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

Offline reinforced finetuning for chunk-based VLA via real-world RL policy distillation with vision-guided copilot

Yihao Wu[1*], Zhenjun Yu[2*], Shun Yin[3], Junbo Tan[1†], Zhihao Wang[4†], Xueqian Wang[1]

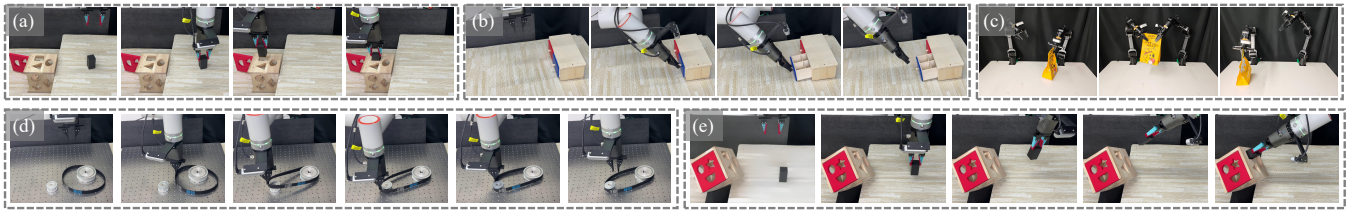


Fig. 1: Overview of experimental tasks. A subset of tasks considered in this paper, they include (a) insert cube from top, (b) pull the drawer, (c) object handover, (d) install the belt, (e) insert cube from side. Extensive experiments demonstrate our outstanding performances on real-world fine tasks with few human efforts.

***Abstract*— Pre-trained VLA do not fully retain their strong performance when fine-tuned for new tasks, hindering robust deployment in new environments. This limitation primarily stem from the constraints of prevalent fine-tuning approaches: supervised fine-tuning (SFT) requires large amounts of high-quality demonstrations; reinforcement learning (RL) is often limited by dataset quality, sparse reward, and the sim-to-real gap. To overcome these limitations, we propose a novel framework that leverages sample-efficient real-world RL to collect data for offline distillation into the VLA. Our method introduces three key components: (1) Vision-guided Copilot that refines human actions toward expert-level action using visual feedback to improve intervention quality and data efficiency. (2) CopRL, a human-in-the-loop RL framework that leverages the Copilot for efficient online exploration and data collection with minimal human intervention; and (3) RaoRFT, an offline RL algorithm that distills high-quality reward-annotated trajectories from CopRL into the VLA. Real-world experiments show our method achieves state-of-the-art performance with minimal human input. Our work provides a practical and effective pathway for deploying high-performance VLA in complex manipulation tasks. Codes and models will be available.**

I. INTRODUCTION

Vision-Language-Action models (VLA) [1]–[4] have shown great potential for a wide range of applications. However, pre-trained VLA often face challenges in maintaining their high performance when adapted to new tasks or environments. This limitation arises because these models, despite their initial success, struggle to generalize effectively to previously unseen situations, thereby hindering their ability to be reliably deployed in real-world applications where variability and unpredictability are common.

One common strategy is supervised fine-tuning (SFT) [5], [6]. Despite its effectiveness, this approach requires large amounts of high-quality demonstration data—a costly and time-consuming process when collected via standard teleoperation. Another alternative is reinforcement learning (RL), which offers a pathway for policy improvement without relying on pre-collected demonstrations. However, RL is often limited by dataset quality, sparse rewards, and the sim-to-real gap. Some methods [7]–[10] attempt to overcome data constraints through simulation-based online RL, though sim-to-real transfer remains challenging. Others [11], [12] explore offline RL with real-world rollouts, yet still depend heavily on human effort to curate suitable data and often fail to produce sufficient high-reward trajectories for effectively finetuning large-scale VLA.

To address these challenges, this paper introduces a novel framework that uses sample-efficient real-world RL to collect interaction data for distillation into the VLA via offline RL fine-tuning. Our approach begins with a **Vision-guided Copilot**, a diffusion-based system that refines human actions toward expert behaviors using visual feedback to improve intervention quality and data efficiency.

By integrating the Vision-guided Copilot into human-in-the-loop interventions, we develop **CopRL**—a real-world RL framework that achieves higher performance with less human effort compared to [13]. CopRL begins with offline pretraining on a minimal set of mixed expert and VLA demonstrations. It then proceeds to online exploration, during which occasional human corrections with the Copilot are stored and used to update the policy. This process forms an efficient data collection cycle. The resulting reward-annotated trajectories generated by CopRL are then utilized by **RaoRFT**, an offline RL algorithm designed to distill high-quality, reward-annotated trajectories from CopRL into the chunk-based VLA. Specifically, RaoRFT employs a stage-based dense reward annotation scheme and advanced offline RL algorithms to fine-tune the VLA, thereby yielding a superior policy that requires minimal human input.

Extensive real-world experiments demonstrate that our method requires only a small number of initial human demonstrations and minimal online interventions to achieve state-of-the-art performance, outperforming methods that rely solely on offline data or require substantially more

This work was supported by the Natural Science Foundation of Shenzhen (No.JCYJ20230807111604008, No. JCYJ20240813112007010)the Natural Science Foundation of Guangdong Province (No.2024A1515010003) and Cross-disciplinary Fund for Research and Innovation (No. JC2024002) of Tsinghua SIGS.

[1]Center for Intelligent Control and Telescience, Tsinghua Shenzhen International Graduate School, Shenzhen, China; [2]Shanghai Jiao Tong University, Shanghai, China; [3]South China Normal University, Guangzhou, China; [4]Meituan, China.

*Indicates equal contribution. †Corresponding author: tjblql@sz.tsinghua.edu.cn, wangzhihao32@meituan.com

human cost.

