Semantics-to-Contact: A Stagewise Framework for Robust Contact-Rich Manipulation

Guanren Qiao[1], **Ruixiang Ouyang**[1], **Sheng Xu**[1], **Yueci Deng**[1,2], **Yunxin Tai**[2], **Kui Jia**[1,2], **Guiliang Liu**[1,3] *
[1]The Chinese University of Hong Kong, Shenzhen,
[2]DexForce Co., Ltd, [3]Shenzhen Loop Area Institute.
liuguiliang@cuhk.edu.cn, guanrenqiao1@link.cuhk.edu.cn

Abstract

Contact-rich manipulation requires precise control under complex contact dynamics while remaining robust to diverse visual conditions. However, real-world reinforcement learning often overfits to the visual appearance of training scenes. We propose Semantics-to-Contact (S2C), a stagewise framework for visually robust contact-rich manipulation. The insight is to decompose the problem into two stages: semantic focusing and contact refinement. In the first stage, a vision-language model localizes task-relevant regions across visually diverse scenes, providing coarse spatial guidance and reducing the burden of exploration. In the second stage, a real-world human-in-the-loop residual reinforcement learning policy learns fine-grained contact behaviors within the localized region using force feedback. To further improve visual robustness, we introduce an object-centric visual augmentation strategy that randomizes background appearance while preserving the robot and task-relevant objects. Experiments on six contact-rich real-world manipulation tasks exhibit that S2C consistently improves success rates over strong baselines and maintains reliable performance under significant visual and positional variations. These results demonstrate that stagewise semantic focusing and contact refinement provide a practical path toward visually robust real-world contact-rich manipulation. Our demos can be seen in https://edem-ai.github.io/S2C/.

1 Introduction

Contact-rich manipulation is a key capability for enabling robots to operate in real-world applications [25], such as insertion [27], alignment [54, 48], and pressing [24, 54], which often require robots to perform fine-grained adjustments under multimodal force signals and frequently changing contact states. In recent years, reinforcement learning (RL) has demonstrated strong adaptability in such tasks [26, 17, 32, 29], particularly learning contact policies through human-in-the-loop (HIL) interaction with the real-world environment [24, 25]. However, real-world reinforcement learning (RWRL) still faces a fundamental challenge: *visual observations are highly susceptible to variations in background textures and lighting conditions, making policies learned in one scenario difficult to generalize to new real-world environments*.

Existing methods often rely on domain randomization in simulation to improve visual robustness. By randomly varying background textures, lighting conditions, and environmental parameters, policies can learn representations that are more robust to visual perturbations across a large number of virtual environments [47, 50, 37]. However, constructing a high-fidelity simulation environment for contact-rich manipulation is challenging [53]. Contact-rich manipulation tasks often exhibit a large sim-to-real gap [16]. For example, in industrial assembly, tight insertion tasks may involve clearances

*Correspondence to: Guiliang Liu <liuguiliang@cuhk.edu.cn>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

smaller than 0.1 mm, leading to complex contact dynamics that are difficult to accurately model in simulation. Therefore, a natural question arises: *Can we learn visually robust contact policies directly on real robots without relying on complex simulation environments?*

To address the above challenges, we propose **Semantics-to-Contact (S2C)**, a stagewise framework for visually robust contact-rich manipulation. We argue that a robot is a complex system in which perception, planning, and control are tightly coupled. Therefore, **visual generalization cannot be achieved by optimizing a single module alone; instead, it requires a stagewise, system-level design** [40]. The key idea is to decompose the task into two complementary stages: *semantic focusing* and *contact refinement*. In the first stage, a vision-language model (VLM) uses RGB-D observations and language instructions to identify task-relevant regions and guide the robot toward a semantic neighborhood. This stage provides coarse spatial grounding [23] across diverse scenes and prevents the RL policy from blindly exploring a large spatial range. In the second stage, human-in-the-loop real-world residual RL learns fine-grained contact behaviors within the localized region using proprioceptive and force feedback. This decomposition is particularly suitable for contact-rich manipulation: once the robot has been guided close to the target, it typically only needs to perform subtle pose adjustments to finish tasks successfully according to local force feedback.

To further enhance visual robustness, we introduce an *Object-centric visual augmentation strategy* during real-world training. Specifically, we extract task-relevant foreground regions from wrist-camera images and apply random perturbations to the background [5, 37], including changes in color and texture. The augmented images are then directly fed into the RWRL policy for training. In this way, the policy is encouraged to focus on manipulable objects and contact-relevant regions, rather than overfitting to the visual background of a specific experimental setup. At the control level, we adopt *residual RL* [27, 17, 2, 1] to improve the efficiency of real-world policy learning. Instead of learning a complete action policy from scratch, residual RL learns local corrections on top of an existing policy, thereby reducing the exploration space and improving learning efficiency.

We evaluate our method on six real-world contact-rich manipulation tasks and test its robustness under diverse visual and spatial variations. Experimental results show that, compared with strong baselines, S2C maintains a high success rate under both positional shifts and visual disturbances, while other methods almost fail to complete the task under visual disturbances. In summary, our contributions are threefold:

- We propose **Semantics-to-Contact (S2C)**, a stagewise real-world reinforcement learning framework that decomposes visually robust contact-rich manipulation into semantic focusing and contact refinement.
- We introduce a simulation-free visual robustness strategy that combines VLM-guided semantic localization with an object-centric visual augmentation strategy, enabling RWRL policies to better generalize across background textures and lighting conditions.
- We conduct many experiments and provide a number of videos to identify the robustness and scalability boundaries of S2C. The results illustrate that S2C achieves strong performance under most robust settings.

2 Related Work

2.1 Reinforcement Learning for Visual Generalization

Visual reinforcement learning often suffers the challenge of generalization to new environments [9, 20, 56, 50, 38, 41]. Practically, the previous works can be categorized into two groups: visual processing and RL structure designing. For visual processing, on one hand, [52, 10, 51] leveraged visual distractions to learn a robust policy from images. For example, VAI [39] extracts a universal visual foreground mask to provide an invariant observation to RL. On the other hand, [49, 20, 28] focused on image augmentation to improve the generalization performance of the policy. Specifically, FGA3 [21] leveraged adaptive adversarial augmentation while maintained semantic consistency to improve robustness of the agent. For RL structure designing, PAD [12] incorporated self-supervision to allow policies continue studying in new environments after deploy. DrQ [49] and SVEA [13] leveraged augmented Q and augmented target Q to provide policy generalization performance. However, many of the previous works mainly focused on simulation environments, while our work improves robustness performance in real-world scenarios.

2.2 Human-in-the-Loop Real-World Reinforcement Learning

Recent works successfully integrated human intervention into policy training [19, 29], and leveraged RL methods to optimize policy performance, enabling precise and dexterous robot manipulation, such as SERL and HIL-SERL [24, 25]. Leveraging the advantages of human guidance, recent work has greatly improved the robot's performance in dexterous manipulation tasks. HitL integrated human assistance by evaluating the uncertainty of diffusion policies [14], while Rac directly learned policy recovery from human corrections for long-horizon tasks [15]. Based on a human-in-the-loop RL system, recent works also made progress in fine-tuning VLA models [46, 22, 18]. ConRFT combined behavior cloning and Q-learning to achieve high sample efficiency [6], while HAPO turned to align the action reference using human interventions, helping VLA models avoid unsatisfactory motions [42]. However, these works require RL to learn the entire task process, without decoupling the manipulation procedure of each task [44, 43], and they do not explicitly consider visual robustness.

3 Problem Formulation

Goal-Conditioned Partially Observable Markov Decision Process. We formulate real-world contact-rich manipulation as an infinite-horizon goal-conditioned partially observable Markov decision process (GC-POMDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mu_0, \gamma, \mathcal{G})$ [32]. Here, $\mathcal{S}$ denotes the latent state space of the full robot-environment system, encompassing the states of the robot, objects, and visual background. Since such information is not fully observable in real-world contact-rich manipulation, the agent receives an observation $o_t \in \mathcal{O}$ at each timestep. Each observation consists of proprioceptive and visual components, i.e., $o_t = (o_t^p, o_t^v)$. The proprioceptive observation o_t^p includes the 6D end-effector pose, linear and angular velocities, and the 6D force-torque measurement. The visual observation o_t^v consists of RGB images captured by wrist-mounted cameras near the end-effector.

