2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

PA-BiCoop: A Primary-Auxiliary Cooperative Framework for General Bimanual Manipulation

Qicheng Bai[1], Ziru Wang[1], Teli Ma[2], Guang Dai[1], Jingdong Wang[3] and Mengmeng Wang[4,1,*]

***Abstract*— Bimanual manipulation is essential for advanced robotic systems because it offers higher efficiency and flexibility compared to single-arm configurations. However, existing approaches either lack inter-arm interaction or ignore the need for a dynamic division of labor, treating the arms as functionally equivalent. To address these limitations, this paper draws inspiration from human bimanual manipulation where one arm handles core operations and the other provides auxiliary support, and proposes PA-BiCoop, a new single-model bimanual cooperation framework with dynamic primary-auxiliary arm differentiation. PA-BiCoop categorizes robotic arms into primary and auxiliary arms with adaptively adjustable roles across task stages, employs two specialized decoders that share a global feature encoder: the primary decoder generates the primary arm's base-coordinate pose and core-task affordance heatmaps, and the auxiliary decoder outputs the auxiliary arm's relative pose in the primary arm's coordinate system. Moreover, we design a dynamic role assignment module to automatically map roles to left/right arms without manual pre-definition. This design facilitates inter-arm knowledge sharing and coordinated manipulation. Extensive experiments demonstrate that our PA-BiCoop achieves superior performance: it outperforms state-of-the-art baselines by 48% on average in RLBench2 simulation tasks and by over 50% on average in real-world tasks, thereby verifying its effectiveness and advancement in bimanual manipulation.**

I. Introduction

Bimanual manipulation [1]–[4] has become an indispensable component in advanced robotic systems, owing to its superior efficiency and operational flexibility compared to unilateral (single-arm) configurations [5]–[9]. Notably, it enables the execution of tasks that are inherently unattainable for single-arm setups, such as transporting large objects, removing bottle caps, or assembling complex components. Compared to single-arm manipulation, bimanual operation poses significantly greater challenges, as it requires simultaneously modeling the states and actions of both arms, along with their collaborative interactions.

As shown in Fig. 1, the predominant approaches in this domain can be categorized into two architectural paradigms currently. The first employs dual independent models to predict the movements of each arm separately, with representative methods including AnyBimanual [10], VoxActB [11], and BUDS [12]. While these approaches draw extensively on the research findings and technical advancements in single-arm modeling, they essentially adopt an independent modeling framework where each arm is assigned its own dedicated model. Therefore, this paradigm has a core limitation: it lacks inter-arm knowledge sharing and interactive information transfer, thereby compromising overall task performance. Additionally, the use of two separate models inevitably doubles the model complexity during both training and inference phases. In contrast, the second paradigm adopts a single shared modeling framework for the joint modeling of both arms, as exemplified by YOTO [13], PerAct2 [14], and Kstar Diffuser [15]. However, it can be observed that despite leveraging a single shared model, these methods usually treat the left and right arms as functionally equivalent without distinguishing their roles. They generate movements for the two arms in one of two rigid modes: either asynchronously following a manually predefined sequence [13] or simultaneously outputting the action spaces of both arms [14]–[17].

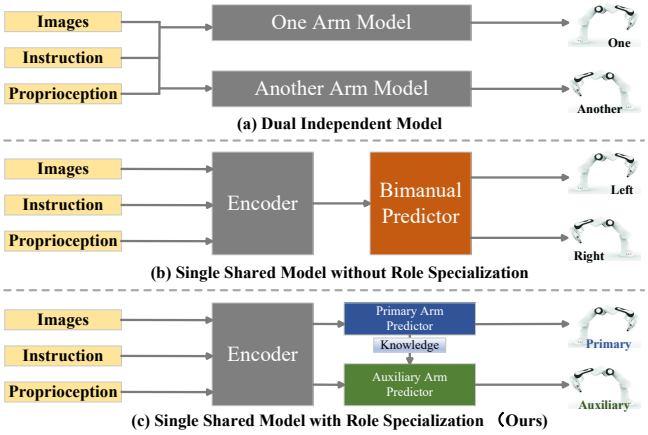

Fig. 1. Three model paradigms for bimanual manipulation.

To illustrate this limitation, this paper draws inspiration from human bimanual manipulation: when humans perform tasks such as assembling parts, wrapping packages, or using tools, their two arms rarely act in a "role-agnostic" manner. Instead, one primary arm typically takes charge of core operations. Examples include aligning components or wielding a tool. Meanwhile, the other auxiliary arm provides auxiliary support, such as stabilizing the workpiece or handing over materials. Crucially, this role division is not fixed, as the left and right arms can dynamically switch roles depending on the task demands. This inherent division of labor and adaptive collaboration is precisely what is missing in the aforementioned shared-model paradigms, ultimately leading to suboptimal coordination efficiency in complex bimanual

This work was supported by Zhejiang Province Natural Science Foundation of China under Grant No. LQN25F030008 and the National Natural Science Foundation of China under Grant No. 62403429.

[1]SGIT AI Lab, State Grid Corporation of China.
[2]The Hong Kong University of Science and Technology, Guangzhou.
[3]Baidu Research, Beijing, China.
[4]Zhejiang University of Technology, Hangzhou.
*Corresponding Author.

tasks.

