CoRe-VLA: Preserving Cross-View Coordination in VLAs under Camera Shifts

Anonymous authors
Paper under double-blind review

Abstract

VLAs combine pretrained vision-language representations with action generation to enable language-guided control across diverse tasks, becoming a mainstream paradigm in embodied intelligence. However, multiple studies have reported VLA's substantial declines in task success under camera shifts, revealing a key vulnerability that limits reliable deployment. To address this vulnerability, existing methods collect paired observations of the same scene from different viewpoints to fine-tune the VLA or train visual adaptation modules. Unfortunately, they require additional data collection and VLA training costs. In this paper, we first identify *cross-view coordination breakdown* under external camera shifts: the robot may rely too heavily on wrist-view cues and consequently execute subtasks in the wrong order when losing global view. Motivated by this, we propose CoRe-VLA, a plug-and-play framework requiring neither additional multi-view data collection nor VLA fine-tuning, which can incorporate with exsiting VLAs. It reconstructs a scene point cloud and renders the observation from the VLA's training viewpoint to restore cross-view coordination. In CoRe-VLA, Render-to-Camera (R2C) Restoration reduces rendering-induced visual degradation, while Execution-Trajectory-Conditioned Alignment (ETCA) reduces robot idle time and mitigates motion conflicts during asynchronous execution. Experiments on 5 real-robot tasks, LIBERO-100 and LIBERO-Plus demonstrate CoRe-VLA substantially improves task success across mainstream VLAs under camera shifts. For example, CoRe-VLA raises $\pi_{0.5}$'s success rate from 13.3% to 83.3% at a 1.6m camera shift in real-robot environment.

1 Introduction

Embodied intelligence extends AI from understanding digital information to interacting with the physical world, enabling robots to perform useful tasks across a wide range of scenarios. Vision-language-action (VLA) models have emerged as a prominent approach to this goal, translating visual observations and language instructions into robot actions with strong performance across diverse manipulation tasks. OpenVLA (Kim et al., 2025b) is a pioneering work that combines pretrained vision-language representations with large-scale robot demonstrations to enable generalist robotic manipulation. π_0 (Black et al., 2025b) combines pretrained vision-language representations with flow matching for dexterous robot control, while $\pi_{0.5}$ (Black et al., 2025a) improves generalization to unfamiliar environments through co-training on heterogeneous data. GR00T (Bjorck et al., 2025) combines real-robot trajectories, human videos, and synthetic data to support generalist manipulation across robot embodiments. These mainstream open-source VLAs provide an accessible foundation for developing general-purpose robotic systems.

Despite these advances, reliable deployment of VLAs remains challenging, particularly because VLAs are highly sensitive to camera shifts. Such shifts are common and often difficult to avoid in practice, as external-camera placement varies across deployment environments and workspace configurations. Prior studies (Fei et al., 2026b; Heo et al., 2026) have reported substantial performance degradation under camera shifts in simulation and real-robot settings. As shown in Fig. 1(a), our evaluation on 5 real-robot tasks further demonstrates this vulnerability: mainstream VLAs' task success rates decline substantially as external-camera shift increases. For example, a shift of 0.4 m reduces $\pi_{0.5}$'s success rate from 96.7% to 66.7%. These findings highlight the gap in maneuverability between familiar viewpoints and after camera repositioning.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

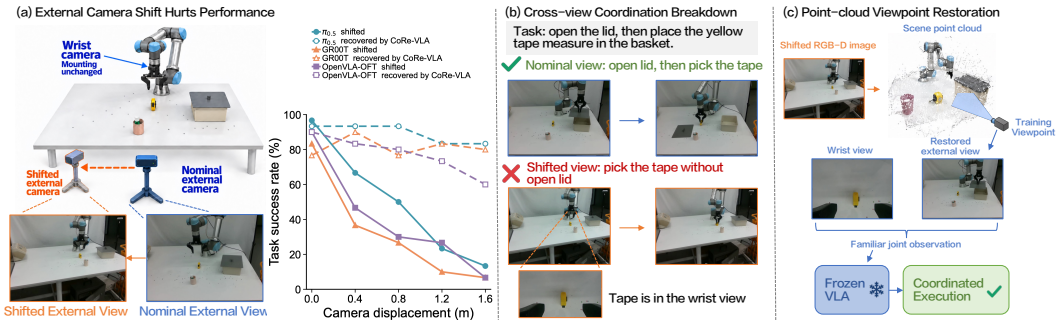


Figure 1: (a) The external camera shifts while the wrist-camera mounting remains unchanged. The accompanying curves compare task success rates on 5 real-robot tasks with and without CoRe-VLA. (b) Under the shifted view, the robot grasps the tape measure visible in the wrist view while the basket lid remains closed, violating the required subtask order. (c) The core idea of CoRe-VLA: using point-cloud reconstruction and re-rendering to restore the external observation and recover cross-view coordination.

Existing approaches try to address viewpoint sensitivity through VLA fine-tuning or training a visual adaptation module, introducing additional data preparation and VLA training costs. Cross-View Action Consistency (Huang et al., 2026b) fine-tunes VLAs for consistent action-flow predictions across viewpoints. AnyCamVLA (Heo et al., 2026) and GS-VLA (Park & Kim, 2026) instead restore familiar training views through pretrained novel-view synthesis and learned Gaussian-based canonicalization, respectively, while keeping the VLA frozen. Their visual adaptation modules depend on multi-view supervision for training or domain adaptation, which must be collected in real robotic environment, further increasing the already substantial cost of robot data collection.

These limitations raise a natural question: how to imporve existing VLAs' viewpoint robustness without additional multi-view data collection or VLA retraining? To guide the design of such an approach, we systematically analyze how these shifts affect robot execution. Our real-robot experiments reveal *cross-view coordination breakdown*: when the external camera shifts, the robot may rely too heavily on wrist-view cues and consequently execute subtasks in the wrong order. For example, in Fig. 1(b), under the shifted view, the robot grasps the tape measure visible in the wrist view while the basket lid remains closed, violating the required subtask order. Although the grasp succeeds, the action does not account for the lid-closed state visible in the external view. This finding motivates us to explore how to restore the nominal external view for the VLA to prevent cross-view coordination breakdown.