Our contribution can be summarized as follows:

- **Vision-guided Copilot.** We design a Vision-guided Copilot, a diffusion-based module that adjusts human teleoperation action inputs toward expert-level share action using visual feedback. This approach significantly enhances both the quality of interventions and the efficiency of data collection in delicate manipulation.
- **CopRL framework.** We propose CopRL(**Cop**ilot-guided online **RL**), a novel framework that integrates offline pretraining with online fine-tuning. By incorporating the Vision-guided Copilot into human-in-the-loop interventions during the online phase, our method achieves sample-efficient real-world RL and greatly reduces the human cost required for subsequent data collection.
- **RaoRFT algorithm.** We develop RaoRFT(**R**eward-**a**nnotated **o**ffline **R**einforced **F**ine-**T**uning), an offline RL method that distills high-quality, reward-annotated trajectories from CopRL into the VLA, featuring a stage-based dense reward annotation scheme for richer learning signals. This approach enhances training efficiency and effectiveness through multi-round training, maintaining a continuous pipeline of data collection and policy refinement.

II. RELATED WORKS

A. Real-world RL and Human Assitances

Sample efficiency and safety are paramount concerns for deploying RL in physical systems. Prior works have addressed these challenges through various paradigms. Frameworks like [14]–[17] demonstrate that careful pipeline design enables practical real-world RL, while other works [18], [19] use human-in-the-loop preference learning or offline Q-learning with human advice rewards for continous action spaces. [13] incorporates human interventions to provide high-quality corrective signals, guiding exploration and significantly accelerating learning. These approaches highlight the value of human oversight in mitigating risks and improving data efficiency.

The quality of human input, however, remains a critical bottleneck. Noisy or suboptimal interventions can hinder learning progress. Methods like [20]–[22] aim to extract clearer expert intent from human demonstrations, but the conditions are mainly based on robot states, which are lack of accurate perception in relatively hard tasks. Building on this concept, our Vision-guided Copilot integrates real-time visual feedback to more effectively refine human teleoperation inputs, ensuring each intervention provides maximally informative guidance for the RL policy and further boosting the efficiency of the human-in-the-loop data collection process.

B. VLA Post-training and Finetuning

Adapting general-purpose Vision-Language-Action (VLA) models to specific downstream tasks primarily follows two lines of work: SFT and RL. SFT [5], [6] requires large volumes of high-quality, task-specific demonstration data, which is often costly and time-consuming to acquire. Its performance is inherently limited by the quality and coverage of the static dataset.

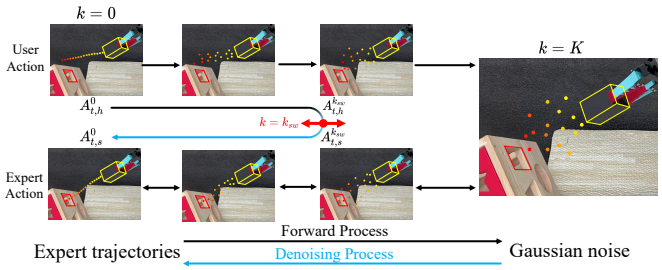


Fig. 2: The principle of the Vision-guided Copilot. The black arrow at the top indicates the forward diffusion process applied to the user action, while the blue arrow at the bottom represents the reverse denoising process toward the target expert action. We switch between these two processes at the intermediate step $k = k_{sw}$, resulting in a hybrid procedure that combines partial forward diffusion and denoising, as depicted by the central black and blue arrows.

Offline RL presents a powerful alternative for policy improvement without online interaction. Methods like [11], [12] apply offline RL to refine VLA on fixed datasets. Recent work [23] also explores adaptive offline RL post-training for VLA flow models, balancing RL signal and variance. However, a fundamental constraint of these methods is their dependence on a pre-collected, static dataset; their performance ceiling is bounded by its quality and diversity.

Some efforts [7]–[10] seek to bypass this limitation by training in simulation. While valuable for algorithmic exploration, the sim-to-real transfer gap often remains a significant hurdle for real-world deployment. In contrast, our RaoRFT algorithm is designed to leverage a continuous stream of high-quality, reward-annotated data generated autonomously in the real world by our CopRL framework. This self-improving data cycle effectively breaks the constraint of static datasets, enabling sustained policy improvement through offline RL distillation.

III. METHOD

Our pipeline, illustrated in Fig. 3, integrates two key components: (1) **CopRL** begins with offline pretraining on a small mixed dataset, followed by online fine-tuning where a **Vision-guided Copilot** refines human interventions into expert-level corrections to update the policy. (2) **RaoRFT** subsequently distills reward-annotated trajectories from CopRL via offline reinforcement fine-tuning into the chunk-based VLA. This cohesive pipeline seamlessly combines offline pretraining, human-in-the-loop online collection, and efficient VLA finetuning, drastically reducing human effort and time cost.

A. Vision-guided Copilot

Inspired by [20], we extend this approach by integrating visual information to train a diffusion-based network called **Vision-guided Copilot**, which takes both vision signals and robot state information to assist human teleoperations. This allows the copilot to be trained on high-variance data,