Motivated by this human bimanual manipulation mechanism, we introduce **PA-BiCoop**, a new single-model **Bi**manual **Coop**eration framework for bimanual manipulation that incorporates **P**rimary-**A**uxiliary differentiation. Unlike previous single-model frameworks, our approach dynamically assigns distinct roles and collaborative functions to the two robotic arms as shown in Fig. 1 (c): we categorize the robotic arms as primary arm and auxiliary arm, with the distinction being non-fixed and adaptively adjusted between the two arms at different task stages. For the primary arm, we employ a primary decoder to generate main affordance heatmaps indicating the current primary task region and output the primary arm's pose under the base coordinate system. For the auxiliary arm, we propose an auxiliary decoder that outputs the auxiliary arm's relative pose in the coordinate system of the primary arm based on the primary arm's affordance heatmaps and global features. These two decoders share a global feature encoder, allowing for the sharing of global information without the need for redundant perception models and thus maintaining the simplicity of the overall architecture. This design of primary and auxiliary decoders facilitates inter-arm knowledge sharing. Furthermore, we design a role assignment module that dynamically maps the primary and auxiliary roles to the left and right arms, eliminating the need for pre-defined manual role sequences. Evaluation across both simulation tasks and real-world tasks demonstrates the effectiveness of our PA-BiCoop, which achieves substantial advancements over prior methods.

Our contributions can be summarized as follows:

- We propose PA-BiCoop, a new single-model bimanual framework that enables dynamic primary-auxiliary specialization, enhancing collaboration through role-aware design.
- We introduce dedicated primary and auxiliary decoders for action prediction and inter-arm interaction, along with a learnable role assignment module enabling automatic role specialization.
- Extensive experiments show that PA-BiCoop outperforms state-of-the-art methods by 48% on RLBench2 [14] and over 50% in real-world tasks, achieving superior overall success rates.

II. Related Work

Current methodologies in bimanual manipulation can be broadly categorized into two architectural paradigms: dual independent models and single shared models.

Dual-Model Architecture. Several approaches mitigate coordination challenges through dual-model architectures [10], [11], [18]–[24]. Early methods [20]–[25] propose to create a "leader and follower" movement, suffering from large memory consumption and fixed roles. Methods such as BUDS [12] and VoxActb [11] decouple the system into stabilizing and acting arms. VoxActb [11] employs VLMs [26], [27] for scene region prioritization and voxel grid reconstruction. However, they lack flexible role switching between arms and are limited to relatively simple tasks, constraining their performance in complex scenarios. Any-Bimanual [10] proposes a model-agnostic plug-and-play framework that generalizes pretrained single-arm policies [5], [6]. Although it uses attention partitioning to isolate arm-specific regions of interest, this design also restricts knowledge sharing between arms. Overall, these methods underutilize inter-arm interaction, limiting their effectiveness in tasks requiring high coordination.

Single-Model Architecture. Alternative approaches employ a single-model framework [13]–[17], [28]–[35] to leverage inter-arm knowledge sharing. Early work such as ACT [17] uses conditional VAEs [36] within an encoder-decoder structure for joint angle prediction. PerAct2 [14] extends PerAct [6] by duplicating its prediction head for simultaneous dual-arm control. InterACT [29] effectively captures dependencies between joint states and visual inputs through a hierarchical attention mechanism. KStar Diffuser [15] incorporates spatiotemporal graphs and differentiable kinematics to guide diffusion models. A common limitation among these methods is their treatment of both arms as functionally equivalent without explicit role specialization, thereby overlooking inherent asymmetries in task division. YOTO [13] employs a single prediction head for bimanual action output but depends on handcrafted coordination sequences, resulting in poor synchronous performance. Although unified in structure, these methods generally fail to explicitly model bimanual coordination, making them susceptible to failure from minor motion discrepancies.

In this paper, we propose a new approach that incorporates dynamic primary–auxiliary arm differentiation within a single shared model. This design enhances cross-arm knowledge sharing and interaction, facilitating high-precision coordination in complex bimanual tasks.

III. Method

A. Problem Formulation

Our goal is to learn a model $a = f_\theta(o, l, p)$ that can complete various bimanual manipulation tasks, where o contains RGB-D images from multiple perspectives, l is the language description of the task, p is the proprioception of two robot arms, and the output a is the actions of two robot arms. The action of each arm consists of the 6-DoF end-effector pose (3-DoF for translation and 3-DoF for rotation), 1-DoF gripper state (open or close), and a binary indicator for whether to allow collision avoidance for the motion planner. In addition, referring to prior works [5], [6], [14], [37], we employed key-frame extraction technology to enhance learning efficiency. Thus, the model only needs to predict action at the next key-frame.

B. Framework

Inspired by the inherent division of labor in human bimanual manipulation, where one arm typically takes the lead in core tasks while the other provides auxiliary support, we analogously categorize the robotic arms in our system into **Primary Arm** and **Auxiliary Arm**. The general pipeline of our proposed PA-BiCoop is shown in Fig. 2. Considering