Point-cloud reconstruction and re-rendering provide a simple and effective way to achieve this goal, as illustrated in Fig. 1(c). A scene point cloud is reconstructed from RGB-D observations and rendered at the training-time external camera pose. This geometric approach requires neither additional multi-view demonstration collection nor VLA training or fine-tuning, making it suitable for integration with existing models. However, translating this idea into reliable robot control presents two challenges: (i) rendering degrades texture, color, and sharpness, and efficiently correcting these combined artifacts remains challenging, (ii) reconstruction and rendering increase latency in VLA inference, slowing task execution.

To address above challenges, we propose Coordination Recovery for Vision-Language-Action Models (**CoRe-VLA**), a framework built on point-cloud reconstruction and viewpoint transformation. To reduce rendering artifacts, Render-to-Camera (**R2C**) Restoration uses a lightweight NAFNet (Chen et al., 2022) trained on degraded original image pairs constructed from existing demonstration videos. Perceptual, color, and pixel losses jointly guide texture recovery, color correction, and detail reconstruction. To mitigate the additional latency from rendering and restoration, Execution-Trajectory-Conditioned Alignment (ETCA) is introduced for asynchronous control. Because the robot continues moving during inference, ETCA matches each newly predicted action chunk with the recorded execution trajectory to identify where execution should continue. This asynchronous control strategy substantially offsets the increase in task execution time introduced by CoRe-VLA while automatically reducing motion conflicts under high inference latency. Together, these components make CoRe-VLA a plug-and-play method for improving viewpoint robustness, requiring neither additional multi-view data collection nor VLA retraining. The main contributions are listed:

1. We identify cross-view coordination breakdown as a key failure mechanism in VLAs under external-camera shifts.
2. We propose ETCA, an asynchronous control strategy that can be applied to all robot policies to reduce task execution time and automatically mitigates motion conflicts.
3. We propose CoRe-VLA, a plug-and-play method that integrates with existing VLAs to preserve cross-view coordination without additional data collection or VLA retraining. Experiments show that CoRe-VLA substantially improves the viewpoint robustness of mainstream VLAs.

2 Related Work

VLAs and Viewpoint Robustness. The core idea of VLA models is to combine a pretrained VLM with an action generation module (Kim et al., 2025b; Black et al., 2025a; Bjorck et al., 2025). Because the VLM directly supplies task-relevant visual representations for downstream control, perturbations to visual inputs can disrupt action generation, and substantially reduce task success rates. LIBERO-Plus (Fei et al., 2026b) identifies camera-viewpoint perturbations as a major vulnerability, while VLA-Arena (Zhang et al., 2025) reports sharp degradation when camera shifts are introduced into its visual perturbation protocol. Li et al. (Li et al., 2026) further attribute viewpoint-induced failures primarily to spatial representation misalignment and show that lightweight visual adaptation can recover performance.

World Action Models World action models (WAMs) leverage pretrained video models to learn robot control while predicting how the scene evolves. DreamZero (Ye et al., 2026) jointly models video and actions to improve generalization to unseen tasks and environments, while Cosmos Policy (Kim et al., 2026) predicts actions, future states, and values for control and planning. However, evaluations on LIBERO-Plus (Fei et al., 2026b) and RoboTwin 2.0-Plus (Zhang et al., 2026) show that WAMs remain sensitive to camera shifts despite their robustness to lighting and noise perturbations (Zhang et al., 2026). Selective Cross-View Consistency (Huang et al., 2026a) addresses this limitation through training on same-state viewpoint pairs, improving performance on held-out extrapolation viewpoints. These findings suggest that adopting a WAM alone does not resolve viewpoint sensitivity, which still requires targeted treatment.

Point Cloud Reconstruction Point cloud can provide explicit geometry for rendering scene observations from arbitrary viewpoints. Relevant methods for reconstructing point cloud from image observations can be grouped into three categories: neural point-based rendering, Gaussian splatting, and direct RGB-D reprojection. Neural point-based methods, represented by NPBG (Aliev et al., 2020), learn point descriptors and a rendering network, but require scene-specific fitting that complicates deployment in changing embodied manipulation scenes. Gaussian-based methods such as 3D Gaussian Splatting (Kerbl et al., 2023) enable real-time rendering, yet their original pipelines require multi-view capture and per-scene optimization rather than immediate reconstruction from the current observation.

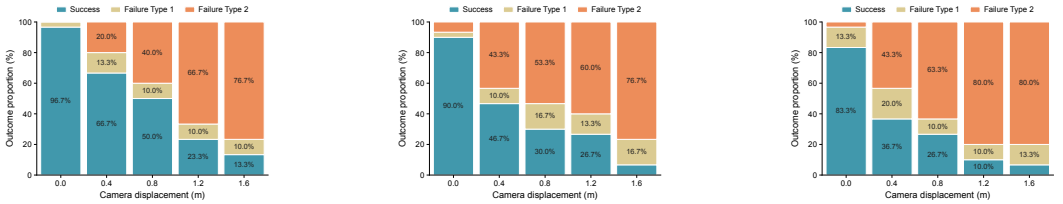
In contrast, direct RGB-D reprojection reconstructs and renders the observed geometry without scene-specific optimization, enabling efficient real-time viewpoint restoration (Schöps et al., 2017). However, its reconstruction accuracy depends strongly on depth accuracy and observation coverage. Depth errors, missing surfaces, and imperfect color alignment make texture loss, color distortion, and reduced sharpness difficult to avoid in practice (Zhou & Koltun, 2014; Fink et al., 2023). Online neural rendering methods such as LiveNVS (Fink et al., 2023) improve visual quality through multi-view feature fusion, but differ from restoring each current observation independently. Jointly reducing these degradations from a single RGB-D observation while maintaining low processing latency remains challenging, motivating our lightweight R2C restoration module.