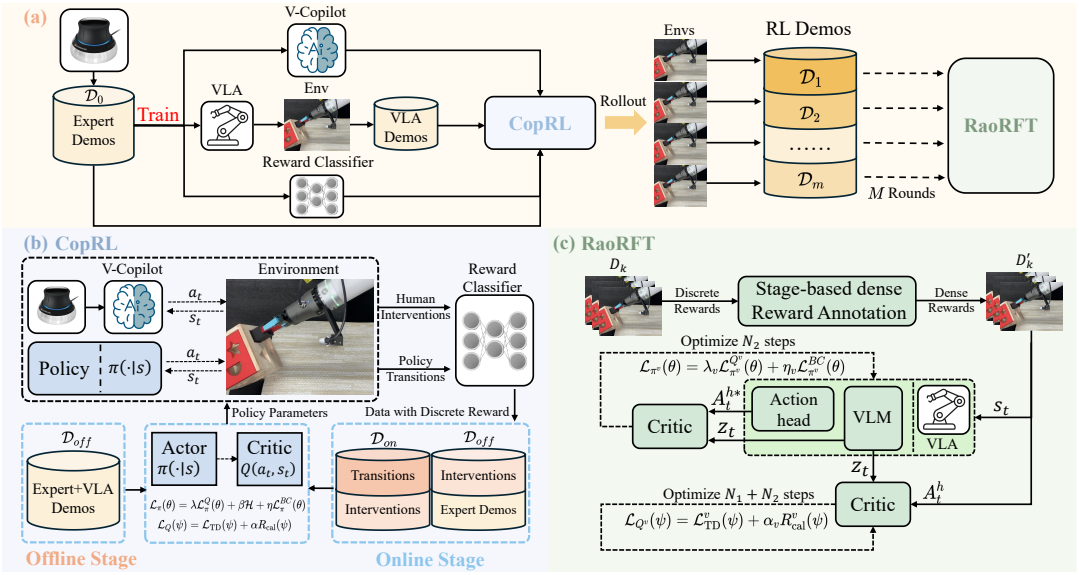


Fig. 3: (a) Overview of our pipeline, which includes (b) **CopRL**: a real-world RL method that employs human-in-the-loop guidance enhanced by the **Vision-guided Copilot** for more precise and efficient operation, and (c) **RaoRFT**: an offline RL post-training process for chunk-based VLA with stage-based dense reward annotations.

increasing the flexibility of the copilot while maintaining its capacity to discriminate between intentional expert maneuvers and exploratory actions, and increase the convergence speed for our real-world RL. The principle of the Vision-guided Copilot is illustrated in Fig 2.

We leverage the Diffusion Policy framework [24], incorporating both image and proprioceptive data as global conditioning in the U-Net to predict noise. The training loss is:

$$\mathcal{L}_{cop} = \text{MSE}\left(\varepsilon^k, \varepsilon_\varphi\left(s_t, \mathbf{A}_t^0 + \varepsilon^k, k\right)\right) \tag{1}$$

where ε^k is random noise for step k, s_t is the state (including image and proprioceptive data), and ε_φ is the noise prediction network.

Our sampling diverges from standard Diffusion Policy. Rather than initializing from Gaussian noise, we perturb human input $\mathbf{A}_{t,h}^0$ via forward diffusion for k_{sw} steps ($k_{sw} \leq K$), yielding $\mathbf{A}_{t,h}^{k_{sw}}$:

$$\mathbf{A}_{t,h}^{k_{sw}} = \mathbf{A}_{t,h}^0 + \varepsilon^{k_{sw}} \tag{2}$$

$$\mathbf{A}_{t,s}^{k_{sw}} = \mathbf{A}_{t,h}^{k_{sw}} \tag{3}$$

where $\mathbf{A}_{t,s}^{k_{sw}}$ is the noised shared action. Here, when $k_{sw} = K$, it corresponds to the standard forward diffusion process.

This noised action undergoes the denoising process through the U-Net ε_φ, progressively yielding the final shared action $\mathbf{A}_{t,s}^0$:

$$\mathbf{A}_{t,s}^{k_{sw}-1} = \alpha\left(\mathbf{A}_{t,s}^{k_{sw}} - \gamma\varepsilon_\varphi\left(s_t, \mathbf{A}_{t,s}^{k_{sw}}, k_{sw}\right) + \mathcal{N}\left(0, \sigma^2 I\right)\right) \tag{4}$$

where $\mathcal{N}\left(0, \sigma^2 I\right)$ is added Gaussian noise.

This design blends user intent with expert strategies, modulated by k_{sw}: where larger k_{sw} increases stochasticity, blurring user intent and converging toward expert distribution (leading to a large $||\mathbf{A}_{t,h}^0 - \mathbf{A}_{t,s}^0||_2^2$); smaller k_{sw} preserves user intent, diverging from expert distribution (leading to a small $||\mathbf{A}_{t,h}^0 - \mathbf{A}_{t,s}^0||_2^2$).

In practice, we use a small k_{sw} to better preserve user intent. This is because the U-Net—which keeps s_t clean by using it only as global conditioning rather than as part of the denoising output—is more sensitive to the high-dimensional clean state s_t with expert information than to the low-dimensional noised $\mathbf{A}_{t,h}^{k_{sw}}$ with user intent. As a result, the denoised shared action $\mathbf{A}_{t,s}^0$ skews toward the expert distribution, increasing $||\mathbf{A}_{t,h}^0 - \mathbf{A}_{t,s}^0||_2^2$. Using a small k_{sw} mitigates this bias and helps maintain intentionality.

During the CopRL online data collection—particularly for human interventions guided by the Vision-guided Copilot—we intentionally use only a wrist camera. This preserves user control when operating beyond its view, reducing state-induced overguidance and promoting exploratory motions, and when operating near and in view of targets, it provides detailed visual feedback to aid precision in delicate tasks. However, for training the final VLA via RaoRFT, we incorporate additional visual inputs, including a third-view camera and (in bimanual setups) an optional second wrist camera, to provide richer perceptual information and improve policy robustness.

B. CopRL: Real-world RL with Vision-guided Copilot

1) Preliminary: In order to generate high-quality training data for finetuning VLA models with few human operations, we introduce **CopRL**, a real-world RL method with Vision-guided Copilot, as a robust policy that is then distilled into the VLA. We formulate the real-world robotic manipulation task as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where $s_t \in \mathcal{S}$ denotes the state space (comprising image observations and proprioceptive state), $a_t \in \mathcal{A}$ is the action space of the robot, $\mathcal{P}(s'|s,a)$ is the state transition probability, $r(s,a)$ is the reward function, and $\gamma \in (0,1)$ is the discount factor. As illustrated in Fig.3(b) ,

the architecture of CopRL employs an actor-critic structure, where:

- The **actor** $\pi_\theta(a|s)$ outputs actions conditioned on states and is optimized to maximize the expected Q-value, enhanced by entropy regularization [25] and a behavior cloning loss, as formalized in Eq. (7).
- The **critic** $Q_\psi(s,a)$ estimates the expected return of state-action pairs, incorporating the Cal-QL method [26] in its loss to improve off-policy generalization, as defined in Eq. (6).

The reason for using the Cal-QL method is that it handles out-of-distribution (OOD) actions by penalizing overestimation, improving policy stability. Its calibration mechanism boosts value estimation in discrete-reward tasks, enhancing sample efficiency, while providing a stable Q-function initialization for online fine-tuning.