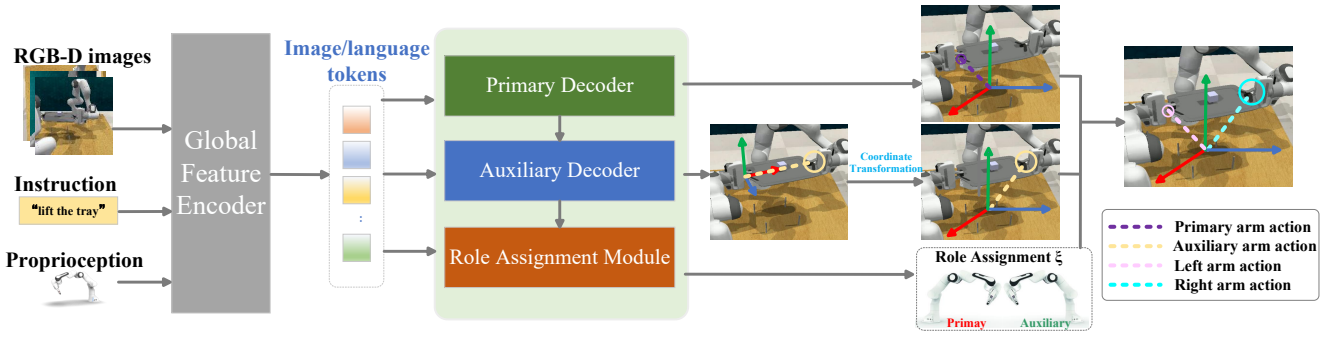

Fig. 2. The framework of PA-BiCoop. Given RGB-D images, instruction, and proprioception, we encode them through a global feature encoder to image/language tokens. The primary decoder, auxiliary decoder, and role assignment module processes these tokens to produce the primary action a_{bc}^P, the auxiliary action a_{pc}^A, and the role assignment ξ. a_{pc}^A is transformed to a_{bc}^A through coordinate transformation. ξ maps primary/auxiliary arms to left/right arms.

both the efficiency of feature extraction and the effectiveness of global feature representation [5]–[7], [38], the RVT [5] is employed as the global shared feature encoder, which processes o, l, and p to produce image tokens and language tokens. These tokens are subsequently decoded by the primary decoder, auxiliary decoder, and role assignment module to output the primary arm action a_{bc}^P, the auxiliary arm action a_{pc}^A, and the role assignment variable ξ respectively.

The primary arm's action a_{bc}^P is generated in the base coordinate system C_{bc} using the global feature representation, leveraging its precise multi-view perception capability. The auxiliary arm's action a_{pc}^A is predicted in the primary arm's coordinate system C_{pc}, dependent on the primary arm's kinematic state. This approach exploits the relative positioning between arms during bimanual coordination tasks, reducing the burden on the auxiliary arm's spatial perception and the complexity of coordination. The role assignment module predicts ξ to enable dynamic role switching: $\xi[0]$=1 designates the left arm as the primary arm (right as the auxiliary arm), while $\xi[1]$=0 reverses this mapping, allowing context-dependent role switching. The resulting a_{pc}^A is transformed to the action a_{bc}^A in the coordinate system C_{bc} using coordinate transformation. Subsequently, we will provide a detailed explanation of the primary decoder, the auxiliary decoder, the role assignment module, the coordinate transformation, and the training loss functions.

C. Primary Decoder

As shown in Fig. 3 (a), the primary decoder uses the global features to generate affordance heatmaps and predict the primary arm's actions. First, we feed image and language tokens into a multi-layer perceptron (MLP) to adjust features. Subsequently, we utilize eight transformer layers (each comprising multi-head self-attention and an MLP) to focus on the task region and output image features across different views. The image features are then processed through two convolutional layers to generate the main affordance heatmaps H_T^P, which indicate the current primary task region for the primary arm across different views, with peak values corresponding to the desired translation coordinates of the primary arm. To meet the accuracy requirements for primary arm prediction in image-patched tokens, we adopt bilinear interpolation [39] to upsample multi-view tokens to their original spatial resolution between convolutional layers. For rotation and other action variables, we use joint features. The joint features are a concatenation of (1) the sum of image features along the spatial dimensions, weighted by H_T^P; and (2) the max-pooled image features along with the spatial dimension. We employ an MLP to process joint features, and further predict a discrete probability distribution P_R^P for Euler angles (discretized into bins with a resolution of 5°) as well as other binary action variables V_O^P. For the primary arm action a_{bc}^P, we perform argmax on H_T^P and P_R^P to output it.

D. Auxiliary Decoder

As shown in Fig. 3 (b), our auxiliary decoder is designed to predict the actions of the auxiliary arm by leveraging both global features and contextual information from the primary decoder. To distill essential information from the primary decoder outputs, we extract the 2D points V_T^P across different views from H_T^P and three Euler angles V_R^P from P_R^P with the highest score. These features are concatenated with V_O^P to form a token encapsulating the most salient aspects of a_{bc}^P. After projection through an MLP, this token is concatenated with a learnable query embedding representing a_{pc}^A to construct action tokens. For image/language tokens, we first apply an MLP to extract features relevant to the auxiliary arm's operational regions, preserving the primary arm's prediction integrity.

The decoder has one cross-attention layer and six self-attention layers. Due to the input of the primary arm action knowledge and the prediction in C_{pc}, the auxiliary decoder only needs to perform the cross-attention between action tokens and image/language tokens. Since these tokens are much smaller than the image/language tokens, they can ignore unimportant areas in the image and retain the features of the key regions while reducing the computational cost. For the final output, these tokens pass through MLP to predict a_{pc}^A. Note that a_{pc}^A contains 2D continuous points across different views $\hat{V}_T^A$ for translation, three continuous Euler angles $\hat{V}_R^A$ for rotation, and other action variables V_O^A.