3 Understanding VLA Failures under External-Camera Shifts

Real-World Diagnostic Setup. To investigate how to design a viewpoint-robust method that require neither multi-view data collection nor VLA training, we systematically analyze how external-camera shifts affect robot execution in multi-camera VLA systems. As a prerequisite, we construct a real-world benchmark using a UR7e robot equipped with an external RGB-D camera and a wrist-mounted camera. Detailed task definitions are provided in Table 1. The tasks cover basic manipula-

Table 1: Real-world task suite used to diagnose VLA failures under external-camera shifts. T1 evaluates basic manipulation, T2–T3 long-horizon execution, and T4–T5 spatial reasoning.

ID	Category	Task instruction
T1	Basic	Pick up the green glue stick and place it in the pink cup.
T2	Long-horizon	Place the yellow tape measure and the pink cup into the basket in sequence.
T3	Long-horizon	Open the lid, then pick up the yellow tape measure and place it in the basket.
T4	Spatial	Pick up the green glue stick and place it on the side of the pink cup farther from the basket.
T5	Spatial	Pick up the object closest to the basket and place it inside.



(a) $\pi_{0.5}$ (b) OpenVLA-OFT (c) GR00T

Figure 2: Outcome distributions on five real-robot tasks under external-camera shifts for (a) $\pi_{0.5}$, (b) OpenVLA-OFT, and (c) GR00T. Failure Type 1 denotes failure to complete any meaningful subtask, while Failure Type 2 denotes local manipulation inconsistent with the global task state.

tion (T1), long-horizon execution (T2-T3), and spatial reasoning (T4-T5). Full task instructions and success criteria are provided in Appendix 5.

Three mainstream VLA models are evaluated: $\pi_{0.5}$ (Black et al., 2025a), OpenVLA-OFT (Kim et al., 2025a), and GR00T (Bjorck et al., 2025). Each model is fine-tuned on 100 demonstrations per task collected at the nominal external-camera pose. During evaluation, the wrist-camera mounting remains fixed relative to the end effector, while the external camera is displaced by 0.4, 0.8, 1.2, or 1.6 m. At each distance, three camera positions are randomly sampled, with two fixed initial object layouts per task. Each task execution is classified as successful or as one of two failure types: Type 1 indicates failure to complete any meaningful subtask, such as moving irregularly or failing to grasp an object; Type 2 indicates a successful local manipulation that is inconsistent with the global task state. For example, the robot may grasp the tape measure before opening the lid in task 3. This taxonomy separates local manipulation failures from failures in selecting task-appropriate actions. The same platform and tasks are reused to evaluate CoRe-VLA in Sec. 5.

Quantitative and Qualitative Failure Analysis. Fig. 2 (a) shows the outcome distributions across external-camera shifts. Across the evaluated models, task success declines as displacement increases, while task-state inconsistency (Failure Type 2) becomes the dominant failure mode. For example, the success rate of $\pi_{0.5}$ drops from 98.0% at the nominal viewpoint to 13.3% at a displacement of 1.6 m, while Type 2 increases from 0% to 76.7% of all trials. In contrast, local execution failures (Failure Type 1) account for only 10.0% to 13.3% of trials under nonzero displacements. These results suggest that viewpoint shifts substantially impair the selection of task-appropriate actions, even when local manipulation remains feasible. In addition to the example discussed in Sec. 1, Task 5 in Fig. 8 in Appendix is also a typical example. The task requires selecting the object closest to the basket based on the global spatial relationships captured by the external camera. After the camera shifts, the robot instead grasps the cup visible in the wrist view, failing to coordinate this local cue with the global spatial constraint. By contrast, the task succeeds when the camera remains at its nominal position.

Together, these observations provide behavioral evidence of *cross-view coordination breakdown*: when the external camera shifts, the robot may rely too heavily on wrist-view cues and consequently execute subtasks in the wrong order. This finding motivates us to focus our method design on restoring the nominal external view for the VLA to prevent cross-view coordination breakdown.

4 METHODOLOGY

In this section, we present Coordination Recovery for Vision-Language-Action Models (CoRe-VLA). As shown in Fig. 3, CoRe-VLA comprises three components: Geometric Viewpoint Restoration, Render-to-Camera (R2C) Restoration, and Execution-Trajectory-Conditioned Alignment (ETCA).

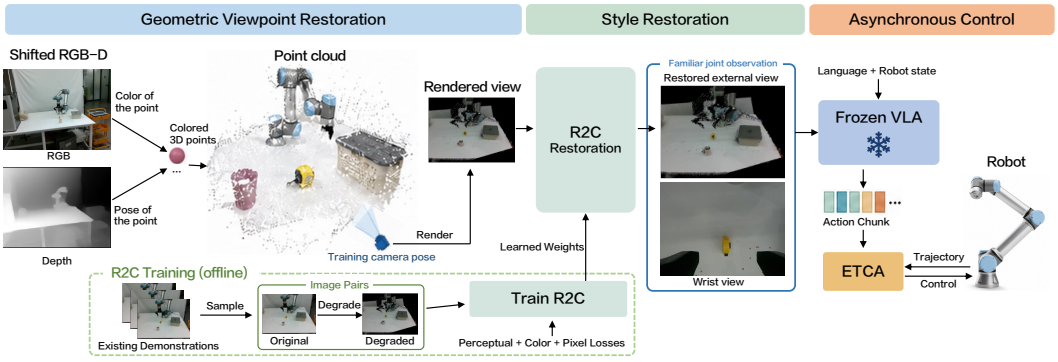


Figure 3: Overview of CoRe-VLA, comprising three components: geometric viewpoint restoration, R2C for image restoration, and ETCA for asynchronous control.

4.1 Geometric Viewpoint Restoration

As outlined in before, we adopt point-cloud reconstruction and re-rendering based on direct RGB-D reprojection to restore the nominal external view for the VLA. This strategy offers two advantages: (i) it provides a simple and efficient way to restore the training-view geometry, (ii) it requires neither training a view-transformation model nor collecting paired multi-view data.

