and supporting.

B Simulation Task Gallery

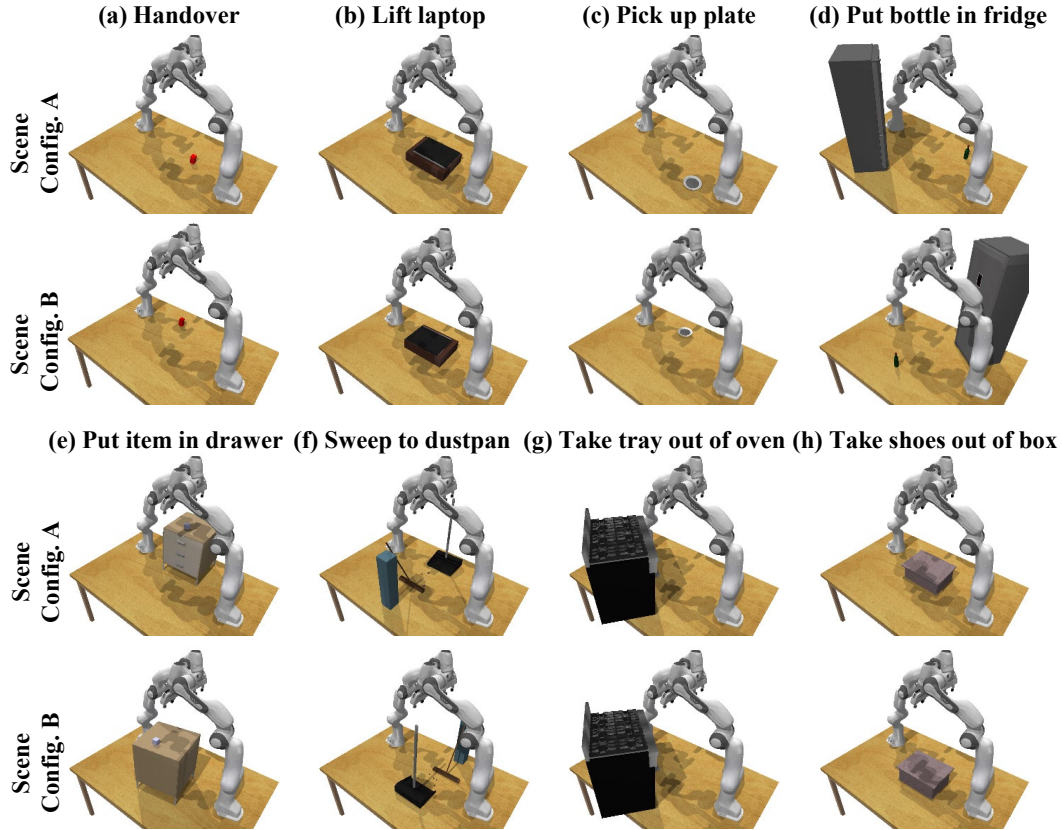


Figure 5: **Simulation task gallery.** We show eight simulated bimanual tasks under two scene configurations. The configurations preserve the task goal and language instruction while varying scene layout or object accessibility, inducing different arm-role assignments.

As illustrated in Figure 5, the benchmark comprises eight bimanual tasks. For each family, we design two distinct scene configurations, denoted Scene Config. A and Scene Config. B, resulting in a total of sixteen task variants. The following description is intended to clarify the task construction and how the two configurations induce different arm-role assignments, rather than to define the language instruction used for training and evaluation. Both configurations preserve the original task goal and language instruction but vary object layouts, target regions, tool placements, or articulated-object orientations to induce different functional role assignments between the two arms. In all cases, the policy must infer the appropriate arm-role assignment from the scene without access to explicit role labels.

The following paragraphs provide additional clarification on the task procedure and role variations.

1. **Handover.** The task involves transferring an object between the two arms. One arm grasps the object, while the other receives it to complete the handover. Compared with the original

task, we expand the object initialization range. In Scene Config. A, the object is initialized closer to one side of the workspace; in Scene Config. B, it is initialized closer to the opposite side. The two regions partially overlap near the center, preventing the policy from relying on a fixed arm identity and requiring it to decide which arm should initiate the grasp based on the current scene.