The action space $\mathcal{A}$ contains continuous end-effector motion commands and a discrete gripper command. Specifically, each action $a_t \in \mathcal{A}$ consists of a 6D end-effector pose delta and a binary gripper command. The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ describes the real-world robot-environment dynamics, and $\mu_0 \in \Delta(\mathcal{S})$ denotes the initial state distribution. The discount factor is denoted by $\gamma \in (0, 1)$.

Each task is associated with a goal $g \in \mathcal{G}$, specified by a target contact region. The reward function $r_t = \mathcal{R}(o_t, a_t, g)$ is a sparse task completion reward. The objective is to learn a goal-conditioned policy $\pi_\theta(a_t \mid o_t, g)$ that maximizes the expected discounted return $J(\pi_\theta) = \mathbb{E}_{\pi_\theta, \mathcal{T}} \left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ .The state-action value function under policy π_θ is defined as

$$Q^\pi(o_t, a_t, g) = \mathbb{E}_{\pi, \mathcal{T}} \left[\sum_{j=t}^{\infty} \gamma^{j-t} r_j \mid o_t, a_t, g \right], \tag{1}$$

Real-World Reinforcement Learning with Human Intervention. Our real-world RL policy is optimized using Soft Actor-Critic (SAC) [11]. The maximum-entropy objective is

$$\pi^* = \arg\max_\pi \mathbb{E}_{\pi, \mathcal{T}} \left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha \mathcal{H}(\pi(\cdot|o_t, g)) \right) \right], \tag{2}$$

where α controls the strength of the regularization of the entropy.

Following human-in-the-loop RWRL [25], a human operator can intervene during training when the robot enters unsafe or unproductive states. We denote the intervention indicator as $l_t \in \{0, 1\}$, where $l_t = 1$ indicates human control and $l_t = 0$ indicates autonomous policy control. The executed action is selected as

$$a_t = \begin{cases} a_t^{\mathrm{RL}}, & l_t = 0, \quad a_t^{\mathrm{RL}} \sim \pi_\theta(\cdot|o_t, g), \\ a_t^{\mathrm{H}}, & l_t = 1, \quad a_t^{\mathrm{H}} \sim \pi_{\mathrm{H}}(\cdot|o_t, g). \end{cases} \tag{3}$$

4 Semantics-to-Contact Stagewise Framework

We present **Semantics-to-Contact (S2C)**, a stagewise framework for visually generalizable contact-rich manipulation. Figure 1 introduces the pipeline of S2C. First, the robot must understand *where*

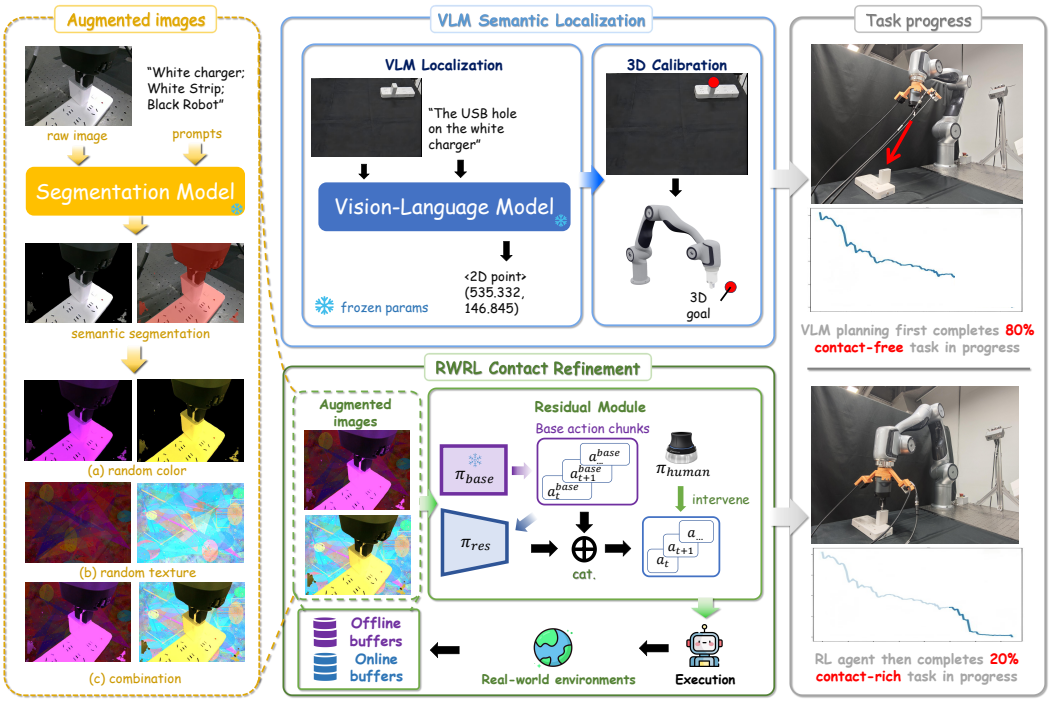


Figure 1: **S2C Framework**: S2C decomposes contact-rich manipulation into two stages: *semantic localization* and *contact refinement*. The semantic focusing stage uses the VLM to predict a task-relevant pre-contact goal $\hat{g}$ and motion planning to move the robot near the desired contact region. The contact refinement stage then uses human-in-the-loop real-world residual RL to learn local force-aware corrections.

the task-relevant contact should happen in a visually diverse scene. Second, once it reaches the vicinity of the target, it must determine *how* to make contact under uncertain force interactions. These two problems have different structures: the former requires open-world semantic understanding and spatial grounding, while the latter requires fine-grained closed-loop control with force feedback. S2C explicitly separates these two challenges into two stages: the first stage performs *semantic localization* to identify task-relevant contact regions, while the second stage performs *contact refinement* through real-world residual reinforcement learning to learn fine-grained contact behaviors.

4.1 Semantic Localization: VLM-Guided Active Perception

The first stage addresses a core difficulty in RWRL: the agent does not know *where* to explore in the visual space. Without prior semantic information, the RL policy would need to wander randomly over large 3D workspace areas, which is sample-inefficient and often unsafe on a real robot. We leverage a VLM to close this semantic gap.

VLM Guidance. At the beginning of each episode, the robot receives a language instruction l (e.g., "Insert into the USB port") and a third-person view RGB-D observation (P_t^{side}, D_t), where P_t^{side} is the third-view RGB image and D_t is the corresponding depth map. The VLM processes P_t together with l and outputs a prediction point. In practice, we utilize an SOTA VLM MolmoPoint 8B [8] with a carefully designed few-shot prompt that asks for the 2D point relevant to l. This model is trained to point by directly selecting which visual token contains the target object and then refining that location by generating additional tokens. Compared to the open-source GPT-5 [31, 45] or Gemini-Robotics-ER-1.5 [33] API, MolmoPoint 8B has stronger image understanding capabilities, enabling it to infer the locations of affordances in an image, while also offering much faster inference speed (2s) than the GPT-5 API ($\geq 10s$). In scenarios similar to USB insertion, it can understand where the insertion hole is located in the image, thus meeting our requirements. Using the depth map D_t, we compute the 3D position of the interactive object in camera coordinates:

$$\mathbf{c}_l^t = D_t(u_l^t, v_l^t)\mathbf{K}^{-1}[u, v, 1]^\top, \quad \mathbf{c}_l^t \in \mathbb{R}^3. \tag{4}$$

where K is the intrinsic matrix of the camera and u, v represent the positions of the points marked by the VLM in the image. We obtain the hand-eye calibration matrix $T_{\text{cam}}^{\text{base}} \in SE(3)$, which denotes the homogeneous transformation matrix from the camera coordinate to the robot base coordinate through a 6-well plate. We fix the position of the camera from a third-person perspective and perform calibration using the eye-to-hand method. The target contact point in the robot base frame is $T_{\text{cam}}^{\text{base}} \cdot c_l$. The robot then executes a motion-planning primitive to reach $p_{\text{target}}^{\text{base}}$. Formally, we define the semantic focus operator:

$$\Phi_{\text{VLM}}(P_t^{side}, l) = p_{\text{target}}^{\text{base}} = T_{\text{cam}}^{\text{base}} \cdot c_l \in SE(3). \tag{5}$$

This operator reduces the effective state space for the subsequent RL from the entire workspace to a small local volume.