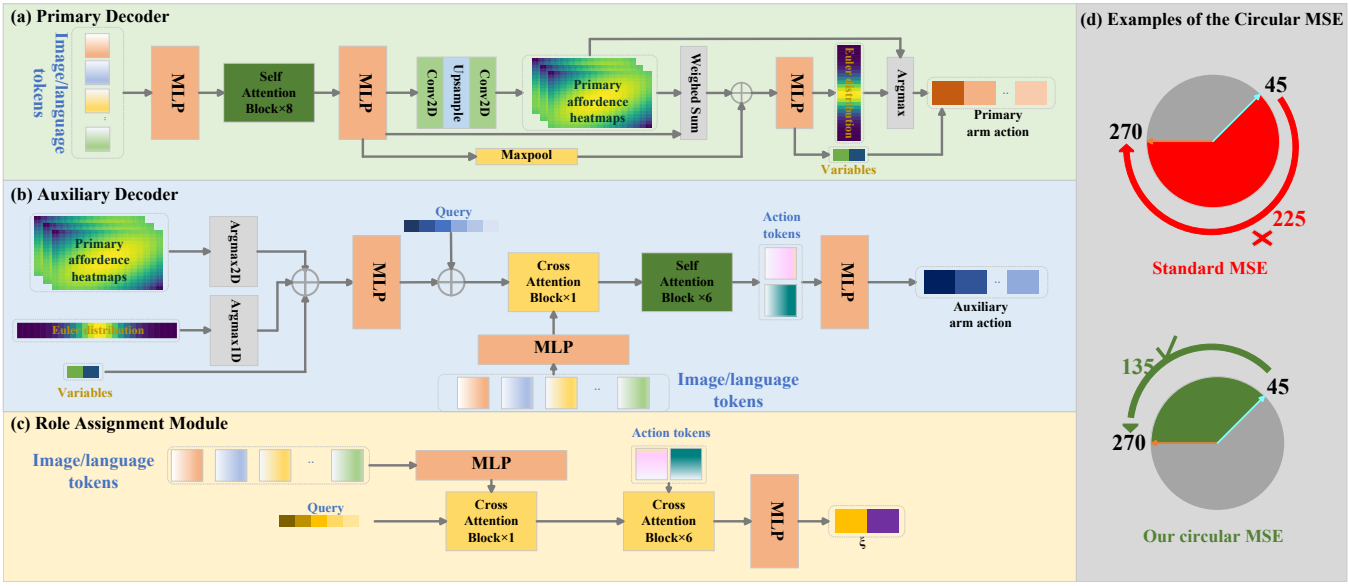

Fig. 3. (a) **The primary decoder.** It primarily employs self-attention blocks, convolutional layers, and MLPs to generate the main affordance heatmaps and ultimately predict actions for the primary arm based on global image/language tokens. (b) **The auxiliary decoder.** This component consists mainly of cross-attention blocks, self-attention blocks, and MLPs, which utilize outputs from the primary decoder along with global image/language tokens to predict actions for the auxiliary arm. (c) **The role assignment module.** Comprising cross-attention blocks and MLPs, this module determines the functional role specialization of the two robotic arms based on the global image/language tokens and the action tokens from the auxiliary decoder. (d) **Examples of the circular MSE.** The circular MSE correctly computes the angular distance between $45°$ and $270°$ (green arc), whereas standard MSE yields an invalid result (red arc).

E. Role Assignment Module

The structure of the proposed role assignment module is illustrated in Fig. 3 (c). We initialize a learnable query embedding with random values to represent the role-specific latent variable ξ. The contextual information derived from both image and language plays a critical role in determining the functional division between the two arms. To achieve dynamic and context-aware role switching, we first compute cross-attention between the role query and the image/language tokens processed by MLP. This interaction enables the query to attend to the semantically salient regions and instructions. We further apply six cross-attention blocks between the role query and the action tokens, thereby aligning role assignment with intended motor actions. Finally, the fused representation is projected through an MLP to produce a binary classification variable, which explicitly dictates the arm assignment under the current perceptual and semantic context.

F. Coordinate Transformation

a_{pc}^A obtained from the auxiliary decoder is in the coordinate system C_{pc}. However, execution requires its representation in the base coordinate system C_{bc}. We therefore transform a_{pc}^A from C_{pc} to C_{bc} through the following operations:

$$V_R^A = \Gamma^{-1}\left(\Gamma(V_R^P) \cdot \Gamma(\hat{V}_R^A)\right) \tag{1}$$

$$V_T^A = V_T^P + \hat{V}_T^A \tag{2}$$

where Γ [40] denotes the Euler-angle-to-rotation-matrix transformation, V_R^A represents the continuous Euler angle in C_{bc}, and V_T^A corresponds to multi-view 2D points in C_{bc}. This transformation yields the base-coordinate action a_{bc}^A. Noted that V_T^A and V_T^P are back-projected to the 3D point for translation in evaluation. Finally, a_{bc}^P and a_{bc}^A are mapped to the left/right arm according to ξ.

G. Loss Function

We train PA-BiCoop using a group of losses. For the primary arm action prediction, we compute cross-entropy losses for H_T^P, P_R^P, and V_O^P:

$$\mathcal{L}^P = CE(H_T^P, Y_T^P) + CE(P_R^P, Y_R^P) + CE(V_O^P, Y_O^P) \tag{3}$$

where ground truth Y_T^P denotes the 2D points across different views projected from the ground-truth 3D point of the primary arm, while Y_R^P and Y_O^P represent rotation, gripper state, and whether to allow collision avoidance.