2. **Lift laptop.** The robot must lift a laptop from the table. One arm first pushes and adjusts the laptop into a graspable pose, while the other arm grasps and lifts it. The laptop position is randomized analogously to the handover task. The functional role assignment depends on which arm has better access to push the laptop and which arm is better positioned to grasp and lift it.
3. **Pick up plate.** The robot needs to pick up a plate from the table. One arm presses the plate edge to create clearance, while the other arm grasps and lifts it. Plate positions are randomized across the workspace. The policy must assign roles according to which arm can more conveniently press the plate edge and which arm can more reliably grasp and lift the plate.
4. **Put bottle in the fridge.** The scene contains a bottle and a fridge. One arm opens the fridge door, while the other arm grasps the bottle and places it inside. Scene configurations vary the relative accessibility of the bottle and fridge door, requiring the policy to infer which arm should operate the articulated door and which arm should manipulate the bottle.
5. **Put item in drawer.** The scene contains a small item and a drawer. One arm opens the target drawer, while the other arm grasps the item and places it inside. We increase drawer-orientation randomization by allowing additional rotation. This changes which arm can more conveniently open the drawer, so the policy must adapt the arm-role assignment according to the drawer pose and object layout.
6. **Sweep to dustpan.** The scene contains a broom, a dustpan, and target dirt. One arm holds and moves the broom to sweep the dirt, while the other arm holds the dustpan to collect it. Tool placements are randomized so that either arm may be closer to the broom or the dustpan. The policy must therefore assign the sweeping and supporting roles based on current tool locations rather than fixed arm identities.
7. **Take the tray out of oven.** The scene contains an oven and a tray inside it. One arm opens the oven door, while the other arm reaches into the oven and takes out the tray. Both arms may operate the oven door depending on the scene configuration. This requires the policy to choose which arm should open the door and which arm should extract the tray based on accessibility.
8. **Take shoes out of box.** The scene contains a hinged shoebox and a pair of shoes inside it. One arm opens the shoebox lid, while the other arm takes the shoes out of the box. The shoebox position is randomized similarly to the handover task. Because the box has a hinged structure, the easier opening direction depends on its pose and location; the policy must infer which arm should open the lid and assign the other arm to retrieve the shoes.

C Evaluation Metrics

We use task success rate as the primary evaluation metric, measured as the percentage of successful rollouts over held-out evaluation episodes. For each task, the evaluation set is balanced across role configurations, with 50 held-out episodes in the base-role configuration and 50 held-out episodes in the role-reversed configuration. Each episode is generated from randomized initial conditions. To ensure a fair comparison, all methods are evaluated on the same held-out episodes under the same evaluation protocol. This balanced evaluation design is independent of the training data ratio, e.g., (50:50) or (95:5), which only specifies the composition of the training demonstrations.

Beyond average task performance, we evaluate whether a policy performs consistently under the two role configurations. This is important because a policy may obtain a high overall success rate while still favoring one role configuration.

For each task t_i , we evaluate the policy under both the base-role and role-reversed configurations. Let S_i^{base} and S_i^{rev} denote the success rates of task t_i under these two configurations, respectively, and let N denote the total number of tasks. All metrics are computed separately for each training data-ratio setting.

- **Configuration-level success.** We report the average success rate for each role configuration:

$$\text{Base} = \frac{1}{N} \sum_{i=1}^N S_i^{\text{base}}, \quad \text{Rev.} = \frac{1}{N} \sum_{i=1}^N S_i^{\text{rev}}. \quad (6)$$

These metrics indicate whether a method achieves comparable performance under the base-role and role-reversed configurations.

- **Mean.** The overall mean success rate averages performance over the two role configurations for each task:

$$\text{Mean} = \frac{1}{N} \sum_{i=1}^N \frac{S_i^{\text{base}} + S_i^{\text{rev}}}{2}. \quad (7)$$

This metric summarizes overall performance across all evaluated role configurations.

- **Harmonic mean.** To emphasize balanced performance within each task, we report the harmonic mean between the two role configurations:

$$\text{HM} = \frac{1}{N} \sum_{i=1}^N \frac{2S_i^{\text{base}}S_i^{\text{rev}}}{S_i^{\text{base}} + S_i^{\text{rev}}}. \quad (8)$$

If $S_i^{\text{base}} + S_i^{\text{rev}} = 0$, the corresponding task-level harmonic mean is defined as zero.

- **Worst-configuration performance.** To measure performance under the weaker role configuration of each task, we compute

$$\text{Worst} = \frac{1}{N} \sum_{i=1}^N \min(S_i^{\text{base}}, S_i^{\text{rev}}). \quad (9)$$

This metric reflects the average success rate of the lower-performing role configuration across tasks.