Building on this design choice, we implement the actor and critic networks with a shared frozen ResNet-18 vision encoder. The actor possesses a 4-layer MLP to output the mean and the standard deviation of the action distributions, and the critic utilizes a 3-layer MLP to predict the Q-value based on the observation features and the actions.

2) Offline Training with Expert and VLA Demonstrations: Performing online RL directly in the real world can be time-consuming. To address this, we first pretrain the policy using a purely offline dataset, which enhances data efficiency and reduces overall convergence time.

We begin by collecting a small expert dataset $\mathcal{D}_0$ consisting of 30 demonstrations with stage-specific discrete rewards. This dataset is used to: train the **Vision-guided Copilot** $f_\varphi(\cdot)$, perform **SFT on VLA models**, and train a **reward classifier** $r_\phi(\cdot)$ for online RL, following the architecture design from [13].

Next, the fine-tuned VLA interacts with the environment, generating 50 additional trajectories. At this point, the VLA's execution success rate remains relatively low, leading to the inclusion of a proportion of failure trajectories. These failure cases provide valuable training signals for the RL agent. Finally, expert and VLA demonstrations are merged into an offline replay buffer $\mathcal{D}_{off}$. Offline training is then conducted on this buffer to pretrain the actor and critic networks, aligning them with the target distribution for the following online training stage.

3) Online Training with Vision-guided Copilot: While the offline stage provides an initial policy from a limited set of demonstrations, its performance is constrained by the scale and quality of the pre-collected data. To overcome this limitation, we further refine the RL agent using a human-in-the-loop learning [13] framework in the online phase. A key feature of this approach is the human intervention mechanism: when the RL policy exhibits risky actions or substantial deviation from the target, a human operator can take over control.

These interventions produce human corrections, which are stored in two buffers: the offline demo buffer $\mathcal{D}_{off}$, which holds the original offline data, and the online replay buffer $\mathcal{D}_{on}$, which accumulates online policy transitions. During the online phase, the transition is sampled uniformly from $\mathcal{D}_{on}$ and $\mathcal{D}_{off}$ to form a training batch for network updating. By incorporating such human corrective signals, both buffers provide high-level guidance that enables safer and more efficient exploration.

While human interventions provide valuable corrective signals, performing teleoperation in practice is often non-trivial. In particular, operating high-DoF robotic systems with low-dimensional controllers (e.g., Spacemouse) in multi-DoF delicate tasks, such as inserting a cube from the side, is challenging. To mitigate this difficulty, we incorporate the **Vision-guided Copilot** to assist the teleoperation process.

Specifically, during human intervention, the input action $\mathbf{A}_{t,h}$ is first processed by the Vision-guided Copilot, which outputs a shared action $\mathbf{A}_{t,s}$ enriched with expert behavior.

$$\mathbf{A}_{t,s} = f_\varphi(\mathbf{A}_{t,h}, k_{sw}, s_t) \tag{5}$$

where f_φ denotes the Vision-guided Copilot, s_t represents the current observation comprising image data and the robot's proprioceptive state, and k_{sw} indicates the diffusion step at which the process switches from forward noise addition to reverse denoising.

During the online training phase, the reward for every state and action is obtained through the prediction from the pretrained reward classifier $r_\phi(s,a)$.

4) Loss Functions: Our loss function comprising two main components: one for updating the critic network and another for updating the actor network. The loss function remains consistent throughout both offline pretraining and online fine-tuning, enabling real-world online interactions to quickly adapt to the offline-pretrained policy. The specific forms of these loss functions are defined as follows:

$$\begin{aligned}\mathcal{L}_Q(\psi) = \mathbb{E}_{s,a,s'}\Big[\Big(Q_\psi(s,a) - \\ \Big(r_\phi(s,a) + \gamma E_{a'\sim\pi_\theta}\left[Q_{\bar{\psi}}(s',a')\right]\Big)\Big)^2\Big] \\ + \alpha\Big(E_{s\sim D, a\sim\pi_\theta}[\max(Q_\psi(s,a), V^\mu(s))] \\ - E_{s,a\sim D}[Q_\psi(s,a)]\Big)\end{aligned} \tag{6}$$

$$\begin{aligned}\mathcal{L}_\pi(\theta) = -\mathbb{E}_{s,a\sim\mathcal{D},a^*\sim\pi_\theta}\Big[\lambda Q_\psi(s,a^*) + \beta\mathcal{H}\left(\pi_\theta(\cdot|s)\right) \\ - \eta\|a^* - a\|_2\Big]\end{aligned} \tag{7}$$

where Q_ψ is the learned Q-function, $Q_{\bar{\psi}}$ is the delayed target Q-function, $V^\mu(s)$ is the value of the reference policy, $\mathcal{H}\left(\pi_\theta(\cdot|s)\right) = -\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\log\pi_\theta(a|s)\right]$, γ is the discount factor, s' and a' the next state and action, $\lambda = 1/avg(|Q_\psi(s,a^*)|)$ which results from [27], and α, β, η the weights for the designed losses.

C. RaoRFT: Reward-annotated Offline RL for Finetuning VLA

The CopRL framework provides a stream of high-quality, reward-annotated trajectories. The RaoRFT algorithm is designed to effectively distill this data into the VLA through offline RL. This process involves three critical steps: first,

transforming the raw data into a suitable training format; second, designing a phased training regimen to ensure stable learning; and finally, defining the optimization objectives that guide the policy improvement.

1) Dense Reward Annotation and Data Pipeline: The raw output from CopRL constitutes a continuous stream of interaction data rather than a static dataset. These data are organized into a sequence of buffers $\{\mathcal{D}_m\}_{m=1}^{M}$, wherein each $\mathcal{D}_m$ contains a batch of trajectories gathered during the m-th round of interaction between the CopRL policy and the environment. While $\mathcal{D}_m$ is employed for training in RaoRFT, CopRL simultaneously collects new trajectories into $\mathcal{D}_{m+1}$, thereby maintaining a continuous pipeline of data collection and policy refinement.