For the auxiliary arm, we extract corresponding ground truths Y_T^A, Y_R^A, and Y_O^A from datasets. Since V_R^A is a periodic continuous variable on $[0°, 360°]$, the general mean squared error (MSE) fails, as shown in Fig. 3 (d). To address this limitation, we introduce a circular MSE loss:

$$\mathcal{L}_{rot}^A = ||\min(|V_R^A - Y_R^A|, 360 - |V_R^A - Y_R^A|)||^2 \tag{4}$$

The auxiliary arm's total action loss combines:

$$\mathcal{L}^A = MSE(V_T^A, Y_T^A) + CE(V_O^A, Y_O^A) + \kappa\mathcal{L}_{rot}^A \tag{5}$$

where the scaling factor $\kappa = 1/360$ (default) normalizes the magnitudes of $\mathcal{L}_{rot}^A$. The role assignment loss is:

$$\mathcal{L}^\xi = CE(\xi, Y_\xi) \tag{6}$$

where Y_ξ represents the ground truth. Finally, the training loss of PA-BiCoop is as follows:

$$\mathcal{L}^{total} = \mathcal{L}^P + \lambda_A\mathcal{L}^A + \lambda_\xi\mathcal{L}^\xi \tag{7}$$

TABLE I

THE EXPERIMENTAL RESULT IN SIMULATION. WE TRAIN ALL POLICIES BASED ON 10 OR 100 TRAINING DEMONSTRATIONS, AND EVALUATE ON THE SAME 25 EPISODES OF THE TEST SET. AS KSTAR DIFFUSER [15] HAS NOT RELEASED THE CODE, WE REPORT ITS PERFORMANCE METRICS FROM THE ORIGINAL PUBLICATION.

Method	Architectural	Avg.	Push Box	Lift Ball	Lift Tray	Put in Drawer	Pick Plate	Pick Laptop	Sweep Duspan	Handover (easy)	Put in Bridge	Take out Tray
20 demos												
RVT-LF [14]	Dual-model	2.4	12	4	0	8	0	0	0	0	0	0
PerAct-LF [14]	Dual-model	3.6	8	4	0	16	0	0	8	0	0	0
AnyBimanual [10]	Dual-model	11.6	28	4	4	12	12	8	16	24	0	8
ACT [17]	Single-model	4	0	24	4	12	0	0	0	0	0	0
PerAct2 [14]	Single-model	4	0	8	0	12	4	0	8	8	0	0
Kstar Diffuser [15]	Single-model	-	80	87	-	-	-	17	83	24	-	-
PA-BiCoop (ours)	Single-model	**61.6**	**84**	**100**	**80**	**80**	**24**	**44**	**100**	**36**	**32**	**36**
100 demos												
RVT-LF [14]	Dual-model	10	52	16	8	12	4	4	0	0	0	4
PerAct-LF [14]	Dual-model	20	56	40	16	28	4	12	28	8	0	8
AnyBimanual] [10]	Dual-model	20	24	32	12	20	32	8	32	28	8	4
ACT [17]	Single-model	6	0	36	8	12	0	0	0	0	0	4
PerAct2 [14]	Single-model	14	8	52	4	12	4	12	0	40	4	8
Kstar Diffuser [15]	Single-model	-	83	98	-	-	-	44	89	27	-	-
PA-BiCoop (ours)	Single-model	**68.8**	**88**	**100**	**88**	**60**	**40**	**48**	**96**	**52**	**48**	**68**

with λ_A, λ_ξ as task-balancing hyperparameters.

IV. EXPERIMENTS

A. Experiment Settings

Simulation. Bimanual manipulation tasks present significantly greater challenges than their single-arm counterparts due to stringent requirements for coordination, synchronization, and symmetry awareness between dual robotic arms. To evaluate the capabilities of PA-BiCoop in these complex regimes, we employ the RLBench2 [14] benchmark encompassing 10 distinct language-conditioned tasks. This suite explicitly includes synchronous, asynchronous, symmetric, and asymmetric manipulation scenarios. For training of PA-BiCoop, we implemented a ξ annotation schema in the dataset to explicitly designate left/right arm roles (primary vs. auxiliary) for model guidance. For environmental observation, a multi-camera system comprising six RGB-D cameras (resolution: 128 × 128) provides comprehensive coverage of the entire workspace. During policy training, each task is supported by 20 or 100 expert demonstrations. Evaluation rigor is ensured by executing 25 episodes per task within the RLBench2 [14] testing set, mitigating stochastic variance.

Real-World. For real-world validation, we employ a dual-arm system comprising two Yahboom DoFbot manipulators and design two representative tasks: `handover` and `grasp banana`. Training data are collected using Moveit in ROS2 to control the robotic arms for demonstration recording. Observations are provided by a calibrated front-facing RGB-D camera capturing images at a resolution of 1280 × 720. We gather 15 demonstration trajectories per task. For evaluation, each task is executed over 10 trials using an NVIDIA RTX 4090 GPU.

Baseline. We evaluate PA-BiCoop against state-of-the-art bimanual manipulation methods, categorized as follows:

(1) Dual-model: RVT-Leader Following (RVT-LF) [14] employs an RVT [5] backbone with a leader-follower mechanism; Perceiver-Actor Leader Following (PerAct-LF) [14] applies a similar leader-follower paradigm using PerAct [6]; AnyBimanual [10] transfers pre-trained PerAct [6] via a skill manager and visual aligner module.

(2) Single-model: Action Chunking with Transformers (ACT) [17] uses a Conditional VAE [36] for joint angle sequence prediction; PerAct2 [14] enhances PerAct by unifying dual-arm action prediction within a shared feature space; Kstar Diffuser [15] is a generative model predicting kinematics-aware actions using a physics-grounded spatial-temporal graph to condition denoising.

TABLE II

ABLATION STUDIES. WE EVALUATED THE INFLUENCE OF DECODER, C_{pc}, AND ξ ON PERFORMANCE ACROSS THREE REPRESENTATIVE TASKS.

Decoder	C_{pc}	ξ	Push Box	Put in Drawer	Take out Tray	Avg.
Primary-Auxiliary	**Y**	**Y**	**88**	**60**	**68**	**72**
Primary-Auxiliary	Y	N	88	60	28	58.7
Primary-Auxiliary	N	Y	44	32	68	48
Primary-Auxiliary	N	N	44	32	28	34.7
Primary-Primary	N	N	20	24	12	18.7

TABLE III

THE RESULTS IN THE REAL WORLD.