- **Configuration gap.** Finally, we measure the average absolute performance gap between the two role configurations:

$$\text{Gap} = \frac{1}{N} \sum_{i=1}^N |S_i^{\text{base}} - S_i^{\text{rev}}|. \quad (10)$$

A smaller gap indicates more consistent performance between the base-role and role-reversed configurations. Since a small gap can also occur when both configurations have uniformly low success rates, we interpret Gap together with Mean, HM, and Worst rather than as a standalone measure of performance.

D Detailed Simulation Results

We provide the detailed per-task simulation results corresponding to the main results in the paper. Table 4 reports task success rates for each method under the balanced and imbalanced demonstration compositions. Each entry is shown as the success rate on the base-role configuration and the role-reversed configuration.

For fair comparison, all methods use the same data splits, observation resolution, action representation, language setting, and evaluation protocol. Except for PerAct², which follows its original

Table 4: Detailed per-task simulation success rates under balanced and imbalanced demonstration compositions. Each entry reports success rates on the base-role and role-reversed configurations, shown as base / reversed. For the 95:5 setting, the reversed configuration corresponds to the under-represented role configuration. **Bold** and underline indicate the best and second-best results within each training composition and configuration. PerAct² is single-task trained following its original protocol and is excluded from rank highlighting.

Balanced demonstrations: 50:50									
Method	Task Success Rate (% , base / reversed)								Config. Avg.
	Handover	Lift laptop	Pick up plate	Put bottle in fridge	Put item in drawer	Sweep to dustpan	Take tray out of oven	Take shoes out of box	
DP3	34 / 34	0 / 0	18 / 6	0 / 0	0 / 0	0 / 0	0 / 0	42 / 58	11.75 / 12.25
ACT	34 / 22	20 / 0	6 / 6	38 / <u>18</u>	0 / 2	10 / 8	0 / 2	54 / <u>96</u>	20.25 / 19.25
PerAct ²	10 / 28	60 / 20	68 / 60	0 / 2	36 / 42	0 / 12	2 / 6	0 / 0	22.00 / 21.25
π_0	52 / 62	24 / <u>26</u>	6 / 6	2 / 0	4 / 6	2 / 4	16 / 14	44 / 74	18.75 / 24.00
π_0 + BiRoAD	66 / 70	20 / 22	8 / 10	8 / 4	12 / <u>12</u>	6 / 10	24 / 18	64 / 72	26.00 / 27.25
3DFA	<u>86</u> / <u>94</u>	78 / 14	<u>64</u> / <u>74</u>	<u>88</u> / 8	<u>82</u> / 96	<u>46</u> / 68	<u>74</u> / 86	<u>78</u> / 100	<u>74.50</u> / <u>67.50</u>
3DFA + BiRoAD	92 / 98	<u>76</u> / 92	84 / 82	92 / 46	90 / 96	50 / <u>52</u>	80 / <u>76</u>	100 / 60	83.00 / 75.25
Imbalanced demonstrations: 95:5, reversed role underrepresented									
Method	Task Success Rate (% , base / reversed)								Config. Avg.
	Handover	Lift laptop	Pick up plate	Put bottle in fridge	Put item in drawer	Sweep to dustpan	Take tray out of oven	Take shoes out of box	
DP3	40 / 46	0 / 0	38 / 0	0 / 0	0 / 0	0 / 0	0 / 0	18 / 2	12.00 / 6.00
ACT	38 / 18	54 / 0	14 / 2	0 / 0	30 / 0	22 / 0	32 / 30	72 / 0	32.75 / 6.25
PerAct ²	8 / 22	6 / 22	78 / 78	0 / 2	10 / 0	0 / 24	4 / 10	0 / 0	13.25 / 19.75
π_0	70 / 50	28 / 2	26 / 14	2 / 0	8 / 0	2 / 0	28 / 36	22 / 10	23.25 / 14.00
π_0 + BiRoAD	76 / 56	36 / <u>22</u>	14 / 14	10 / 0	14 / 0	26 / 0	42 / 34	76 / 16	36.75 / 17.75
3DFA	100 / <u>84</u>	100 / 20	<u>64</u> / <u>44</u>	<u>80</u> / 22	<u>90</u> / <u>24</u>	<u>50</u> / 74	<u>88</u> / 90	<u>78</u> / <u>42</u>	<u>81.25</u> / <u>50.00</u>
3DFA + BiRoAD	<u>96</u> / 86	<u>96</u> / 78	<u>72</u> / 66	92 / 22	94 / 42	86 / <u>32</u>	94 / <u>86</u>	94 / 94	90.50 / 63.25

single-task training protocol, all other methods are trained in a multitask setting. Accordingly, rank-ing highlights in Table 4 are applied only among multitask-trained methods.