Object-centric Visual Augmentation. To improve visual robustness without relying on simulation, we introduce an object-centric visual augmentation module [36] during RWRL training. We choose YOLOE as the segment model because it is lightweight enough for online preprocessing and supports promptable object detection and segmentation. This allows us to specify task-relevant foreground regions using language prompts or visual prompts, without training a task-specific segmentation model from scratch. We choose prompt guidance over visual because the wrist-mounted perspective moves along with the robotic arm, so the reference images provided would not offer effective guidance. Given two wrist-camera images $\{P_{t,k}^{\text{wrist}}\}_{k=1}^{2}$ and a set of prompts $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$ describing the robot arm, gripper, and manipulated objects, YOLOE predicts a foreground mask for each view:

$$M_{t,k} = f_{\text{YOLOE}}\left(P_{t,k}^{\text{wrist}}, \mathcal{L}\right), \quad k \in \{1, 2\}, \tag{6}$$

where $M_{t,k} \in \{0,1\}^{H \times W}$ is a binary foreground mask. Pixels with $M_{t,k} = 1$ correspond to regions that should be preserved, including the robot arm, gripper, manipulated object, and contact-relevant structures, while pixels with $M_{t,k} = 0$ correspond to background regions.

For each wrist-camera image, we sample a random background texture $B_{t,k}$ from an augmentation distribution $q(B)$ and construct a background-randomized image:

$$\bar{P}_{t,k}^{\text{wrist}} = M_{t,k} \odot P_{t,k}^{\text{wrist}} + (1 - M_{t,k}) \odot B_{t,k}, \tag{7}$$

where $\odot$ denotes element-wise multiplication. The augmentation distribution includes random colors, textures, and visual distractors. This operation randomizes the appearance of task-irrelevant background while preserving the geometric and semantic structure of the foreground.

In addition to background randomization, we also perturb the segmented foreground to prevent the policy from overfitting to specific object colors or lighting conditions. Let $\mathcal{A}_{\text{fg}}(\cdot)$ denote a foreground augmentation operator that applies random color jitter and brightness changes [5]. The final augmented wrist-camera image is:

$$\tilde{P}_{t,k}^{\text{wrist}} = M_{t,k} \odot \mathcal{A}_{\text{fg}}\left(P_{t,k}^{\text{wrist}}\right) + (1 - M_{t,k}) \odot B_{t,k}. \tag{8}$$

The resulting augmented observation is: $\tilde{o}_t = \left(o_t^p, \tilde{P}_{t,1}^{\text{wrist}}, \tilde{P}_{t,2}^{\text{wrist}}\right)$. Appendix C supplements other augmentation details. Unlike simulation-based domain randomization, it does not require modeling and 3D assets. The physical interaction remains real, while task-irrelevant visual factors are randomized at the image level.

4.2 Contact Refinement: Human-in-the-loop Residual Reinforcement Learning

Once the robot reaches the pre-contact region, the remaining challenge is to establish local contact. This stage is dominated by small pose errors and force feedback. These factors are difficult to model accurately, but can be learned through real-world interaction. We therefore utilize human-in-the-loop real-world residual RL to finish contact refinement.

Base Policy Pre-training. We first collect a set of demonstration trajectories: $\mathcal{D}_{\text{expert}} = \left\{(\tilde{o}_t, a_t^{\text{expert}})\right\}$, where a_t^{expert} denotes the demonstrated action. We use the Action-Chunking Transformer (ACT) [55, 30] as our base policy, which is trained by behavior cloning:

$$\theta_{\text{base}}^* = \arg\min_{\theta} \mathbb{E}_{(\tilde{o}_t, a_t^{\text{expert}}) \sim \mathcal{D}_{\text{expert}}} \left[\left\| \pi_\theta^{\text{base}}(\tilde{o}_t) - a_t^{\text{expert}} \right\|_2^2 \right]. \tag{9}$$

Algorithm 1 Semantics-to-Contact

Randomly initialize Critic Q_i (set targets $\tilde{Q}_i = Q_i$) for $i = 1, 2$ and Actor π_θ^{res}. Choose discount γ, temperature η, batch size N, delay ratio c, $\mathcal{D}_{online}$ and $\mathcal{D}_{offline}$. Load base policy π_θ^{base}.
while True **do**
  for $t = 0$ to T **do**
    Get focused region with $\Phi_{\text{VLM}}(P_t^{side}, l)$
    Get augmented images with Eq. 8 $\tilde{o}_t = \left(o_t^p, \tilde{P}_{t,1}^{\text{wrist}}, \tilde{P}_{t,2}^{\text{wrist}}\right)$,
    Take residual action $a_t^{res} \sim \pi_\theta^{res}(\cdot|\tilde{o}_t, a_t^{base})$ and base action $a_t^{base} \sim \pi_\theta^{base}(\cdot|\tilde{o}_t)$
    $a_t^{RL} = a_t^{base} + a_t^{res}$ or Take human intervention $a_t^H \sim \pi_H(\cdot|o_t)$
    $\mathcal{D}_{\text{online}} \leftarrow \mathcal{D}_{\text{online}} \cup \{(\tilde{o}_t, a_t^{RL/H}, r_t, \tilde{o}_{t+1})\}$.
    $\mathcal{D}_{\text{offline}} \leftarrow \mathcal{D}_{\text{offline}} \cup \{(\tilde{o}_t, a_t^{\text{H}}, r_t, \tilde{o}_{t+1})\}$.
    for 1 to c **do**
      Sample batch b_{online} of $\frac{N}{2}$ from $\mathcal{D}_{online}$ and batch $b_{offline}$ of $\frac{N}{2}$ from $\mathcal{D}_{offline}$
      $y_t = r_t + \gamma \min_{i\in\{1,2\}} \tilde{Q}_i(\tilde{o}_{t+1}, a_{t+1}^{RL}) + \gamma\eta \log \pi_\theta^{res}(\cdot|\tilde{o}_{t+1}, a_t^{base})$ with $\mathbf{b}$
      Update Q networks with $\mathcal{L}^{critic} = \frac{1}{N}\sum(y_t - \min_{i=\{1,2\}} Q_i(\tilde{o}_t, a_t^{RL}))^2$
    end for
    With $\mathbf{b}$, update π_θ^{res} maximizing objective: $\mathbb{E}\left[\min_{i=\{1,2\}} Q_i(\tilde{o}_t, a_t^{RL}) - \eta \log \pi_\theta^{res}(\cdot|\tilde{o}_t, a_t^{base})\right]$
  end for
end while

The ACT policy provides a reasonable initial behavior and reduces the amount of random exploration required during RWRL. However, because behavior cloning is limited by demonstration coverage, it may still fail under visual shifts, pose perturbations, and contact uncertainty.

Residual Reinforcement Learning. Instead of replacing the ACT policy, we use it as a base policy and learn a residual correction:

$$a_t^{RL} = a_t^{\text{base}} + \Delta a_t, \qquad a_t^{\text{base}} = \pi_\theta^{\text{base}}(\tilde{o}_t), \qquad \Delta a_t \sim \pi_\theta^{\text{res}}(\cdot \mid \tilde{o}_t, a_{base}, g). \tag{10}$$

Here, π_θ^{res} is the residual policy optimized by RWRL. The residual action Δa_t only needs to correct errors of the ACT policy. This significantly reduces the effective exploration space compared with learning the full action policy from scratch.