Specifically, given an aligned RGB-D observation (G_s, D_s) from the shifted external camera, our goal is to synthesize an image G_p at the training camera pose. This process uses the source and target camera intrinsics, K_s and K_0, and the known rigid transformation $(R_{0\leftarrow s}, t_{0\leftarrow s})$ between their coordinate systems. For each pixel $\mathbf{u}$ in shifted external view with valid depth, its homogeneous coordinates $\tilde{\mathbf{u}}$ are first back-projected into 3D using D_s and K_s. The resulting point is transformed into the training camera coordinate system and projected onto the target image plane:

$$\begin{aligned} \mathbf{x}_0(\mathbf{u}) &= R_{0\leftarrow s}\left[D_s(\mathbf{u})K_s^{-1}\tilde{\mathbf{u}}\right] + t_{0\leftarrow s}, \\ \mathbf{v} &= \pi(K_0\mathbf{x}_0(\mathbf{u}))\,, \end{aligned} \tag{1}$$

where $\mathbf{x}_0(\mathbf{u})$ denotes the transformed 3D point and $\pi([x, y, z]^\top) = [x/z, y/z]^\top$ performs perspective division. Each point retains its source color $G_s(\mathbf{u})$, and the transformed colored point cloud is rasterized to produce G_p. By presenting the current scene in the training-view geometry, G_p provides a basis for restoring coordination between global task context and wrist-level visual cues.

4.2 Render-to-Camera Restoration

Geometric viewpoint restoration recovers the training-view geometry, but rendering from a sparse and imperfectly reconstructed point cloud can introduce loss of material and texture detail, color distortion, and reduced sharpness into G_p. Jointly reducing these three degradations from a single RGB-D observation while maintaining low processing latency is challenging. Non-learning method pipelines require manual tuning and can accumulate errors, while off-the-shelf deep-learning methods can not address these artifacts jointly. Multi-step diffusion methods can also incur substantial latency.

These considerations motivate a lightweight restoration module with supervision specifically tailored to rendering-induced degradations. In our proposed Render-to-Camera (R2C) Restoration, a lightweight NAFNet-based network (Chen et al., 2022) is trained with perceptual, color, and pixel losses to jointly promote texture recovery, color consistency, and fine-detail reconstruction. At inference time, the network f_θ maps the rendered image G_p to the restored external image G_r.

R2C is trained on image pairs constructed from existing 2D task demonstration videos, avoiding additional multi-view data collection (Fig. 3). Each original frame G_0 is paired with a degraded counterpart $\tilde{G}_0$ that reproduces rendering artifacts while preserving the scene state and viewpoint. Construction details are provided in Appendix C. For a restored prediction $G_r = f_\theta(\tilde{G}_0)$ and its original target G_0, the joint training objective is

$$\mathcal{L}_{\text{R2C}} = \lambda_{\text{per}}\mathcal{L}_{\text{per}} + \lambda_{\text{col}}\mathcal{L}_{\text{col}} + \lambda_{\text{pix}}\mathcal{L}_{\text{pix}}, \tag{2}$$

where

$$\mathcal{L}_{\text{per}} = \sum_{\ell} \frac{a_\ell}{N_\ell} \left\| \phi_\ell(G_r) - \phi_\ell(G_0) \right\|_1 ,$$
$$\mathcal{L}_{\text{col}} = \frac{1}{C} \left(\left\| \mu(G_r) - \mu(G_0) \right\|_1 + \left\| \sigma(G_r) - \sigma(G_0) \right\|_1 \right) + \frac{\rho}{N} \left\| G_r - G_0 \right\|_1 , \tag{3}$$
$$\mathcal{L}_{\text{pix}} = \frac{1}{N} \sum_{j=1}^{N} \sqrt{(G_r[j] - G_0[j])^2 + \epsilon^2}.$$

Here, ϕ_ℓ denotes frozen VGG16 features extracted after ImageNet normalization, while μ and σ are channel-wise spatial means and standard deviations. $N = HWC$ and N_ℓ count scalar elements in the image and feature map, respectively. C is the number of color channels, and $G[j]$ denotes the j-th scalar image element. The Charbonnier penalty uses $\epsilon = 10^{-3}$. Only the restoration network is updated, while the VLA remains frozen.

4.3 Execution-Trajectory-Conditioned Alignment

Although geometric viewpoint restoration and R2C enable CoRe-VLA to recover a high-quality joint observation familiar to the VLA from training, the additional processing substantially increases observation-to-action latency. Under synchronous control, the robot must remain idle until the next action chunk is available. Asynchronous inference and execution offer a practical way to reduce this idle time, allowing the robot to continue executing its current action chunk while the next is computed. However, since the robot continues moving during inference, the incoming chunk generally cannot be executed from its beginning. Instead, a method is needed to determine where execution should begin within the incoming chunk to avoid motion conflicts. Such method is particularly important under high and variable latency.

To address the challenge, we propose Execution-Trajectory-Conditioned Alignment (ETCA). It treats action-chunk switching as a trajectory-matching problem: it identifies the prefix of the incoming chunk that best matches the motion already executed during inference, allowing execution to continue from the remaining suffix. The matching considers both the overall trajectory and its terminal joint state, accounting for execution history and the robot's current configuration.

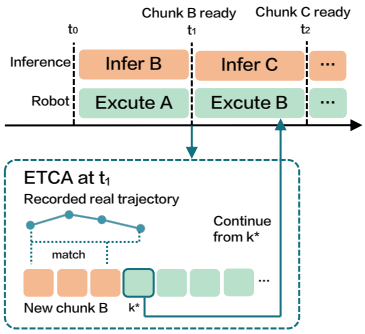


Figure 4: ETCA matches recorded robot motion to a new action chunk and selects where execution resumes.