To provide richer learning signals for the VLA, we process the data in each buffer through a **Stage-Based Dense Reward Annotation scheme**. This method segments each trajectory into stages defined by discrete rewards. For each state within a stage, given the current end-effector position P_{ee} and orientation R_{ee}, a dense reward is computed based on the progression towards the next sub-goal [28](e.g., the target end-effector pose P_{ee}^{*}, R_{ee}^{*} at the end of each stages):

$$\mathbf{r} = \mathbf{r}_{ep} + \mathbf{r}_{ep} * \mathbf{r}_{eo} \tag{8}$$

$$\mathbf{r}_{ep} = exp(-d_p/\sigma) \tag{9}$$

$$\mathbf{r}_{eo} = exp(-d_\theta/\sigma) \tag{10}$$

where $d_\theta = ||log(R_{ee}^{*}R_{ee}^{T})||_F$ is the Frobenius norm of the logarithm of $SO(3)$, $d_p = ||P_{ee}^{*} - P_{ee}||_2$ is Euclidean distance, and σ the dense reward coefficient.

This transformed data is stored in a sequence of annotated buffers $\{\mathcal{D}'_m\}_{m=1}^{M}$, which form the direct input for the subsequent offline training stages.

2) Phased Training Strategy: With the densely annotated buffers prepared, we employ a two-phase training strategy to ensure stable and efficient policy distillation.

Phase 1 is dedicated to SFT for initial alignment. In the initial round, we construct a high-quality demonstration dataset by combining successful trajectories from buffer $\mathcal{D}'_1$ with human expert demonstrations from $\mathcal{D}_0$, totaling 200 trajectories. This dataset is used for SFT of the VLA, ensuring that the initial policy output aligns with expert behavior and the task dynamics, thereby providing a stable and sensible baseline from which RL can effectively proceed.

Phase 2 is dedicated to offline RL for policy enhancement. Subsequent rounds ($M-1$ in total) perform offline RL to further refine the VLA policy. We adopt an Actor-Critic architecture in which the VLA serves as the actor $\pi_\theta^v(\cdot|s_t)$, generating action chunks $A_t^h = (a_t, \ldots, a_{t+h})$. The critic $Q_\psi^v(z_t, A_t^h)$ is implemented as an MLP that estimates Q-values using the action chunk and a state representation z_t. Here, z_t represents the VLM/transformer output for Transformer-based VLA, or the input feature encoding otherwise. Each round of training consists of two steps: first, the critic is trained independently for N_1 steps to stabilize value estimates; then, both actor and critic are updated jointly for N_2 steps to refine the policy.

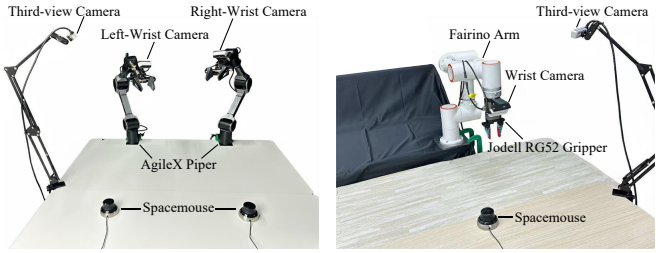


Fig. 4: The overall setup of our experiments for (a) bimanual scenarios and (b) single-arm.

3) Optimization Objectives: The loss function for the critic network retains the Cal-QL method. Considering the action-chunk-based VLA, we adopt the Q-Chunking [29] technique, adapting the TD error calculation accordingly. Specifically, the proposed loss function consists of two components: a TD loss term from Q-Chunking and a calibrated Q regularizer in the style of Cal-QL:

$$\mathcal{L}_{Q^v}(\psi) = \mathcal{L}_{\mathrm{TD}}^{v}(\psi) + \alpha_v R_{\mathrm{cal}}^{v}(\psi) \tag{11}$$

where $\mathcal{L}_{\mathrm{TD}}^{v}$ and $R_{\mathrm{cal}}^{v}(\psi)$ is given by:

$$\mathcal{L}_{TD}^{v}(\psi) = \mathbb{E}_{s_t,\, A_t^h,\, s_{t+h}\sim\mathcal{D}'_m}\left[\left(Q_\psi^v\left(z_t, A_t^h\right) - \sum_{t'=1}^{h}\gamma^{t'} r_{t+t'} - \gamma^h Q_{\bar{\psi}}^v\left(z_{t+h}, A_{t+h}^h\right)\right)^2\right] \tag{12}$$

$$R_{\mathrm{cal}}^{v}(\psi) = \mathbb{E}_{s_t\sim\mathcal{D}'_m,\, A_t^{h*}\sim\pi_\theta^v}\left[\max\left(Q_\psi^v(z_t, A_t^{h*}),\, V^\mu(s_t)\right)\right] - \mathbb{E}_{s_t,\, A_t^h\sim\mathcal{D}'_m}\left[Q_\psi^v(z_t, A_t^h)\right] \tag{13}$$

where $V^\mu(s_t)$ represents the value function of the reference policy and α_v is the regularization strength coefficient.

The loss function for the VLA (which serves as the actor network) is optimized to maximize the Q-value and with a behavior cloning loss for convergence stabilization.

$$\mathcal{L}_{\pi^v}(\theta) = -\mathbb{E}_{s, A_t^h\sim\mathcal{D}'_m, A_t^{h*}\sim\pi_\theta^v}\left[\lambda_v Q_\psi^v(z_t, A_t^h) - \eta_v \left\|A_t^{h*} - A_t^h\right\|^2\right] \tag{14}$$

with $\lambda_v = 1/avg(|Q_\psi^v(z_t, A_t)|)$ inspired by [27], and η_v the coefficient for the behavior cloning loss.