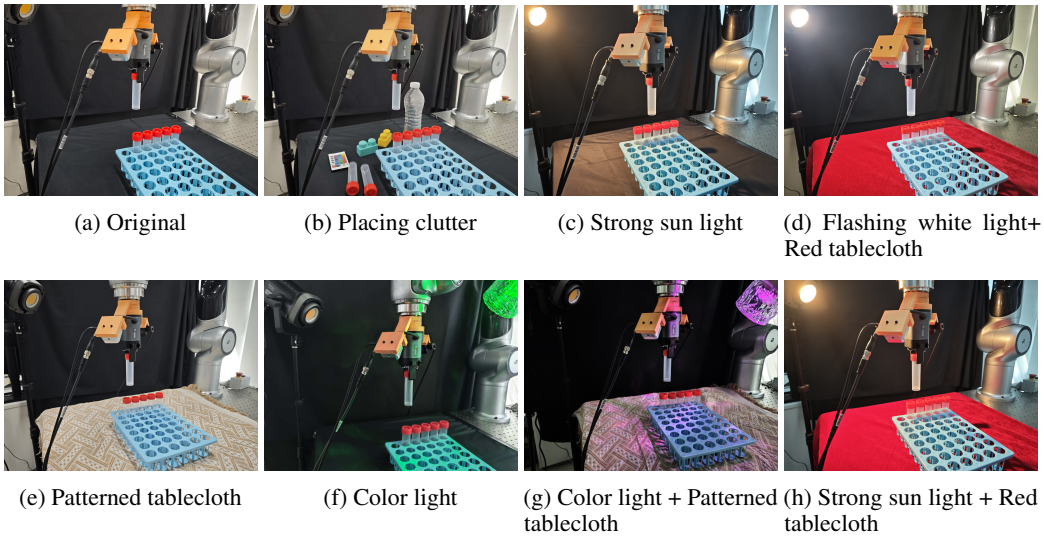
We update our optimization objective with:

$$\pi_\theta^{\text{res}*} = \arg\max_{\pi_\theta^{\text{res}}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha\mathcal{H}\left(\pi_\theta^{\text{res}}(\cdot|\tilde{o}_t, a_{base}, g)\right)\right)\right], \tag{11}$$

Human-in-the-loop Residual Reinforcement Learning System. During training, we deploy the VLM as the server side, where images captured by a third-person camera are used by the VLM to annotate object positions. The base policy and RL act as the client side and drives the robotic arm through impedance control, while the views from two wrist cameras serve as the visual inputs for RL. Meanwhile, during the human-in-the-loop process, we also adopt a client-server architecture within the RWRL. The client is responsible for interacting with the environment, either by allowing the robot to act autonomously according to the base policy or by enabling human intervention. The server is responsible for updating the actor and critic components of the RL policy. We use ResNet-10 as a pretrained encoder and follow HIL-SERL [25] to transform the state representation from the robot base frame into an end-effector-centered coordinate frame. Algorithm 1 presents the overall procedure of the S2C framework.

5 Experiments

Experiment Settings. To conduct a comprehensive evaluation, we quantify the performance of S2C in six contact-rich manipulation tasks from the following perspectives: 1) *Robustness*: How well does S2C perform with diverse contact-rich manipulation tasks under position and visual changes? 2)



(a) Original (b) Placing clutter (c) Strong sun light (d) Flashing white light+ Red tablecloth

(e) Patterned tablecloth (f) Color light (g) Color light + Patterned tablecloth (h) Strong sun light + Red tablecloth

Figure 2: Different visual robustness settings.

Complementarity: Is each component of S2C necessary? 3) *Scalability*: Where are the scalability boundaries of S2C?

We use the Agile Diana force-controlled robotic arm, which has 7 degrees of freedom per arm and employs impedance control. However, interaction with the arm is still achieved by sending end-effector pose commands. For teleoperation, we use a 3D SpaceMouse to send incremental end-effector changes to the robotic arm during both human-in-the-loop operation and data collection. The cameras we employ are RealSense D405 and Kingfisher depth cameras. VLM prompts can be checked in Appendix D.

Task Description. We design six contact-rich atomic skill manipulation tasks, including: (1) *Charger grasp-insertion*: Grasp a charger and put it into a power strip. (2) *USB insertion (easy)*: Insert a USB plug into a fixed charger. (3) *USB insertion (hard)*: Insert a USB plug into a charger wherever it is located. (4) *Test tube insertion (easy)*: Insert the test tube into the fixed well slot. (5) *Test tube insertion (medium)*: Insert a row of six test tubes in sequence. (6) *Test tube insertion (hard)*: Insert into the empty slots among a row of test tubes, where the empty positions are random. Please refer to Appendix A for other task description and training details. Appendix B displays the running setting for each baseline. The entire experiment is conducted in a natural industrial environment without any black cloth occlusion. Figure 2 presents our different visual robustness settings.

Metrics. In our experiments, we adopt two key metrics: (1) *Success Rate (SR)*: The ratio of successful task completions within the allotted time to the total number of test trials. (2) *Training Time (TT)*: Across all experimental settings, we record the online training time (min).

Baselines. Our method is compared with five other approaches: (1) *ACT* [55]: Action Chunking with Transformers, a method that leverages transformer architecture to generate action sequences in chunks for robotic manipulation. (2) *SERL* [24]: A RWRL framework without human-in-the-loop. (3) *Octo-small* [34]: A large transformer-based policy trained on 800k trajectories from the Open X-Embodiment dataset. (4) *HIL-SERL*: The first human-in-the-loop vision-based RWRL system. (5) *HIL-ConRFT* [6]: A RWRL reinforced fine-tuning approach for VLA model.

5.1 Evaluate the impact of position and visual changes in contact-rich tasks.

We first evaluate whether the learned policies can generalize to spatial shifts and visual appearance changes in real-world contact-rich manipulation. Table 1 reports training time and success rate under the black-tablecloth setting with positional generalization. Imitation learning or pretrained policies, including ACT and Octo-small, achieve only limited success, indicating that behavior cloning are insufficient for precise contact-rich execution under shifted target positions. SERL without human intervention fails on all tasks. In contrast, HIL-SERL and HIL-ConRFT achieve average success rates

Table 1: Training time and success rate under training setting. Each setting is tested 20 times. We implement HIL-SERL and HIL-ConRFT following the experimental settings described in their original papers; therefore, they did not use the same VLM-based pre-contact localization and motion-planning initialization as S2C. To ensure consistency and comparison of the success rate and training time, all methods are compared within the same workspace and start from the same initial region in each experiment. We also use the same evaluation protocol and success criterion for all methods, such that a trial is counted as successful only when the object is fully inserted into the target slot. For training time, we report the learner-side actor–critic optimization time because real-robot rollout collection and actor–critic optimization run as separate processes.

Method	Charger		USB-Easy		USB-Hard		Tube-Easy		Tube-Medium		Tube-Hard		Avg. SR
	TT	SR	TT	SR	TT	SR	TT	SR	TT	SR	TT	SR	
ACT	–	0/20	–	1/20	–	0/20	–	2/20	–	0/20	–	0/20	2.50%
SERL	120	0/20	120	0/20	120	0/20	120	0/20	120	0/20	120	0/20	0.00%
Octo-small	–	0/20	–	2/20	–	0/20	–	5/20	–	0/20	–	0/20	5.83%
HIL-SERL	**30**	19/20	60	**20/20**	120	18/20	30	**20/20**	120	16/20	120	16/20	**90.83%**
HIL-ConRFT	**30**	18/20	40	**20/20**	**60**	19/20	**20**	**20/20**	100	16/20	100	**17/20**	**91.76%**
S2C	**30**	**20/20**	**30**	**20/20**	**60**	**20/20**	**20**	**20/20**	**90**	**17/20**	**90**	16/20	**94.17%**

Table 2: Success rate under visual and positional robustness. Only HIL-based RWRL methods and S2C are evaluated. Each setting is evaluated 20 times.

Method	Charger	USB-Easy	USB-Hard	Tube-Easy	Tube-Medium	Tube-Hard	Avg. SR
Patterned tablecloth + Position							
HIL-SERL	0/20	0/20	0/20	0/20	0/20	0/20	0.00%
HIL-ConRFT	0/20	0/20	0/20	0/20	0/20	0/20	0.00%
S2C	**18/20**	**15/20**	**11/20**	**13/20**	**14/20**	**12/20**	**69.17%**
Strong sun light + Position							
HIL-SERL	0/20	0/20	0/20	0/20	0/20	0/20	0.00%
HIL-ConRFT	0/20	0/20	0/20	0/20	0/20	0/20	0.00%
S2C	**20/20**	**17/20**	**15/20**	**20/20**	**16/20**	**13/20**	**84.17%**
Color light + Position							
HIL-SERL	0/20	0/20	0/20	0/20	0/20	0/20	0.00%
HIL-ConRFT	0/20	0/20	0/20	0/20	0/20	0/20	0.00%
S2C	**18/20**	**18/20**	**19/20**	**16/20**	**12/20**	**12/20**	**79.17%**
Flashing white light + Red tablecloth + Position							
HIL-SERL	0/20	0/20	0/20	6/20	2/20	3/20	9.17%
HIL-ConRFT	0/20	0/20	0/20	8/20	4/20	4/20	13.33%
S2C	**18/20**	**19/20**	**17/20**	**17/20**	**15/20**	**13/20**	**82.50%**

of 90.83% and 91.76%, respectively, while S2C achieves 94.17%. The residual design and VLM guidance also enable S2C to complete the tasks with only a short training time.