Specifically, let $\mathbf{Q}^{\text{real}} = [\mathbf{q}_0^r, \ldots, \mathbf{q}_{M-1}^r]$ denote the measured joint states recorded between observation capture and chunk arrival, and $\mathbf{A} = [\mathbf{a}_0, \ldots, \mathbf{a}_{N-1}]$ the incoming action chunk. For each candidate alignment index $k \in \mathcal{K}$, the predicted arm-joint prefix $[\mathbf{a}_0^q, \ldots, \mathbf{a}_k^q]$ is linearly resampled to M states, denoted by $\widetilde{\mathbf{A}}^{(k)}$, to enable comparison with the recorded trajectory. The path and terminal joint-state errors are

$$d_{\text{path}}(k) = \frac{1}{M} \sum_{m=0}^{M-1} \left\| \mathbf{q}_m^r - \widetilde{\mathbf{a}}_m^{(k)} \right\|_2 ,$$
$$d_{\text{end}}(k) = \left\| \mathbf{q}_{M-1}^r - \mathbf{a}_k^q \right\|_2 . \tag{4}$$

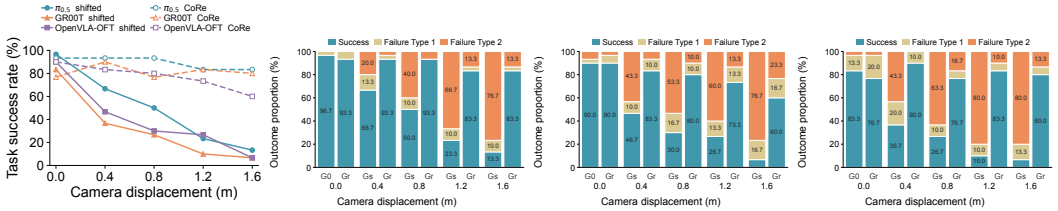
ETCA selects the best-matching prefix by minimizing these errors, with motion-direction consistency and elapsed time providing auxiliary guidance:

$$k^\star = \arg\min_{k \in \mathcal{K}} \left[d_{\text{path}}(k) + \lambda_{\text{end}} d_{\text{end}}(k) + \lambda_{\text{dir}} d_{\text{dir}}(k) + \lambda_{\text{time}} |k - k_{\text{time}}| \right] , \tag{5}$$

where d_{dir} penalizes inconsistent motion directions and k_{time} is an alignment index estimated from elapsed inference time. A match is accepted only if its path and terminal errors satisfy predefined thresholds and it is sufficiently distinct from competing candidates. Execution then continues from the remaining suffix with a short blending transition. Otherwise, the chunk is rejected and inference restarts from the latest observation.

Table 2: Real-robot outcome rates (%) across external-camera shifts. Raw and CoRe denote the shifted view and CoRe-VLA-restored view, respectively. SR denotes success rate, and F2 denotes the proportion of all trials classified as Failure Mode 2 (task-state inconsistency).

Shift (m)	VLA	Metric	T1		T2		T3		T4		T5		Overall	
			Raw	CoRe	Raw	CoRe	Raw	CoRe	Raw	CoRe	Raw	CoRe	Raw	CoRe
0.0	$\pi_{0.5}$	SR	100.0	83.3	83.3	83.3	100.0	100.0	100.0	100.0	100.0	100.0	96.7	93.3
		F2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	OpenVLA-OFT	SR	83.3	83.3	83.3	100.0	100.0	83.3	100.0	100.0	83.3	83.3	90.0	90.0
		F2	0.0	0.0	16.7	0.0	0.0	0.0	0.0	0.0	16.7	16.7	6.7	3.3
	GR00T	SR	83.3	66.7	66.7	66.7	66.7	66.7	100.0	83.3	100.0	100.0	83.3	76.7
		F2	0.0	0.0	16.7	0.0	0.0	16.7	0.0	0.0	0.0	0.0	3.3	3.3
0.4	$\pi_{0.5}$	SR	33.3	83.3	83.3	100.0	33.3	83.3	83.3	100.0	100.0	100.0	66.7	93.3
		F2	33.3	0.0	16.7	0.0	33.3	16.7	16.7	0.0	0.0	0.0	20.0	3.3
	OpenVLA-OFT	SR	66.7	66.7	66.7	100.0	0.0	66.7	33.3	83.3	66.7	100.0	46.7	83.3
		F2	0.0	16.7	33.3	0.0	100.0	0.0	50.0	16.7	33.3	0.0	43.3	6.7
	GR00T	SR	100.0	66.7	33.3	100.0	16.7	100.0	33.3	83.3	0.0	100.0	36.7	90.0
		F2	0.0	0.0	66.7	0.0	83.3	0.0	50.0	0.0	16.7	0.0	43.3	0.0
0.8	$\pi_{0.5}$	SR	16.7	83.3	33.3	100.0	66.7	100.0	83.3	100.0	50.0	83.3	50.0	93.3
		F2	33.3	16.7	66.7	0.0	33.3	0.0	16.7	0.0	50.0	16.7	40.0	6.7
	OpenVLA-OFT	SR	66.7	66.7	33.3	50.0	50.0	83.3	0.0	100.0	0.0	100.0	30.0	80.0
		F2	0.0	0.0	66.7	33.3	50.0	16.7	100.0	0.0	50.0	0.0	53.3	10.0
	GR00T	SR	100.0	83.3	16.7	100.0	16.7	100.0	0.0	0.0	0.0	100.0	26.7	76.7
		F2	0.0	0.0	83.3	0.0	33.3	0.0	100.0	83.3	100.0	0.0	63.3	16.7
1.2	$\pi_{0.5}$	SR	16.7	50.0	16.7	100.0	16.7	83.3	66.7	100.0	0.0	83.3	23.3	83.3
		F2	66.7	50.0	83.3	0.0	83.3	0.0	0.0	0.0	100.0	16.7	66.7	13.3
	OpenVLA-OFT	SR	66.7	83.3	16.7	50.0	50.0	100.0	0.0	100.0	0.0	33.3	26.7	73.3
		F2	0.0	0.0	83.3	16.7	16.7	0.0	100.0	0.0	100.0	50.0	60.0	13.3
	GR00T	SR	50.0	50.0	0.0	100.0	0.0	100.0	0.0	66.7	0.0	100.0	10.0	83.3
		F2	16.7	16.7	100.0	0.0	100.0	0.0	100.0	33.3	83.3	0.0	80.0	10.0
1.6	$\pi_{0.5}$	SR	0.0	50.0	33.3	100.0	0.0	83.3	16.7	100.0	16.7	83.3	13.3	83.3
		F2	100.0	50.0	66.7	0.0	83.3	0.0	50.0	0.0	83.3	16.7	76.7	13.3
	OpenVLA-OFT	SR	33.3	66.7	0.0	50.0	0.0	66.7	0.0	66.7	0.0	50.0	6.7	60.0
		F2	0.0	0.0	100.0	50.0	100.0	16.7	83.3	0.0	100.0	50.0	76.7	23.3
	GR00T	SR	33.3	50.0	0.0	83.3	0.0	100.0	0.0	66.7	0.0	100.0	6.7	80.0
		F2	0.0	16.7	100.0	16.7	100.0	0.0	100.0	33.3	100.0	0.0	80.0	13.3