IV. EXPERIMENTS

A. Overview of Experiments

We conducted experiments across five distinct tasks, encompassing a diverse set of characteristics as illustrated in Fig. 1. These tasks cover a range of manipulation challenges, including high-precision tasks with constrained pitch and yaw axes (e.g., insert cube from top), multi-DoF delicate and subtle manipulation (e.g., insert cube from side), dexterous object operation (e.g., install the belt), high-tolerance grasping (e.g., pull the drawer), and bimanual collaboration (e.g., object handover).

We now detail the specifications of each task:

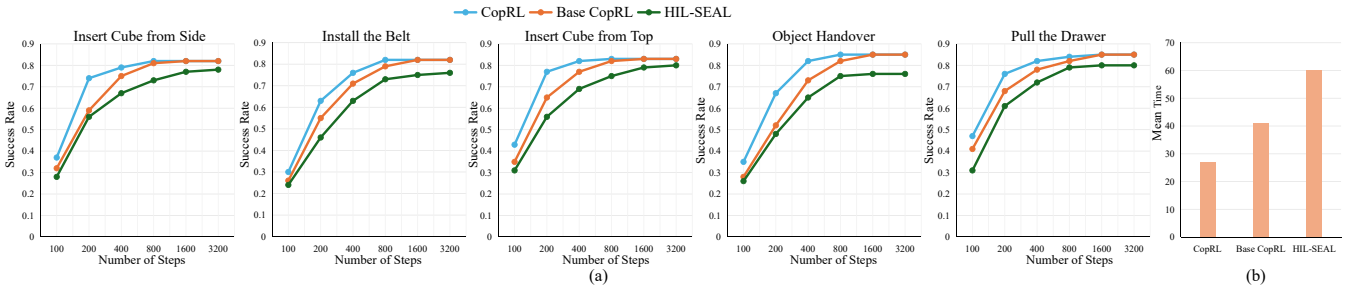
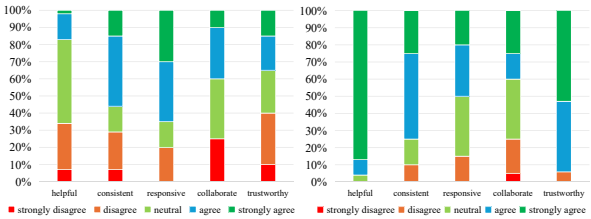


Fig. 5: (a) The success rate of 5 tasks. (b) Mean convergence time of CopRL for the 5 tasks.



(a) Base-CopRL (w/o Vision-guided Copilot) (b) CopRL (w/ Vision-guided Copilot)

Fig. 6: Results of the human user qualitative scores (a) with Vision-guided Copilot or (b) without the Vision-guided Copilot.

- **Insert Cube from Top**: Grasp a cube and insert it vertically into a container. Key challenge: precise position control for accurate insertion.
- **Insert Cube from Side**: Insert the cube into a tilted container. Requires continuous pose adjustment and is more challenging for low-DoF teleoperation.
- **Install the Belt**: Mount a synchronous belt onto fixed pulleys. Must adapt to unpredictable belt deformation during assembly.
- **Pull the Drawer**: Open a drawer smoothly using compliant motion and force control, avoiding jams.
- **Object Handover**: Transfer an object between two arms. Primary difficulty is bimanual coordination.

For all tasks, we collected several RGB images with one from a third-view camera and one or two images from wrist-mounted cameras according to the number of arms. The proprioceptive state information and rewards (e.g., successful insertion, grasping, or task completion) are also collected. The overall setup is demonstrated in Fig. 4. For single-arm tasks, we use a Fairino Arm paired with a Jodell RG52 Gripper, operating at a control frequency of 10 FPS. In the bi-manual scenario, two AgileX Piper arms are used with a control frequency of 30 FPS.

For the training settings, we set the denoise step for Vision-guided Copilot $K = 100$, the coefficient for the Q regularizer $\alpha = \alpha_v = 1.0$, the entropy loss weight $\beta = 0.01$, the weights of behavior cloning losses $\eta = \eta_v = 0.5$, the reward discount factor $\gamma = 0.99$, the dense reward coefficient $\sigma = 0.25$. As the offline RL training for VLA, we set the pretraining steps for the critic network $N_1 = 2000$, and the ensemble update steps $N_2 = 18000$. The number of training rounds is set to $M = 10$ for total convergence of the VLA models.

B. Experiment Results on CopRL

To evaluate the performances of our proposed CopRL and other SOTA baselines, we assess various performance metrics including **Convergence Speed**, **Success Rate**, **Human Intervention Time** and **User ratings**.

In the comparison of RL agent convergence speed, we primarily evaluate **CopRL (with Vision-guided Copilot)** against **Base-CopRL (without Vision-guided Copilot)**, as well as **HIL-SERL** [13]. The convergence progress is measured by tracking the success rate at specific checkpoints during training. Specifically, the success rate refers to the ratio of successful trials achieved over 50 repeated executions of a designated task using the model saved at each evaluation point.

Experimental results of five tasks, as shown in Fig. 5(a), show that CopRL achieves faster convergence than Base-CopRL, confirming that the Vision-guided Copilot accelerates RL training by providing targeted human guidance. This directs the agent more efficiently toward reward-rich regions and delivers more effective supervision. Additionally, Base-CopRL converges faster than HIL-SERL, indicating that our offline RL pretraining and designed losses contributes to accelerated convergence.

Having shown that CopRL accelerates learning, we next examine its ability to reduce human effort and enable autonomous collection of large-scale reward-labeled data.

We evaluate the total human intervention time required to reach a 80% success rate. As shown in Fig. 5(b), CopRL significantly reduces human intervention compared to HIL-SERL, indicating higher autonomy. This improvement stems from two main factors: the Vision-guided Copilot facilitates high-precision teleoperation in challenging tasks and provides critical guidance that promotes sample-efficient and stable policy improvement; meanwhile, offline pretraining, which usese calibrated Q regularizers for robust out-of-distribution predictions and behavior cloning for imitating expert demonstrations, effectively reduces rollout time during online fine-tuning.