Table 2 further evaluates the robustness of the learned policies under combined visual and positional changes. Although HIL-SERL and HIL-ConRFT perform well under the training black-tablecloth setting, their performance drops significantly when the background texture or lighting condition changes. This suggests that these policies are highly sensitive to task-irrelevant visual cues and tend to overfit to the appearance of the training scene. In contrast, S2C maintains substantially higher success rates in various visual disturbances, achieving an overall average success rate of 78.75% in all visual and positional robustness settings. S2C remains highly effective under red tablecloth and strong lighting conditions, with average success rates of 82.5% and 84.17%, respectively. Performance decreases under more challenging tablecloth and color light perturbations, but S2C still achieves success rates of 69.17 % and 79.17 % in these settings. These results indicate that VLM guidance and Object-centric visual augmentation improve robustness to visual background and lighting variations by enforcing the policy to focus on the contact-relevant regions.

5.2 Verify the complementarity of each core component of S2C.

We further conduct ablation experiments to verify the complementarity of the core components of S2C. Table 3 evaluates different variants under a challenging setting that combines strong sunlight, a

Table 4: Comparison of perception and VLM backbones on the charger insertion task. Inference time denotes the time required to localize the target in each episode. Success rate is evaluated over 10 real-robot trials.

Perception/VLM Backbone	**Inference Time (s)** $\downarrow$	**Success Rate** $\uparrow$
YOLOE	**0.07**	0/10 (0%)
SAM 3	0.12	3/10 (30%)
Qwen3-VL-8B	2.00	5/10 (50%)
GPT-5.5 API	15.00	**10/10 (100%)**
MolmoPoint-8B	2.00	**10/10 (100%)**

red tablecloth, and positional perturbations. HIL-SERL and HIL-ConRFT also use standard image augmentations in their original papers, including random cropping, random shifting. Removing either HIL or the contact module results in complete failure across all tasks. The failure of the w/o HIL variant indicates that real-world reinforcement learning remains extremely difficult without human corrections during unsafe or unproductive exploration. Similarly, the 0% success rate of w/o Contact demonstrates that semantic localization alone is insufficient for contact-rich manipulation: although the semantic module can guide the robot toward the target region, it cannot resolve fine-grained contact uncertainties or perform force-sensitive corrections during insertion.

To further evaluate robustness under extreme visual and spatial disturbances, we have performed a more fine-grained ablation study, as shown in Table 3. Under strong sunlight, a red tablecloth, and position variation, "w/o OC-Aug" fails on both tasks, whereas full S2C achieves 16/20 on Charger Insertion and 14/20 on USB Insertion. Similarly, in cluttered scenes, "w/o OC-Aug" achieves only 2/20 and 1/20, compared with 16/20 and 15/20 for full S2C. Because VLM guidance is retained in this ablation, these large performance gaps directly show that reliable semantic localization alone is insufficient to prevent the local policy from overfitting to task-irrelevant visual cues. nevertheless, the VLM guidance provides an auxiliary role. With object-centric enhancement retained, "w/ no VLM achieves 10/20 and 12/20 under combined lighting, background, and position perturbation, and 13/20 and 10/20 in cluttered scenes. Restoring VLM guidance improves these results to 16/20 and 14/20 in the former setting and to 16/20 and 15/20 in the latter. The effect is particularly pronounced for USB Insertion in clutter, where VLM guidance improves success from 10/20 to 15/20. The assistance of VLM reduces the difficulty of the task.

Table 3: Fine-grained ablation of the semantic components. The ablation is conducted on Charger Insertion and USB Insertion (hard). "w/o VLM" removes VLM-guided motion planning, whereas "w/o OC-Aug" removes object-centric augmentation. Each setting is evaluated 20 times.

Method	Charger	USB (Hard)	Avg. SR
Strong sun light + Red tablecloth + Position			
w/o HIL	0/20	0/20	0.00%
w/o VLM	10/20	12/20	55.00%
w/o OC-Aug	0/20	0/20	0.00%
w/o Contact	0/20	0/20	0.00%
S2C	**16/20**	**14/20**	**75.00%**
Placing clutter around objects			
w/o HIL	0/20	0/20	0.00%
w/o VLM	13/20	10/20	57.50%
w/o OC-Aug	2/20	1/20	7.50%
w/o Contact	0/20	0/20	0.00%
S2C	**16/20**	**15/20**	**77.50%**

During preliminary model selection, we have evaluated several perception and vision-language models, including lightweight models such as YOLOE and SAM3 [4], as well as larger models such as Qwen3-VL [3], and the GPT-5.5 API [31]. Table 2 illustrates comparison of different VLM backbones on the charger insertion task. Although GPT-5.5 provided accurate predictions, its inference latency was too high for practical deployment. YOLOE and SAM3 produced results quickly, but were easily distracted by nearby objects with similar colors and struggled to localize targets accurately from semantic instructions. Qwen3-VL was less reliable than MolmoPoint on downstream tasks such as affordance prediction and point-based localization. They also occasionally produced large localization errors that could pose safety risks during robot execution. Considering the trade-off between inference speed and localization accuracy, we selected MolmoPoint.

5.3 Explore the scalability boundaries of S2C.

To understand the generalization boundary of S2C, we conduct a more rigorous evaluation of its generalization and robustness across the following settings: (1) Task A: variations in objects around the workspace. (2) Task B: tablecloths of different colors. (3) Task C: out-of-distribution objects appearing within the white power strip. (4) Task D: strong light exposure, static or flashing. (5) Task E: exposure to color light. (6) Task F: slight external forces applied to the robotic arm. (7) Task G: indoor lights are turned on or off. (8) Task H: slight adjustments to the insertion position within a small range during execution. (9) Task I: replacement with a black charger.

As shown in Table 5, S2C exhibits strong robustness to Task A, B, D, E, F, H, G. However, the evaluation also reveals clear failure boundaries. S2C performs poorly when out-of-distribution objects appear inside the white power strip, especially for the Charger task, where the success rate drops to 0/10. Another important boundary appears when replacing the charger with a black charger. Both Charger and USB-Easy fail in this setting, indicating that S2C still struggles with severe object-level appearance shifts. Generally, these results illustrate that S2C generalizes well to environmental variations, including clutter in the workspace, changes in tablecloths, changes in illumination, slight physical disturbances, and small positional changes. Its main limitations arise from perturbations that change the appearance and geometry of the task-critical contact region.

Table 5: Scalability evaluation of S2C across different tasks. Each task is tested 10 times.

Task	A	B	C	D	E	F	G	H	I
Charger	10/10	9/10	0/10	10/10	8/10	10/10	10/10	10/10	0/10
USB-Easy	10/10	8/10	3/10	10/10	8/10	10/10	10/10	10/10	0/10

Table 6: Performance of S2C on additional contact-rich removal tasks

Task	Training Time (min)	Success Rate (%)
Test-tube Removal	25	90 (9/10)
Hard-drive Removal	35	100 (10/10)

To add the task diversity and generalization, We have extended the application of S2C to some removal settings. To evaluate whether S2C generalizes beyond insertion tasks, we further tested the framework on two removal-oriented manipulation tasks: (1) grasping and extracting a test tube from a tube slot, and (2) grasping and removing a hard drive from a drive slot. These tasks require the robot to establish a stable grasp and execute a constrained extraction motion, rather than inserting an object into a target receptacle. In this experiment, we choose diffusion policy [7] as the base policy. We conducted 10 independent trials for each task in Table 6, and S2C successfully completed all trials, achieving a success rate of 100% and 90% on both tasks. These results provide additional evidence that the proposed semantic-to-contact decomposition is applicable not only to insertion, but also to contact-rich removal tasks.

6 Conclusion

We present **Semantics-to-Contact**, a stagewise framework for robust contact-rich manipulation. S2C decomposes contact-rich manipulation into two stages: semantic focusing and contact refinement, where VLM-guided motion planning first brings the robot to a task-relevant pre-contact region, and Object-centric visual augmentation with human-in-the-loop real-world residual RL then enables visually robust, force-aware contact control. Across six real-world contact-rich tasks, S2C achieves strong robustness under positional shifts and diverse visual disturbances, demonstrating the effectiveness of integrating open-vocabulary perception with force-aware real-world learning. Future work will focus on extending S2C to robust long-horizon manipulation tasks.