These results complement the aggregated metrics in the main paper by showing how each method performs across individual task families and role configurations.

E Additional Experiments

E.1 Data-Level Baselines

We further compare BiRoAD with two data-level strategies for addressing role imbalance on 3DFA under the 95:5 training split: arm-swap augmentation and minority oversampling.

Arm-swap augmentation. We augment the training data by swapping the left- and right-arm proprioceptive inputs, target actions, and corresponding wrist-camera observations with probability 0.5. This baseline examines whether exposure to swapped arm-specific signals can reduce bias toward fixed arm identities.

Minority oversampling. We rebalance training by oversampling the minority role-reversed demonstrations, giving the two role configurations equal expected sampling frequency. This baseline examines the effect of increasing training exposure to the underrepresented role configuration.

Table 5: Comparison with data-level baselines on 3DFA under the 95:5 training split.

Method	Config.-level Avg. $\uparrow$	Mean $\uparrow$	HM $\uparrow$	Worst $\uparrow$	Gap $\downarrow$
3DFA	81.25 / 50.00	65.62	56.56	46.75	37.75
+ arm-swap augmentation	86.50 / 57.25	71.88	64.48	55.75	32.25
+ minority oversampling	78.25 / 55.75	67.00	63.15	55.25	23.50
+ BiRoAD	90.50 / 63.25	76.88	71.21	63.25	27.25

Results. Table 5 shows that arm-swap augmentation improves performance on both role configurations, while minority oversampling improves the underrepresented configuration at a small cost to majority performance. BiRoAD achieves the strongest overall performance, highlighting the benefit of explicitly structuring shared and role-specific representations for role adaptation under imbalance.

E.2 Branch Representation Analysis

We further examine whether the symmetric and antisymmetric branches exhibit the intended complementary structure, and how strongly they contribute during inference.

Linear probes. We train task-specific linear probes on frozen projected symmetric and antisymmetric features to predict the functional role of each arm. The antisymmetric features achieve task-averaged accuracies of 98.8% and 98.1% under the 50:50 and 95:5 settings, respectively, while the symmetric features yield 50.0% in both settings. These results show that functional-role information is linearly accessible from the antisymmetric branch, consistent with its intended role-specific function.

Branch activation. As shown in Table 6, both branches exhibit non-negligible activation magnitudes during inference, approximately 10–20% of the input-feature magnitude. Both α_S and α_A increase under the imbalanced 95:5 setting compared with the balanced 50:50 setting, suggesting greater reliance on BiRoAD under role imbalance. Within the 95:5 setting, α_A is higher for the minority role-reversed configuration than for the base-role configuration (0.182 vs. 0.144), indicating a stronger role-specific contribution for the underrepresented configuration.

Table 6: Relative branch activation magnitudes for 3DFA + BiRoAD. α_A and α_S denote the projected antisymmetric and symmetric contributions relative to the input-feature magnitude.

Configuration	50:50		95:5	
	α_A	α_S	α_A	α_S
Base-role	0.108	0.127	0.144	0.196
Role-reversed	0.113	0.126	0.182	0.184

F Training and Inference Cost

We report the training and inference costs of the base policies and their BiRoAD-enhanced variants. For the 3DFA backbone, both the base policy and the BiRoAD-enhanced policy are trained from scratch. The 3DFA base policy was trained on 2 A100 GPUs for 500k iterations, requiring approximately 29 hours, with a peak memory usage of 5.54 GB per GPU. Under the same GPU configuration and number of training iterations, the 3DFA + BiRoAD model required approximately 30 hours, with a peak memory usage of 5.56 GB per GPU.

For the π_0 backbone, both the base policy and the BiRoAD-enhanced policy are fully fine-tuned from the same pretrained checkpoint for 50k iterations on 4 A100 GPUs. Reported results use the best-performing checkpoints within this budget, selected at 30k iterations for the base policy and 50k for π_0 + BiRoAD. The π_0 base policy required approximately 19 hours for the full 50k iterations, with a peak memory usage of 32.7 GB per GPU. Under the same GPU configuration and training

budget, the π_0 + BiRoAD model required approximately 19.5 hours, with a peak memory usage of 32.6 GB per GPU.

During inference, BiRoAD adds only a lightweight symmetric–antisymmetric decomposition to the bimanual trajectory or action-token representations, while leaving the backbone encoders and the denoising schedule unchanged. As a result, its inference latency remains comparable to that of the base policy.