At last, we conducted a subjective user study (the number of participants is 20) comparing the CopRL (**with Vision-guided Copilot**) and the Base CopRL (**without Vision-guided Copilot**). Each participant performed all tasks using both algorithms in a blinded setup, with 5 practice episodes per method. During evaluation, 10 trials per task were conducted for each algorithm. After the trials, participants provided feedback on each system, rating their agreement on a 5-point Likert scale across five attributes: “helpful”, “consistent”, “responsive”, “collaborative”, and “trustworthy”. Results (Fig. 6) show significantly higher ratings for

TABLE I: Performance comparison of SFT vs RaoRFT across different VLA models

Policy	fine-tune	Insert Cube(Top)		Insert Cube(Side)		Install the Belt		Pull the Drawer		Object Handover	
		SR↑	CT(s)↓	SR↑	CT(s)↓	SR↑	CT(s)↓	SR↑	CT(s)↓	SR↑	CT(s)↓
ACT	SFT	0.79	62	0.78	68	0.81	56	0.83	54	0.75	42
	RaoRFT	**0.88**	**54**	**0.85**	**51**	**0.87**	**52**	**0.89**	**46**	**0.83**	**33**
OpenVLA-OFT	SFT	0.83	58	0.82	55	0.83	53	0.89	45	0.81	34
	RaoRFT	**0.91**	**50**	**0.89**	**45**	**0.91**	**46**	**0.95**	**37**	**0.88**	**28**
π_0	SFT	0.86	53	0.84	48	0.82	44	0.88	38	0.86	25
	RaoRFT	**0.93**	**46**	**0.92**	**39**	**0.89**	**41**	**0.94**	**33**	**0.92**	**22**

CopRL across all dimensions.

C. Experimental Results on RaoRFT

We compare the performance of offline RL (**RaoRFT**) and SFT on VLA models, including OpenVLA-OFT and π_0. We also include ACT [30] as a reference. Performance is measured via **Success Rate**: the proportion of successful executions in 50 trials per task, and **Cycle Time**: the average time to complete a task successfully.

In terms of dataset configuration, we constructed the following two types for training and evaluation: $\mathcal{D}_a$, a complete robot interaction dataset collected via our CopRL framework containing both high-quality and suboptimal trajectories; and $\mathcal{D}_b$, a curated expert subset filtered from $\mathcal{D}_a$ with higher quality. Furthermore, we forgo the use of human-collected data because our human-in-the-loop RL framework—as validated by RLDG [5]—yields higher-quality demonstrations than those acquired through manual teleoperation.

SFT relies on high-quality demonstrations and is trained solely on $\mathcal{D}_b$, while our offline RL method RaoRFT uses explicit reward signals to learn from mixed-quality data and is trained exclusively on $\mathcal{D}_a$. This setup reflects the advantage of RaoRFT in extracting useful policies from diverse, uncurated data. It also aligns with practical constraints where pure expert data is scarce and costly, demonstrating the robustness and scalability of our approach. The comparison between RaoRFT on $\mathcal{D}_a$ and SFT on $\mathcal{D}_b$ is designed to highlight the efficiency of our method in leveraging raw interaction data without expert-level filtering.

As shown in Tab. I, RaoRFT consistently outperforms SFT across all tasks. This indicates that RaoRFT not only enhances base model performance but also improves data utilization by effectively learning from suboptimal trajectories.

D. Ablation Study

1) The Effectiveness of First Round SFT in RaoRFT: In this ablation study, we focus on comparing two strategies in the post-training phase of the VLA: (1) **SFT+RL**: a hybrid approach using SFT in the first round followed by RL in subsequent rounds, and (2) **RL-only**: a pure RL approach applied throughout all rounds. We chose the **Insert Cube from Side** task for our ablation study due to its high demand for precise control, making it a challenging benchmark for policy robustness. As shown in Fig. 7, quantitative results demonstrate that the SFT+RL strategy significantly outperforms the RL-only approach in both convergence speed and final task success rate. Specifically, after one SFT round, the SFT+RL model quickly aligns with the target data, achieving over 70% success within four rounds, while RL-only stagnates at 30% success.

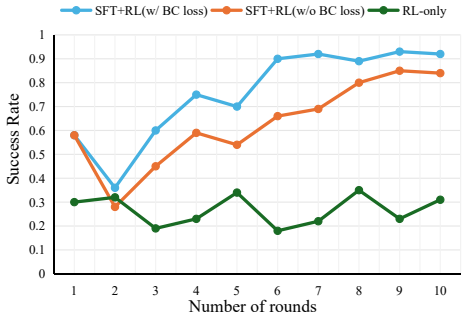


Fig. 7: Success rate for RaoRFT with or without the behavior cloning loss, along with the pure RL training paradigm.

This performance discrepancy stems from the distinct learning mechanisms of the two methods. The SFT constrains the VLA's output to stay close to the demonstration data, ensuring basic action rationality and safety, which provides a stable foundation for RL optimization. In contrast, RL-only suffers from high randomness and exploration inefficiency, especially in high-dimensional spaces, often leading to local optima or irrelevant actions. In conclusion, the SFT phase acts as a crucial initialization, reducing RL exploration difficulty and cost, demonstrating the effectiveness of our phased training strategy.

2) Benefits of Behavior Cloning Loss: In this ablation study, we evaluate the critical role of the BC loss in both CopRL and RaoRFT, using the Insert Cube from Side task as well.

The first part focuses on CopRL training, comparing the effect of incorporating the BC loss against pure RL training. As shown in Fig. 8, experimental results indicate that While both methods achieve similar success rates (75%-85%), the BC loss approach converges significantly faster, requiring 30% fewer training steps. This is due to the BC loss guiding exploration by enforcing imitation of demonstration actions, which improves sample efficiency and safety by preventing inefficient or risky random exploration in early stages.