Acknowledgments

This work is supported in part by NSFC under Grant 62606447, Shenzhen Science and Technology Program under grant KJZD20240903104008012, Shenzhen Science and Technology Program under grant ZDCY20250901113000001, CUHK-CUHK(SZ)-GDSTC Joint Collaboration Fund No. 2025A0505000053, and GuangDong Key Laboratory of Big Data Computing (2021B1212040002).

References

[1] Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. Residual off-policy rl for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025.

[2] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement-residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations, (ICLR)*, 2026.

[5] Chao Chen, Jiacheng Xu, Weijian Liao, Hao Ding, Zongzhang Zhang, Yang Yu, and Rui Zhao. Focus-then-decide: Segmentation-assisted reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence, (AAAI)*, volume 38, pages 11240–11248, 2024.

[6] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research, (IJRR)*, 44(10-11):1684–1704, 2025.

[8] Christopher Clark, Yue Yang, Jae Sung Park, Zixian Ma, Jieyu Zhang, Rohun Tripathi, Mohammadreza Salehi, Sangho Lee, Taira Anderson, Winson Han, et al. Molmopoint: Better pointing for vlms with grounding tokens. *arXiv preprint arXiv:2603.28069*, 2026.

[9] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International conference on machine learning (ICML)*, 2019.

[10] Jiameng Fan and Wenchao Li. Dribo: Robust deep reinforcement learning via multi-view information bottleneck. In *International conference on machine learning (ICML)*, 2022.

[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning, (ICML)*, 2018.

[12] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations (ICLR)*, 2021.

[13] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems (NeurIPS)*, 2021.

[14] Zhanpeng He, Yifeng Cao, and Matei Ciocarlie. Uncertainty comes for free: Human-in-the-loop policies with diffusion models. *arXiv preprint arXiv:2503.01876*, 2025.

[15] Zheyuan Hu, Robyn Wu, Naveen Enock, Jasmine Li, Riya Kadakia, Zackory Erickson, and Aviral Kumar. Rac: Robot learning for long-horizon tasks by scaling recovery and correction. *arXiv preprint arXiv:2509.07953*, 2025.

[16] Yunfan Jiang, Chen Wang, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. *arXiv preprint arXiv:2405.10315*, 2024.

[17] Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 international conference on robotics and automation (ICRA)*, 2019.

[18] Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025.

[19] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[20] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems (NeurIPS)*, 2020.

[21] Jeong Woon Lee and Hyoseok Hwang. Fourier guided adaptive adversarial augmentation for generalization in visual reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[22] Guanxing Lu, Rui Zhao, Haitao Lin, He Zhang, and Yansong Tang. Human-in-the-loop online rejection sampling for robotic manipulation. *arXiv preprint arXiv:2510.26406*, 2025.

[23] Shuo Lu, Yinuo Xu, Kecheng Yu, Siru Jiang, Yongcan Yu, Yubin Wang, Haitao Yang, Yuxiang Zhang, Bin Wang, Ran He, et al. Worldcoder-bench: Benchmarking physically grounded 3d world synthesis. *arXiv preprint arXiv:2606.01869*, 2026.

[24] Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969. IEEE, 2024.

[25] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.

[26] Guanren Qiao, Sixu Lin, Ronglai Zuo, Zhizheng Wu, Kui Jia, and Guiliang Liu. Signbot: Learning human-to-humanoid sign language interaction. *arXiv preprint arXiv:2505.24266*, 2025.

[27] Guanren Qiao, Ruixiang Ouyang, Sheng Xu, Ruixing Jin, Yueci Deng, Yunxin Tai, Kui Jia, and Guiliang Liu. Focus-then-contact: Speeding up robotic contact-rich task learning with affordance-guided real-world residual reinforcement learning. In *Forty-third International Conference on Machine Learning, (ICML)*, 2026.

[28] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[29] Carl Orge Retzlaff, Srijita Das, Christabel Wayllace, Payam Mousavi, Mohammad Afshari, Tianpei Yang, Anna Saranti, Alessa Angerschmid, Matthew E Taylor, and Andreas Holzinger. Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. *Journal of Artificial Intelligence Research (JAIR)*, 79:359–415, 2024.

[30] Qiannan Shen and Jing Zhang. MFTFormer: Meteorological-Frequency-Temporal Transformer with Block-Aligned Fusion for Traffic Flow Prediction. In *2026 9th International Symposium on Big Data and Applied Statistics (ISBDAS)*, pages 248–251. IEEE, 2026. doi: 10.1109/isbdas69350.2026.11484097. URL https://doi.org/10.1109/isbdas69350.2026.11484097.

[31] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[32] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018.

[33] Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025.

[34] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[35] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2025. URL https://github.com/HumanSignal/label-studio. Open source software available from https://github.com/HumanSignal/label-studio.

[36] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, (ICCV)*, pages 24591–24602, 2025.

[37] Jiahui Wang, Chao Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. Focus-then-reuse: Fast adaptation in visual perturbation environments. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems, (NeurIPS)*, 2025.

[38] Shuo Wang, Zhihao Wu, Xiaobo Hu, Jinwen Wang, Youfang Lin, and Kai Lv. What effects the generalization in visual reinforcement learning: policy consistency with truncated return prediction. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2024.

[39] Xudong Wang, Long Lian, and Stella X Yu. Unsupervised visual attention and invariance for reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2021.

[40] Yi Wang, Xinchen Li, Pengwei Xie, Pu Yang, Buqing Nie, Yunuo Cai, Qinglin Zhang, Chendi Qu, Jeffrey Wu, Jianheng Song, Xinlin Ren, Jingshun Huang, Mingjie Pan, Siyuan Feng, Zhi Chen, and Jianlan Luo. Learning while deploying: Fleet-scale reinforcement learning for generalist robot policies, 2026. URL https://arxiv.org/abs/2605.00416.

[41] Yujia Wang, Yuyan Li, Jiuming Liu, Fang-Lue Zhang, Xinhu Zheng, and Neil A. Dodgson. RL-ScanIQA: Reinforcement-Learned Scanpaths for Blind 360° Image Quality Assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 37401–37412, 2026. URL https://arxiv.org/abs/2603.14297.

[42] Wenke Xia, Yichu Yang, Hongtao Wu, Xiao Ma, Tao Kong, and Di Hu. Human-assisted robotic policy refinement via action preference optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[43] Linqi Xiao, Abhishek Bhattacharyya, Aparaajitha Gomathinayakam Latha, Venkateswarlu Gudepu, and Andrea Fumagalli. Robotic process automation testing and validation platform for open-source 5g stack. In *2025 IEEE 30th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–2. IEEE, 2025.

[44] Linqi Xiao, Aparaajitha Gomathinayakam Latha, Balagangadhar Bathula, Tianliang Zhang, and Andrea Fumagalli. Optical network robotic automation platform for multi-vendor optical transport network: accelerating validation, troubleshooting, and proof of concept efforts. In *2025 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2025.

[45] Linqi Xiao, Aparaajitha Gomathinayakam Latha, Venkateswarlu Gudepu, and Andrea Fumagalli. A yang-grounded llm agent supporting multi-vendor openroadm optical transport networks. In *Proceedings of the ACM SIGCOMM 2026 Conference*, pages 2109–2115, 2026.

[46] Charles Xu, Qiyang Li, Jianlan Luo, and Sergey Levine. Rldg: Robotic generalist policy distillation via reinforcement learning. *arXiv preprint arXiv:2412.09858*, 2024.

[47] Qinwen Xu, Jiaming Liu, Rui Zhou, Shaojun Shi, Nuowei Han, Zhuoyang Liu, Chenyang Gu, Shuo Gu, Yang Yue, Gao Huang, et al. Twinrl-vla: Digital twin-driven reinforcement learning for real-world robotic manipulation. *arXiv preprint arXiv:2602.09023*, 2026.

[48] Sheng Xu, Ruixing Jin, Huayi Zhou, Bo Yue, Guanren Qiao, Yunxin Tai, Yueci Deng, Kui Jia, and Guiliang Liu. From reaction to anticipation: Proactive failure recovery through agentic task graph for robotic manipulation. *arXiv preprint arXiv:2605.11951*, 2026.

[49] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations (ICLR)*, 2020.

[50] Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. Rl-vigen: A reinforcement learning benchmark for visual generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[51] Hongyu Zang, Xin Li, and Mingzhong Wang. Simsr: Simple distance-based state representations for deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2022.