Method	Avg.	Grasp Banana	Handover
ACT [17]	5	10	0
PerAct2 [14]	35	30	40
PA-BiCoop	**85**	**90**	**80**

Implementation Details. Following PerAct2 [14], we implement $\mathrm{SE}(3)$ observation augmentation for expert training demonstrations to enhance model robustness. All comparative methods undergo standardized training: 100k iterations on NVIDIA A100 GPUs with a global batch size of 64. Model optimization employs the LAMB optimizer [41] with an initial learning rate of 5×10^{-4}, utilizing a cosine decay schedule with a linear warmup phase spanning 3k iterations.

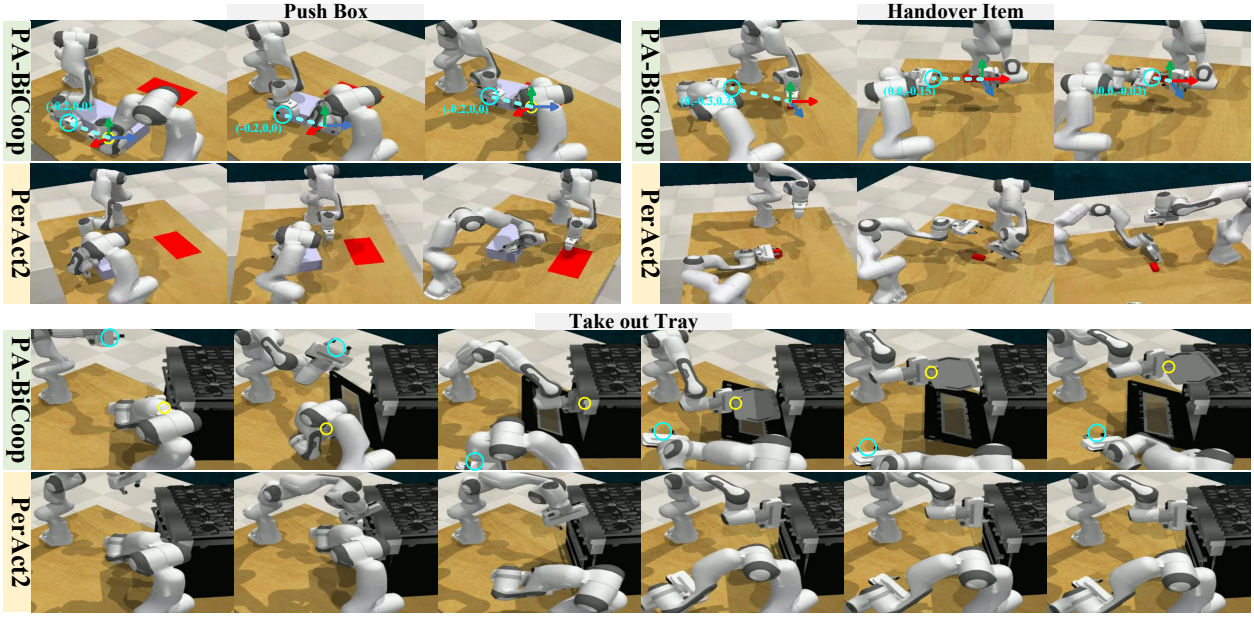

Fig. 4. The visualization on RLBench2. The yellow circles represent the primary arm actions, the blue circles represent auxiliary arm actions, and the vector beside the blue circle is the translation vector of the auxiliary arm in C_{pc}.

B. Simulation Results

TABLE I presents comparative results between PA-BiCoop and state-of-the-art methods under 20 and 100 demonstration training regimes. PA-BiCoop achieves an overall performance improvement of at least 48% against all baselines. We observe that:

(1) For symmetrical and synchronous tasks (e.g., `push box`, `lift ball`, `lift tray`), temporal misalignment induces insufficient force application or object instability, resulting in task failure. Our relative coordinate system prediction mechanism ensures operational stability, achieving success rates exceeding 84%. In the `lift ball`, we even achieved a 100% success rate.

(2) For asynchronous and asymmetric tasks such as `put in drawer` and `handover (easy)`, where bimanual coordination complexity increases, PA-BiCoop maintains at least 4% superiority over baselines despite performance attenuation. In the `put in drawer` task, the auxiliary arm dynamically determines which drawer to open based on linguistic instructions. Despite this requirement, PA-BiCoop achieves a success rate of at least 60%, demonstrating our auxiliary decoder's strength in spatial relationship perception.

(3) Furthermore, PA-BiCoop demonstrates substantial advancement in long-horizon tasks (`put in fridge`, `take out tray`), elevating success rates from under 10% to over 45% based on 100 training demonstrations. This performance leap is directly attributable to our observation-driven role switching mechanism.

Across all evaluated tasks, Kstar Diffuser [15] and PerAct2 [14] demonstrate significantly lower success rates than PA-BiCoop despite employing a comparable single shared model for bimanual motion prediction. This performance gap originates from their treatment of the left and right arms as functionally equivalent, without distinguishing their respective roles, thereby failing to capture the critical aspects of bimanual manipulation. These results validate the effectiveness of PA-BiCoop in achieving synergistic bimanual coordination.

C. Ablation Studies

In this section, we evaluate the core proposed components of our PA-BiCoop via ablation studies in TABLE II on representative tasks: `push box`, `put in drawer`, and `take out tray`.