(a) Success rate (b) $\pi_{0.5}$ (c) OpenVLA-OFT (d) GR00T

Figure 5: Real-robot performance across 5 manipulation tasks under external-camera shifts. (a) Task success rates. (b–d) Outcome proportions for $\pi_{0.5}$, OpenVLA-OFT, and GR00T, respectively. Failure Types 1 and 2 denote local execution failure and task-state inconsistency, respectively. G_0 and G_s denote raw external-camera observations captured at the nominal and shifted positions, respectively, while G_r denotes observations transformed by CoRe-VLA.

5 Experiment

5.1 Experimental Setup

We evaluate CoRe-VLA on the 5 real-robot tasks introduced in Sec. 3, as well as LIBERO (Liu et al., 2023) and LIBERO-Plus (Fei et al., 2026b) in simulation. Three mainstream VLAs are included in our experiments: $\pi_{0.5}$ (Black et al., 2025a), OpenVLA-OFT (Kim et al., 2025b), and GR00T (Bjorck et al., 2025). The VLA parameters remain frozen when applying CoRe-VLA. Our experiments focus on camera position shifts rather than isolated orientation changes, which can be compensated for by 2D image transformations such as rotation when task-relevant objects remain visible. In all experiments, the camera is directed toward the tabletop center after each position shift to help keep task-relevant objects within view.

Task performance is measured by success rate and the two failure categories defined in Sec. 3: local execution failure and task-state inconsistency. ETCA is assessed using mean completion time and mean idle time.

5.2 Evaluation in Real-robot Environment

This experiment tests whether CoRe-VLA improves task success under external-camera shifts and reduces task-state inconsistency in real world environment. We evaluate $\pi_{0.5}$, OpenVLA-OFT, and

GR00T on the 5 real-robot tasks in Table 1, with external-camera displacements of 0.0 to 1.6 m. At each nonzero displacement, three camera positions are randomly selected, and each task is tested under two initial object layouts. We compare raw external-camera observations with observations restored by CoRe-VLA. The comparisons use identical camera positions, object layouts, and control settings, while wrist images are used without modification. Success and the two failure categories defined in Sec. 3 are recorded for each trial.

Table 2 reports per-task and overall success rates and Type 2 failure proportions under both observation conditions. Figure 5 further visualizes these results with success-rate curves and outcome-proportion plots, comparing raw observations with CoRe-VLA across camera displacements. As shown in Fig. 5(a), success rates with raw observations decline sharply as displacement increases, whereas CoRe-VLA maintains substantially higher performance across the evaluated displacement levels. At 1.6 m, CoRe-VLA increases the success rate of $\pi_{0.5}$ from 13.3% to 83.3%, and that of GR00T from 6.7% to 80.0%. These results demonstrate that observation restoration can recover substantial task performance without adapting the VLA itself.

The outcome proportions in Fig. 5(b-d) further show that these improvements are accompanied by a marked reduction in task-state inconsistency (Type 2). At 1.6 m, the proportion of all trials ending in Type 2 failures decreases from 76.7% to 13.3% for $\pi_{0.5}$ and from 80.0% to 13.3% for GR00T. These reductions address the dominant failure pattern identified in Sec. 3 and support the intended effect of CoRe-VLA: restoring the nominal external view for the VLA to prevent cross-view coordination breakdown. Qualitative examples and the corresponding analysis are shown in Fig. 8 in Appendix.

5.3 Evaluation on LIBERO

To evaluate whether the benefits observed in real-robot experiments extend to a broader task set, we conduct experiments on all 100 tasks in LIBERO (Liu et al., 2023). External-camera displacement ranges from 0.0 to 0.9 m in increments of 0.1 m, providing a finer-grained evaluation of viewpoint robustness. Note that camera displacements of 1 m or greater are not evaluated due to the limited size of LIBERO scenes. At each nonzero displacement, ten camera positions are randomly sampled. We compare raw external observations with CoRe-VLA-restored observations, using identical camera poses, initial states, and random seeds. The wrist observation remains unmodified, and the VLA parameters are kept frozen.

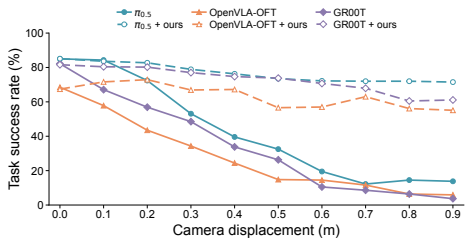


Figure 6: Task success rates on LIBERO-100 under external-camera shifts, without CoRe-VLA (solid) and with CoRe-VLA (dashed).

As shown in Fig. 6, CoRe-VLA substantially mitigates the performance degradation caused by increasing camera displacement for both $\pi_{0.5}$ and OpenVLA-OFT. At 0.9 m, it improves the success rate of $\pi_{0.5}$ from 13.8% to 71.5%, and that of OpenVLA-OFT from 5.9% to 55.1%. Although performance still varies with displacement, the restored observations yield a much smaller overall decline than raw observations. Together with the real-robot results, these findings support the effectiveness of CoRe-VLA across a broader collection of manipulation tasks and different VLA backbones, without additional multi-view data collection or VLA retraining.

Table 3: Success rates (%) on LIBERO-Plus Camera.