[52] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

[53] Xiang Zhang, Masayoshi Tomizuka, and Hui Li. Bridging the sim-to-real gap with dynamic compliance tuning for industrial insertion. In *2024 IEEE International Conference on Robotics and Automation, (ICRA)*, pages 4356–4363. IEEE, 2024.

[54] Zhewei Zhang, Puyue Wang, Guanren Qiao, Yijie Weng, Jiawei Hu, Guo Li, Lujia Wang, Junyan Wang, Tao Gu, Hongliang Lu, et al. Lira: Local cross-layer information routing for vision-language-action decoding. *arXiv preprint arXiv:2608.07596*, 2026.

[55] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[56] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, pages 4396–4415, 2022.

Supplementary Materials

A Task Setup and Policy Training Details

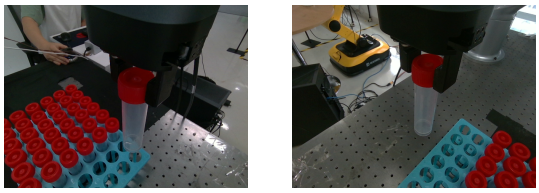
In this section, we provide details regarding how each task is set up, including hardware and software; as well as details on policy training.

A.1 Tube Insertion (easy)

In the Tube Insertion task, the robot is required to insert a cylindrical test tube into a fixed slot on a test-tube rack. The test tube is initially grasped by the robot and positioned above the rack with small variations in position and orientation. Successful completion of the task requires the robot to align the tube with the target slot and carefully guide it downward until it is fully inserted.

A.1.1 Cropped Images

We cropped the images to focus on the task-relevant parts of the scene, as shown in Fig. 3.



wrist_1 wrist_2

Figure 3: Sample input images from cameras used as inputs to the policy.

A.1.2 Policy Training Details

In Table 7, we report additional details of the policy training for this task.

Table 7: Policy training details for the Tube Insertion (easy) task.

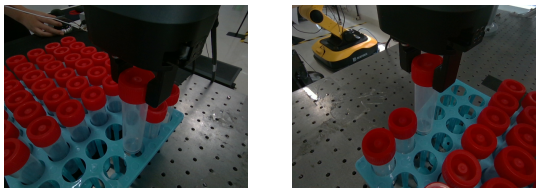
Parameter	**Value**
base policy	ACT
Observation space	wrist_1, wrist_2, tcp_pose, tcp_vel, tcp_f/t
Action space	6D twist + 1D gripper
Classifier views	wrist_1, wrist_2
Classifier accuracy	99%
Initial offline demonstrations	30
Max episode length	100 environment steps
Reset method	Auto reset
Randomization range	5 cm in x and y, 10 deg in rz
Proprio encoder size	128
Residual Policy MLP size	256×256
Discount factor	0.98
Optimizer	Adam
xyz bound	45cm*20cm*20cm
rpy bound	0*0*$[-10, 10]deg$

A.2 Tube Insertion (medium)

In the Sequential Tube Insertion task, the robot is required to insert a row of test tubes into multiple slots on a test-tube rack in a predefined order. Each test tube is grasped by the robot and must be aligned with the next target slot before insertion. Successful completion of the task requires the robot to repeatedly localize the correct empty slot, adjust the tube pose, and perform stable downward insertion without disturbing previously placed tubes.

A.2.1 Cropped Images

Fig. 4 shows the hardware setup for the motherboard assembly task, which presents the robot, the camera placements, and the task arrangement.



wrist_1 wrist_2

Figure 4: Sample input images from cameras used as inputs to the policy.

A.2.2 Policy Training Details

In Table 8, we report additional details of the policy training for this task.

Table 8: Policy training details for the Tube Insertion (medium) task.

Parameter	**Value**
base policy	ACT
Observation space	wrist_1, wrist_2, tcp_pose, tcp_vel, tcp_f/t
Action space	6D twist+ 1D gripper
Classifier views	wrist_1, wrist_2
Classifier accuracy	99%
Initial offline demonstrations	40
Max episode length	100 environment steps
Reset method	Auto reset
Randomization range	5 cm in x and y, 10 deg in rz
Proprio encoder size	128
Residual Policy MLP size	256×256
Discount factor	0.98
Optimizer	Adam
xyz bound	45cm*30cm*30cm
rpy bound	0*0*$[-10, 10]deg$

A.3 Charger grasp-insertion

In the Charger Grasp-Insertion task, the robot first grasps a charging plug and then precisely inserts it into a corresponding socket on a power strip. The plug is initially placed at a nearby location with small variations in pose. Successful completion of the task requires accurate grasping, reliable pose alignment between the plug and the socket, and careful control during the insertion phase to avoid misalignment or collisions. This task emphasizes sequential manipulation, fine-grained spatial alignment, and contact-aware insertion under tight geometric constraints.

A.3.1 Cropped Images

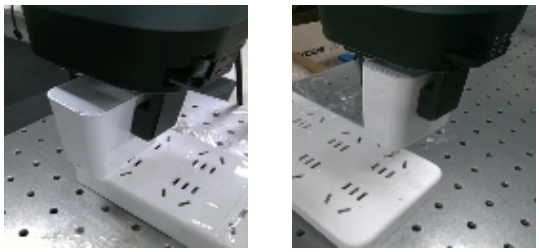
Fig. 5 shows the hardware setup for the motherboard assembly task, which presents the robot, the camera placements, and the task arrangement.

A.3.2 Policy Training Details

In Table 8, we report additional details of the policy training for this task.

A.4 Tube Insertion (hard)

In the Hard Tube Insertion task, the robot must insert a test tube into whichever slot is empty in a partially occupied test-tube rack. The empty slot varies across trials, requiring the robot to visually



wrist_1 wrist_2

Figure 5: Sample input images from cameras used as inputs to the policy.

Table 9: Policy training details for the Charger grasp-insertion.

Parameter	**Value**
base policy	ACT
Observation space	wrist_1, wrist_2, tcp_pose, tcp_vel, tcp_f/t
Action space	6D twist + 1D gripper
Classifier views	wrist_1, wrist_2
Classifier accuracy	99%
Initial offline demonstrations	38
Max episode length	100 environment steps
Reset method	Auto reset
Randomization range	5 cm in x and y, 30 deg in rz
Proprio encoder size	128
Residual Policy MLP size	256×256
Discount factor	0.98
Optimizer	Adam
xyz bound	45cm*30cm*30cm
rpy bound	0*0*$[-60, 60]deg$

identify the available opening before performing insertion. Successful completion requires accurate slot localization, precise alignment, and stable contact-aware insertion while avoiding collisions with neighboring tubes and the rack.

A.4.1 Cropped Images

Fig. 6 shows the hardware setup for the motherboard assembly task, which presents the robot, the camera placements, and the task arrangement.



wrist_1 wrist_2

Figure 6: Sample input images from cameras used as inputs to the policy.

A.4.2 Policy Training Details

In Table 10, we report additional details of the policy training for this task.

A.5 USB Insertion (easy)

In the USB Insertion (Easy) task, the robot starts with a USB plug already held in the gripper and inserts it into a USB port on a charging adapter. The charging adapter is fixed in position and

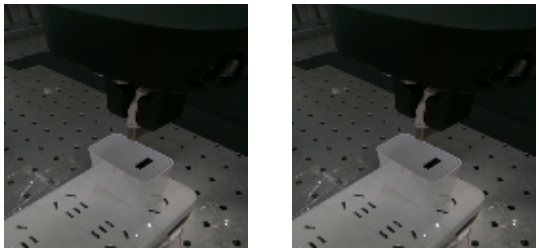
Table 10: Policy training details for the Tube Insertion (hard) task.

Parameter	Value
base policy	ACT
Observation space	wrist_1, wrist_2, tcp_pose, tcp_vel, tcp_f/t
Action space	6D twist + 1D gripper
Classifier views	wrist_1, wrist_2
Classifier accuracy	99%
Initial offline demonstrations	40
Max episode length	100 environment steps
Reset method	Auto reset
Randomization range	5 cm in x and y, 10 deg in rz
Proprio encoder size	128
Residual Policy MLP size	256×256
Discount factor	0.98
Optimizer	Adam
xyz bound	45cm*30cm*30cm
rpy bound	0*0*$[-10, 10]deg$

orientation throughout the task. Successful completion requires accurate alignment between the USB plug and the port and careful control during the insertion motion to avoid misalignment or collisions with the port edges. This variant primarily evaluates precise insertion control under constrained geometry with a known target pose.