The second part focuses on the RaoRFT, proving the effects of introducing BC loss into the RaoRFT algorithm. As shown in Fig. 7, results show a 10% improvement in success rate and earlier convergence with BC loss. This is because the BC loss acts as a regularizer to prevent policy collapse and an implicit supervisor that guides the

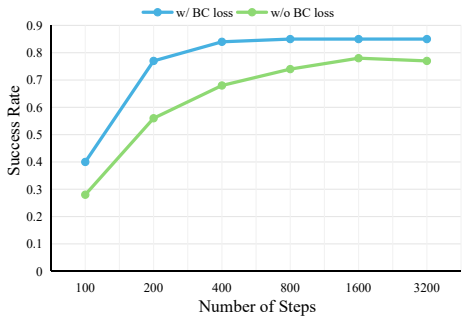


Fig. 8: Success rate for CopRL with or without the behavior cloning loss.

agent to prioritize successful behaviors. Although failure data may reinforce suboptimal actions, additional regularization through the value function (Q) helps mitigate this, directing the learning process toward optimal behaviors. The combination of imitation and optimization leads to more stable and efficient learning, reducing training time and improving overall performance.

V. CONCLUSION

In this work, we present a unified framework that effectively combines Vision-guided Copilot assistance, sample-efficient real-world RL (CopRL), and offline policy distillation (RaoRFT) to address the performance degradation of VLA when adapting to new tasks. Our approach substantially reduces the reliance on extensive human supervision while achieving superior task performance through autonomous, high-quality data generation. Extensive real-world experiments validate that our method enables robust and precise manipulation with minimal human intervention, outperforming existing baselines in both success rate and efficiency. Future work will explore extending this framework to more diverse task domains and investigating its scalability to larger model architectures.

REFERENCES

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[2] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.

[3] 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.

[4] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, *et al.*, "π0. 5: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[5] C. Xu, Q. Li, J. Luo, and S. Levine, "Rldg: Robotic generalist policy distillation via reinforcement learning," *arXiv preprint arXiv:2412.09858*, 2024.

[6] Y. Guo, J. Zhang, X. Chen, X. Ji, Y.-J. Wang, Y. Hu, and J. Chen, "Improving vision-language-action model with online reinforcement learning," *arXiv preprint arXiv:2501.16664*, 2025.

[7] Z. Zhang, K. Zheng, Z. Chen, J. Jang, Y. Li, S. Han, C. Wang, M. Ding, D. Fox, and H. Yao, "Grape: Generalizing robot policy via preference alignment," *arXiv preprint arXiv:2411.19309*, 2024.

[8] G. Lu, W. Guo, C. Zhang, Y. Zhou, H. Jiang, Z. Gao, Y. Tang, and Z. Wang, "Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning," *arXiv preprint arXiv:2505.18719*, 2025.

[9] Z. Chen, R. Niu, H. Kong, and Q. Wang, "Tgrpo: Fine-tuning vision-language-action model via trajectory-wise group relative policy optimization," *arXiv preprint arXiv:2506.08440*, 2025.

[10] S. Tan, K. Dou, Y. Zhao, and P. Krähenbühl, "Interactive post-training for vision-language-action models," *arXiv preprint arXiv:2505.17016*, 2025.

[11] Y. Chen, S. Tian, S. Liu, Y. Zhou, H. Li, and D. Zhao, "Conrft: A reinforced fine-tuning method for vla models via consistency policy," *arXiv preprint arXiv:2502.05450*, 2025.

[12] D. Huang, Z. Fang, T. Zhang, Y. Li, L. Zhao, and C. Xia, "Co-rft: Efficient fine-tuning of vision-language-action models through chunked offline reinforcement learning," *arXiv preprint arXiv:2508.02219*, 2025.

[13] J. Luo, C. Xu, J. Wu, and S. Levine, "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning," *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025.

[14] J. Luo, P. Dong, Y. Zhai, Y. Ma, and S. Levine, "Rlif: Interactive imitation learning as reinforcement learning," *arXiv preprint arXiv:2311.12996*, 2023.

[15] T. Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, and S. Levine, "Offline meta-reinforcement learning for industrial insertion," in *2022 international conference on robotics and automation (ICRA)*. IEEE, 2022, pp. 6386–6393.

[16] Z. Hu, A. Rovinsky, J. Luo, V. Kumar, A. Gupta, and S. Levine, "Reboot: Reuse data for bootstrapping efficient real-world dexterous manipulation," *arXiv preprint arXiv:2309.03322*, 2023.

[17] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, S. Schaal, C. Finn, A. Gupta, and S. Levine, "Serl: A software suite for sample-efficient robotic reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 961–16 969.

[18] D. J. Hejna III and D. Sadigh, "Few-shot preference learning for human-in-the-loop rl," in *Conference on Robot Learning*. PMLR, 2023, pp. 2014–2025.

[19] B. Luo, Z. Wu, F. Zhou, and B.-C. Wang, "Human-in-the-loop reinforcement learning in continuous-action space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 15 735–15 744, 2023.

[20] T. Yoneda, L. Sun, B. Stadie, M. Walter, *et al.*, "To the noise and back: Diffusion for shared autonomy," *arXiv preprint arXiv:2302.12244*, 2023.

[21] N. Amirshirzad, A. Kumru, and E. Oztop, "Human adaptation to human–robot shared control," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 2, pp. 126–136, 2019.

[22] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, "Shared control–based bimanual robot manipulation," *Science Robotics*, vol. 4, no. 30, p. eaaw0955, 2019.

[23] H. Zhang, S. Zhang, J. Jin, Q. Zeng, Y. Qiao, H. Lu, and D. Wang, "Balancing signal and variance: Adaptive offline rl post-training for vla flow models," *arXiv preprint arXiv:2509.04063*, 2025.

[24] 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*, p. 02783649241273668, 2023.

[25] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[26] M. Nakamoto, S. Zhai, A. Singh, M. Sobol Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine, "Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 244–62 269, 2023.

[27] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20 132–20 145, 2021.

[28] K. Jiang, Z. Fu, J. Guo, W. Zhang, and H. Chen, "Learning whole-body loco-manipulation for omni-directional task space pose tracking with a wheeled-quadrupedal-manipulator," *IEEE Robotics and Automation Letters*, 2024.

[29] Q. Li, Z. Zhou, and S. Levine, "Reinforcement learning with action chunking," *arXiv preprint arXiv:2507.07969*, 2025.

[30] 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.