Effects of primary-auxiliary. We designed a primary-primary architecture, in which each arm's actions were processed independently by a dedicated primary decoder, to validate the efficacy of our primary-auxiliary cooperative framework. Due to the absence of role specialization and knowledge sharing between the arms, this model exhibits a significant decline in performance, with the average success rate across various tasks dropping by more than 16%.

Effects of C_{pc}. Predicting the auxiliary arm pose in the coordinate system C_{bc} results in a 24% reduction in average success rates. This performance degradation occurs because the auxiliary arm pose exhibits substantial variations in this coordinate system, which substantially increases the learning complexity of the auxiliary arm pose estimation. Conversely, our PA-BiCoop predicts the auxiliary arm pose in C_{pc}, where the pose is defined relative to the precise primary arm pose and remains invariant to manipulated object displacements. This stability proves particularly beneficial in synchronous tasks, where constant spatial relationships between the primary and auxiliary arms significantly reduce bimanual coordination complexity. Our ablation study conclusively demonstrates the efficacy of the design C_{pc} for coordinated bimanual manipulation.

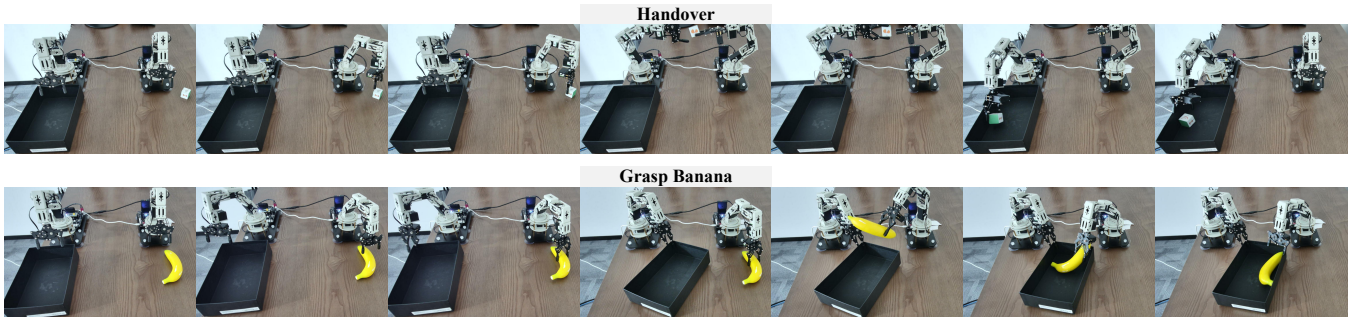

Fig. 5. The visualization of our PA-BiCoop in real-world tasks using two Yahboom DoFbot manipulators.

Effects of ξ. Our experiments reveal that in the long-horizon `take out tray` task, success rate declines to 28% without observation-driven role switching between arms. This performance degradation occurs because restricting the primary role to a single arm during extended tasks reduces manipulation precision for the other arm operation. Consequently, our ablation studies validate the effectiveness of ξ in resolving this limitation.

D. Qualitative Analysis

As shown in Fig. 4, we evaluate PA-BiCoop against PerAct2 [14] across RLBench2 [14] benchmarks, conducting detailed qualitative assessments on synchronous, asynchronous, and long-horizon tasks.

In the `push box` task, both arms can serve as the primary arm. Crucially, the relative pose between arms remains constant throughout pushing the box. The action prediction of the auxiliary arm in C_{pc} enables it to have the same high accuracy as the action prediction of the primary arm, demonstrating exceptional robustness and environmental adaptability. Conversely, PerAct2's [14] requires both arms to perform independent object recognition, preventing effective synchronization and consequently achieving suboptimal success rates.

In the `handover item` task, the arm nearer to the target item is designated as the primary arm, while the other arm is the auxiliary arm to hand over the item. Equipped with precise detection and positioning capabilities, our primary decoder enables the primary arm to accurately grasp the object. During handover, the auxiliary arm operates without the need for repeated precise object detection. Instead, it simply approaches the origin of C_{pc} in C_{pc}, thereby significantly reducing the coordination complexity between the two arms. In contrast, PerAct2 requires both arms to perform precise object detection and localization, which substantially increases the computational burden on the model. Even minor inaccuracies in this process can lead to failures in the `handover item` task.

In the `take out tray` task, PA-BiCoop dynamically adapts role between arms: designating the right arm as the primary arm during oven-door operation, then switching primary role to the left arm for tray extraction. This adaptive coordination enables seamless bimanual cooperation. PerAct2 [14] fails to model such coordination mechanisms, resulting in significantly degraded long-horizon task performance.

E. Real-World Results

To further validate the effectiveness of our PA-BiCoop , we evaluate it on two real-world tasks, both requiring spatiotemporal coordination between the arms. Performance results are summarized in TABLE III. Despite the limited number of demonstrations in the training datasets, our PA-BiCoop outperforms existing approaches by over 50%. Although PerAct2 [14] and ACT [17] also employ a single shared model, they do not differentiate the roles of the two arms and exhibit limited capability in tasks demanding precise bimanual coordination. Qualitative results of PA-BiCoop on both tasks are visualized in Fig. 5. The `handover` and `grasp banana` tasks involve placing objects into a fixed or movable box, and both arms must operate in coordination due to the limitation of spatial distance. Our PA-BiCoop facilitates knowledge exchange between arms, dynamically assigns complementary roles, and achieves a high degree of coordination, leading to successful task completion.