A.5.1 Cropped Images

Fig. 7 shows the hardware setup for the motherboard assembly task, which presents the robot, the camera placements, and the task arrangement.



wrist_1 wrist_2

Figure 7: Sample input images from cameras used as inputs to the policy.

A.5.2 Policy Training Details

In Table 11, we report additional details of the policy training for this task.

A.6 USB Insertion (hard)

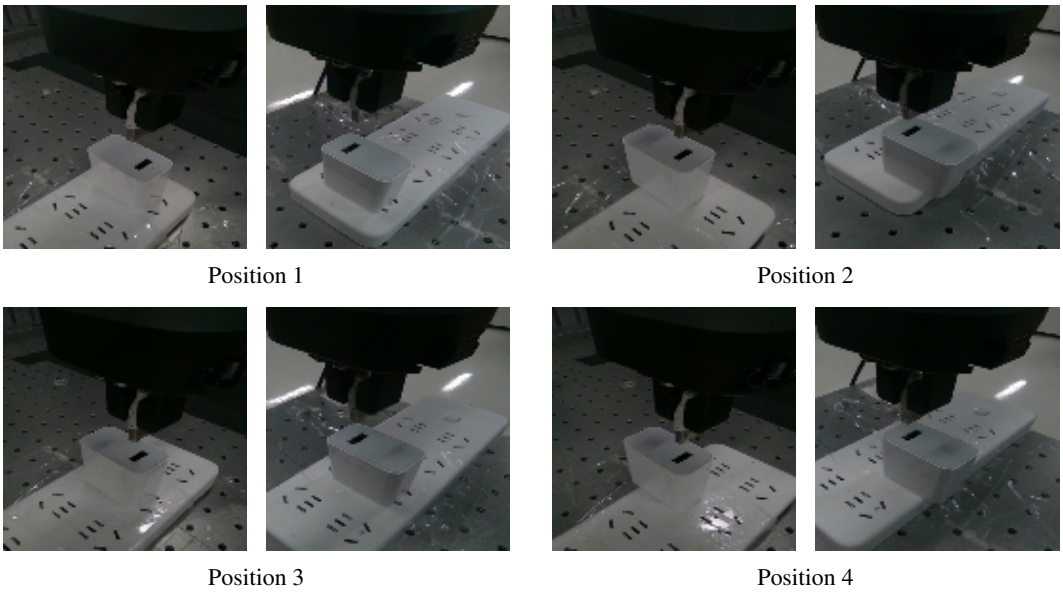
In the USB Insertion (Hard) task, the robot starts with a USB plug already held in the gripper and inserts it into a USB port on a charging adapter whose position is randomly selected from a set of predefined locations. The target port pose therefore varies across episodes. Successful completion requires the policy to first localize the target port and then perform accurate pose alignment and contact-aware insertion. Compared to the easy variant, this task introduces target pose uncertainty and emphasizes robustness to spatial variation in addition to precise insertion control.

A.6.1 Cropped Images

Fig. 8 shows the hardware setup for the motherboard assembly task, which presents the robot, the camera placements, and the task arrangement.

Table 11: Policy training details for the USB Insertion (easy) task.

Parameter	Value
base policy	ACT
Observation space	wrist_1, wrist_2, tcp_pose, tcp_vel, tcp_f/t
Action space	6D twist
Classifier views	wrist_1, wrist_2
Classifier accuracy	99%
Initial offline demonstrations	35
Max episode length	100 environment steps
Reset method	Auto reset
Randomization range	5 cm in x and y, 30 deg in rz
Proprio encoder size	128
Residual Policy MLP size	256×256
Discount factor	0.98
Optimizer	Adam
xyz bound	50cm*25cm*25cm
rpy bound	0*0*$[-60, 60]deg$



Position 1 Position 2

Position 3 Position 4

Figure 8: Sample input images from cameras used as inputs to the policy.

A.6.2 Policy Training Details

In Table 12, we report additional details of the policy training for this task.

B Running Setting

To ensure a fair comparison, we train Octo-small and ACT using 50 data samples. We train ACT for 10000 epochs and Octo-small for 20000 epochs. For SERL, HIL-ConRFT, HIL-SERL and S2C, we initially used 30-40 expert trajectories to fill $D_{offline}$. The π_{base} for S2C was pretrained with 40 data samples for 5000 epochs. SERL, HIL-ConRFT, and HIL-SERL all use BC to pretrain the actor and critic. SERL and HIL-SERL use an MLP network for BC, while HIL-ConRFT uses Octo-small. We deploy the VLM on A800 GPUs, communicating via client-server. The RWRL method was trained using 5090 GPUs, while the imitation learning method was trained using A800 GPUs.

Table 12: Policy training details for the USB Insertion (hard) task.

Parameter	Value
base policy	ACT
Observation space	wrist_1, wrist_2, tcp_pose, tcp_vel, tcp_f/t
Action space	6D twist
Classifier views	wrist_1, wrist_2
Classifier accuracy	99%
Initial offline demonstrations	35
Max episode length	100 environment steps
Reset method	Auto reset
Randomization range	10 cm in x and y, 30 deg in rz
Proprio encoder size	128
Residual Policy MLP size	256×256
Discount factor	0.98
Optimizer	Adam
xyz bound	50cm*25cm*25cm
rpy bound	0*0*$[-60, 60]deg$

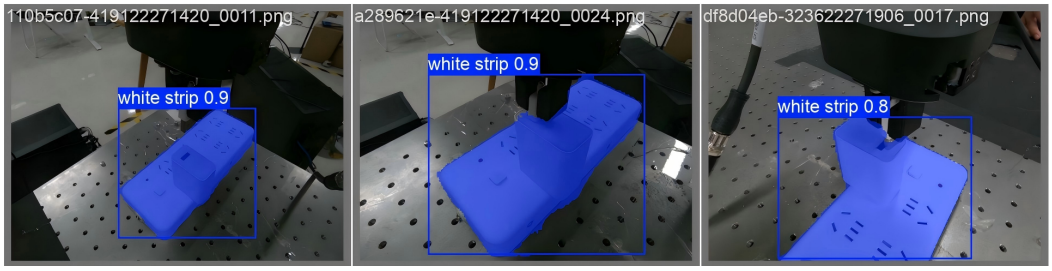


Figure 9: YOLOE segmentation validation examples.

C Visual Augmentation Details

In practice, promptable segmentation may be imperfect when objects are transparent or visually similar to the background. To improve mask quality, we optionally calibrate the YOLOE visual features with a lightweight linear probe. Specifically, given a small set of labeled foreground masks $\mathcal{D}_{\text{mask}} = \{(P_i, M_i^*)\}$, we freeze the YOLOE backbone and train a linear segmentation head h_ψ:

$$\psi^* = \arg\min_{\psi} \mathbb{E}_{(P_i, M_i^*) \sim \mathcal{D}_{\text{mask}}} \left[\mathcal{L}_{\text{seg}} \left(h_\psi \left(f_{\text{YOLOE}}(P_i) \right), M_i^* \right) \right], \tag{12}$$

We use the polygon tool in Label Studio [35] to annotate the categories to be fine-tuned. Figure 9 shows examples of the results on the validation set after fine-tuning.

D VLM instructions

USB (Easy / Hard): “Point to the square USB port on the white charger.” Charger: “Point to the center of the lower-left three-hole socket on the white power strip placed on the tablecloth.” Tube (Easy / Medium): “Point to the first empty slot from the top in the rightmost column of the blue test-tube rack.” Tube (Hard): “Point to the red test tube in the middle of the rightmost column of the blue test-tube rack.”

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have introduced the main claim in our abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:[Yes]

   Justification: We discussed our limitations in Section 5.3.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper is an application-oriented piece about robots, and does not involve theoretical proofs or assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have fully described the methods and experimental details in Sections 4, 5, and the supplementary materials. At the same time, we have many videos demonstrating the effectiveness of our method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: After acceptance, we will provide the code and data in the future.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer:[Yes]

Justification: We have fully described experimental details in Sections 5 and the supplementary materials.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For each task, we have evaluated multiple times under different settings and calculated the average success rate metric before calculating the success rate.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have fully described the computer resources in the supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: It is satisfied with https://neurips.cc/public/EthicsGuidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper is not related to a high-risk model or data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have provided proper citations for all of them.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new asset, dataset or benchmark.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our experiments are not related to human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our experiments are not related to human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used only for writing, editing, or formatting purposes

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.