V. Conclusion

This paper addresses key challenges in bimanual manipulation, where previous models lack inter-arm interaction and ignore the dynamic division of labor. Drawing on human bimanual mechanisms, we propose PA-BiCoop, a single-model framework with dynamic primary-auxiliary arm differentiation: it includes two specialized decoders that share a global encoder, the primary decoder generates the primary arm's base-coordinate pose and core-task affordance heatmaps, while the auxiliary decoder outputs the auxiliary arm's relative pose. It also has a dynamic role assignment module that automatically maps primary or auxiliary roles to left and right arms without manual pre-definition. Extensive experiments demonstrate PA-BiCoop's efficacy in complex bimanual manipulation, confirming it enhances coordination efficiency and adaptability and lays a foundation for robust robotic bimanual systems.

Limitations and prospects for the future. Currently, PA-BiCoop faces difficulties in extremely long-horizon tasks, e.g., multi-step assembly involving dozens of sequential operations or tasks with extended pauses. In future work, we will aim to optimize the role assignment module by

incorporating task stage prediction capabilities or introducing a lightweight task memory mechanism.

REFERENCES

[1] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotic mechanisms," *The International Journal of Robotics Research*, vol. 8, no. 2, pp. 44–61, 1989.

[2] E. Paljug, X. Yun, and V. Kumar, "Control of rolling contacts in multi-arm manipulation," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 441–452, 2002.

[3] N. Sarkar, X. Yun, and V. Kumar, "Dynamic control of 3-d rolling contacts in two-arm manipulation," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 364–376, 1997.

[4] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation—a survey," *Robotics and Autonomous systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[5] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 694–710.

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

[7] H. Fang, M. Grotz, W. Pumacay, Y. R. Wang, D. Fox, R. Krishna, and J. Duan, "Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation," in *International Conference on Machine Learning (ICML)*, 2025.

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

[9] 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.

[10] G. Lu, T. Yu, H. Deng, S. S. Chen, Y. Tang, and Z. Wang, "Anybimanual: Transferring unimanual policy for general bimanual manipulation," in *Conference on Computer Vision (ICCV)*, 2025.

[11] I.-C. A. Liu, S. He, D. Seita, and G. S. Sukhatme, "Voxact-b: Voxel-based acting and stabilizing policy for bimanual manipulation," in *Conference on Robot Learning (CoRL)*, 2025, pp. 4354–4370.

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

[13] 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 *Robotics: Science and Systems (RSS)*, 2025.

[14] M. Grotz, M. Shridhar, Y.-W. Chao, T. Asfour, and D. Fox, "Peract2: Benchmarking and learning for robotic bimanual manipulation tasks," in *Conference on Robot Learning (CoRL)*, 2024.

[15] Q. Lv, H. Li, X. Deng, R. Shao, Y. Li, J. Hao, L. Gao, M. Y. Wang, and L. Nie, "Spatial-temporal graph diffusion policy with kinematic modeling for bimanual robotic manipulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 17 394–17 404.

[16] 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," *arXiv preprint arXiv:2410.07864*, 2024.

[17] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[18] D. Sirintuna, I. Ozdamar, and A. Ajoudani, "Carrying the uncarriable: a deformation-agnostic and human-cooperative framework for unwieldy objects using multiple robots," in *the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[19] J. Gao, X. Jin, F. Krebs, N. Jaquier, and T. Asfour, "Bi-kvil: Keypoints-based visual imitation learning of bimanual manipulation tasks," in *the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 850–16 857.

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

[21] Y. Zhou, M. Do, and T. Asfour, "Coordinate change dynamic movement primitives—a leader-follower approach," in *the IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 5481–5488.

[22] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[23] Y. Zhou, M. Do, and T. Asfour, "Learning and force adaptation for interactive actions," in *the IEEE-RAS international conference on humanoid robots (humanoids)*. IEEE, 2016, pp. 1129–1134.

[24] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023.

[25] M. Zhang, P. Jian, Y. Wu, H. Xu, and X. Wang, "Dair: Disentangled attention intrinsic regularization for safe and efficient bimanual manipulation," *arXiv preprint arXiv:2106.05907*, 2021.

[26] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[27] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[28] 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.

[29] A. C.-W. Lee, I. Chuang, L.-Y. Chen, and I. Soltani, "Interact: Inter-dependency aware action chunking with hierarchical attention transformers for bimanual manipulation," in *Conference on Robot Learning (CoRL)*. PMLR, 2025, pp. 1730–1743.

[30] D. Yu, H. Xu, Y. Chen, Y. Ren, and J. Pan, "Bikc: Keypose-conditioned consistency policy for bimanual robotic manipulation," *arXiv preprint arXiv:2406.10093*, 2024.

[31] 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," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2804–2818, 2023.

[32] K. F. Gbagbe, M. A. Cabrera, A. Alabbas, O. Alyunes, A. Lykov, and D. Tsetserukou, "Bi-vla: Vision-language-action model-based system for bimanual robotic dexterous manipulations," in *the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2024, pp. 2864–2869.

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

[34] R. Caccavale, M. Saveriano, G. A. Fontanelli, F. Ficuciello, D. Lee, and A. Finzi, "Imitation learning and attentional supervision of dual-arm structured tasks," in *the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2017, pp. 66–71.

[35] L. X. Shi, A. Sharma, T. Z. Zhao, and C. Finn, "Waypoint-based imitation learning for robotic manipulation," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 2195–2209.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[37] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 18 081–18 090.

[38] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 1199–1210.

[39] E. J. Kirkland, "Bilinear interpolation," in *Advanced computing in electron microscopy*. Springer, 2010, pp. 261–263.

[40] A. Mueller, "Modern robotics: Mechanics, planning, and control," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 100–102, 2019.

[41] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," in *International Conference on Learning Representations (ICLR)*, 2020.