Method	Backbone	SR (%) ↑
VLA based methods		
Cross-View AC (Huang et al., 2026b)	$\pi_{0.5}$	87.2
GAM (Han et al., 2026)	DA3-Giant	83.1
Anchor-Align (Dalal et al., 2026)	Prismatic-Qwen2.5-0.5B	**96.3**
AVA-VLA (Xiao et al., 2026)	OpenVLA-OFT	69.4
SRPO (Fei et al., 2026a)	OpenVLA	83.4
$\pi_{0.5}$ (Black et al., 2025a)	$\pi_{0.5}$	72.0
$\pi_{0.5}$ + CoRe-VLA (Ours)	$\pi_{0.5}$	<u>89.1</u>
WAM based methods		
SCVC (Huang et al., 2026a)	Cosmos Policy	76.0
Robust-WAM (Yan et al., 2026)	GE-Act	54.8
Robust-WAM (Yan et al., 2026)	FastWAM	28.7

5.4 Evaluation on LIBERO-Plus

To test robustness under standardized camera shifts and compare CoRe-VLA with existing methods, we further evaluate CoRe-VLA on the camera

subset of LIBERO-Plus (Fei et al., 2026b). In the expriment, The officially released $\pi_{0.5}$ weights are used and remain frozen throughout evaluation. The external view is restored using simulator-provided depth and camera parameters, while the wrist observations remain unchanged.

Table 3 shows that CoRe-VLA improves the success rate of $\pi_{0.5}$ from 72.0% to 89.1%, a gain of 17.1 percentage points without additional multi-view data collection or VLA fine-tuning. Although Anchor-Align reports a higher score, it adapts the VLA, whereas CoRe-VLA leaves the policy parameters unchanged. The reported WAM results also indicate that world modeling alone does not guarantee robust execution under camera shifts.

5.5 Ablation Study

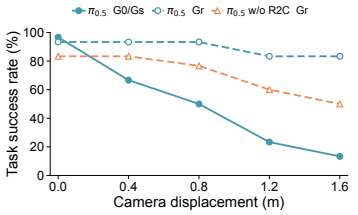


Figure 7: Effect of R2C Restoration on $\pi_{0.5}$ task success under external-camera shifts.

Effect of R2C Restoration on execution outcomes. To isolate the contribution of R2C, we remove it from CoRe-VLA while retaining point-cloud reconstruction, re-rendering, and the frozen $\pi_{0.5}$ backbone. All other experimental settings are identical to those described in Sec. . Figure 7 compares this variant with full CoRe-VLA and direct use of the raw camera observations on the 5 real-robot tasks. Even at zero displacement, success falls from 96.7% with the raw image to 83.3% without R2C, showing that re-rendering itself introduces a harmful appearance mismatch. R2C raises success at this setting to 93.3%, recovering most of the lost performance. As displacement increases, newly exposed but unobserved regions can make the rendered image less complete and less sharp. At 1.6 m, success is 50.0% without R2C but 83.3% with it, a gain of 33.3 percentage points. This improvement is consistent with R2C reducing rendering-induced texture loss, color distortion, and blur, making the restored external view more useful alongside the unchanged wrist view.

Table 4: Effect of ETCA on execution efficiency.

Method	Latency (ms)	Success (%)	Completion (s)	Idle (s)
$\pi_{0.5}$	244	96.7	47.2	22.5
$\pi_{0.5}$ + ETCA	244	96.7	34.4	9.0
$\pi_{0.5}$ + CoRe-VLA w/o ETCA	478	93.3	64.5	39.1
$\pi_{0.5}$ + CoRe-VLA	478	93.3	43.7	13.8

Effect of ETCA on execution time. To assess whether ETCA mitigates the execution-time overhead of viewpoint restoration, four configurations are compared at the nominal camera position across our real-robot benchmark, with two initial layouts and three repetitions per layout. All inference runs on an NVIDIA RTX 5060 Ti (16 GB), with other settings held constant. Configurations without ETCA use synchronous execution. Timing statistics include only matched trials completed successfully by all four configurations.

As shown in Table 4, ETCA reduces CoRe-VLA's mean completion time from 64.5 to 43.7 s (32.2%) and idle time from 39.1 to 13.8 s (64.7%), despite unchanged single-inference latency. It also reduces completion time for standalone $\pi_{0.5}$ by 27.1%, with no observed success-rate decrease in either comparison. These results support ETCA's effectiveness in reducing execution time through asynchronous inference and execution rather than faster individual inferences. We further analyze how the individual loss terms affect R2C Restoration quality in Appendix E.

6 Conclusion and Limitations

This work identifies *cross-view coordination breakdown* in VLAs under camera shifts, which is the major cause of task failure. The proposed CoRe-VLA demonstrates that this coordination breakdown can be mitigated without collecting additional data or fine-tuning the VLA: using point-cloud reconstruction and re-rendering to restore the joint observations familiar to the VLA during training. Although extensive experiments show that CoRe-VLA substaintially improves VLA viewpoint robustness and ETCA can reduce task completion time, there are several limitation in this work. First, CoRe-VLA relies on reasonably accurate camera depth measurements. Additionally, Large viewpoint changes or occlusions can reveal regions absent from the source observation, which neither reprojection nor R2C can reliably recover. Developing viewpoint robust methods that are independent of camera hardware and reliable under large camera displacements remains an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to polish manuscript language, preliminarily screen relevant literature, and discuss the feasibility of the method design. We did not use generative AI to implement CoRe-VLA or deploy the experiments. We have reviewed all AI-assisted material, including every AI-polished sentence, and verified the existence of related work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision*, pp. 696–712. Springer, 2020.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pp. 17–33. Springer, 2022.

Dwip Dalal, Shivansh Patel, Chahit Jain, Jeonghwan Kim, Utkarsh Mishra, Alex Baratian, Hyeonjeong Ha, Heng Ji, Svetlana Lazebnik, and Unnat Jain. Generalizable vla finetuning via representation anchoring and language-action alignment. *arXiv preprint arXiv:2607.13429*, 2026.

Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. Srpo: Self-referential policy optimization for vision-language-action models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6718–6728, 2026a.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026b.

Laura Fink, Darius Rückert, Linus Franke, Joachim Keinert, and Marc Stamminger. Livenvs: Neural view synthesis on live rgb-d streams. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

Jisang Han, Seonghu Jeon, Jaewoo Jung, René Zurbrügg, Honggyu An, Tifanny Portela, Marco Hutter, Marc Pollefeys, Seungryong Kim, and Sunghwan Hong. Geometric action model for robot policy learning. *arXiv preprint arXiv:2606.17046*, 2026.

Hyeongjun Heo, Seungyeon Woo, Sang Min Kim, Junho Kim, Junho Lee, Yonghyeon Lee, and Young Min Kim. Anycamvla: Zero-shot camera adaptation for viewpoint robust vision-language-action models. *arXiv preprint arXiv:2603.05868*, 2026.

Bingqi Huang, Bingchuan Wei, Yingkai Cai, and Zhaokui Wang. Selective cross-view consistency for world action models: Held-out viewpoint robustness without test-time camera information. *arXiv preprint arXiv:2608.21402*, 2026a.

Bingqi Huang, Bingchuan Wei, Xuan Wang, Yingkai Cai, and Zhaokui Wang. Cross-view action consistency for camera-robust vision-language-action policies. *arXiv preprint arXiv:2608.06965*, 2026b.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139–1, 2023.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/748becc400a57c0e31cfe6a2e7951467-Paper-Conference.pdf.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. Vla models are more generalizable than you think: Revisiting physical and spatial modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35025–35035, June 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. volume 36, pp. 44776–44791, 2023.

Yechan Park and HyunJin Kim. Gs-vla: Plug-and-play viewpoint canonicalization for frozen vla policies via gaussian splatting. *arXiv preprint arXiv:2608.19066*, 2026.

Thomas Schöps, Martin R Oswald, Pablo Speciale, Shuoran Yang, and Marc Pollefeys. Real-time view correction for mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2455–2462, 2017.

Haoxuan Wang, Gengyu Zhang, Yan Yan, Yuzhang Shang, Ramana Kompella, and Gaowen Liu. Real-time robot execution with masked action chunking. In *International Conference on Learning Representations*, volume 2026, pp. 120161–120179, 2026.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: Improving vision-language-action models with active visual attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13453–13463, 2026.

Haodong Yan, Junfeng Li, Junjie He, Zhide Zhong, MingMing Yu, Wenxuan Song, Jiaguan Zhu, Yangyang Zheng, Yuqiao Du, Jiadi You, et al. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models. *arXiv preprint arXiv:2608.05903*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Borong Zhang, Jiahao Li, Jiachen Shen, Yuhao Zhang, Yishuai Cai, Lu Liu, Hailu Ji, Yuanpei Chen, Juntao Dai, Jiaming Ji, et al. Vla-arena: An open-source framework for benchmarking vision-language-action models. *arXiv preprint arXiv:2512.22539*, 2025.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, et al. Do world action models generalize better than vlas? a robustness study. *arXiv preprint arXiv:2603.22078*, 2026.

Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics*, 33(4):1–10, 2014.

APPENDIX

This appendix provides the success criteria for the five real-robot tasks (Appendix A) and additional details on the LIBERO benchmarks (Appendix B). Appendix C is reserved for the construction of R2C training pairs, while Appendix D is reserved for analysis of qualitative execution examples. Appendix E presents a preliminary visual ablation of R2C, and Appendix F discusses related asynchronous execution methods

A REAL-ROBOT TASK SUCCESS CRITERIA

Table 5 specifies the success criteria for the tasks defined in Table 1. For sequential tasks, all required steps must be completed in the instructed order.

B INTRODUCTION OF SIMULATION BENCHMARKS

LIBERO (Liu et al., 2023) is a simulated benchmark for language-conditioned robot manipulation, providing task instructions, proprioceptive states, and images from workspace and wrist cameras. It includes suites focused on spatial relationships, object identity, and task goals, while LIBERO-100 contains 100 tasks involving diverse object interactions and motor skills. LIBERO-Plus (Fei et al., 2026b) extends this benchmark with controlled perturbations across seven dimensions to assess model robustness beyond standard task completion. Its Camera subset contains 1,599 instances across the Spatial, Object, Goal, and Long suites, with perturbations to camera distance, position, and orientation.

C CONSTRUCTION OF R2C TRAINING PAIRS

Training pairs are constructed from RGB frames in existing 2D task demonstrations, without depth maps, camera calibration, or additional views. For each original frame G_0, a severity-controlled projective warp creates an image with uncovered regions. Within the valid image region, short gaps are sampled near object contours, followed by sparse clusters of missing pixels; all uncovered and missing pixels are set to black. The warp parameters and missing-pixel masks vary across frames, producing degraded inputs $\tilde{G}_0$ paired with their original targets G_0. This procedure approximates the visible gaps caused by point-cloud re-rendering, rather than synthesizing a geometrically accurate new view.

D QUALITATIVE EXECUTION EXAMPLES

Figure 8 compares executions of Task 1, Task 3, and Task 5 under nominal, shifted, and CoRe-VLA-restored external views. In Task 1, the shifted view causes a simple pick-and-place task to fail even though the relevant objects remain visible, while the nominal and restored-view executions succeed. Task 3 and Task 5 more directly illustrate cross-view coordination breakdown. In Task 3, the robot must open the lid before placing the tape measure in the basket, but the shifted-view execution acts

Table 5: Success criteria for the five real-robot tasks.

Task	Success criterion
T1	The robot grasps the green glue stick and releases it inside the pink cup.
T2	The robot places and releases the yellow tape measure inside the basket, then places and releases the pink cup inside the basket.
T3	The robot first opens the lid so that it no longer covers the basket, then places and releases the yellow tape measure inside the basket.
T4	The robot grasps the green glue stick and releases it on the side of the pink cup farther from the basket, within a $\pm 45^\circ$ sector about the basket-to-cup direction.
T5	The robot grasps the object initially closest to the basket and releases it inside the basket.

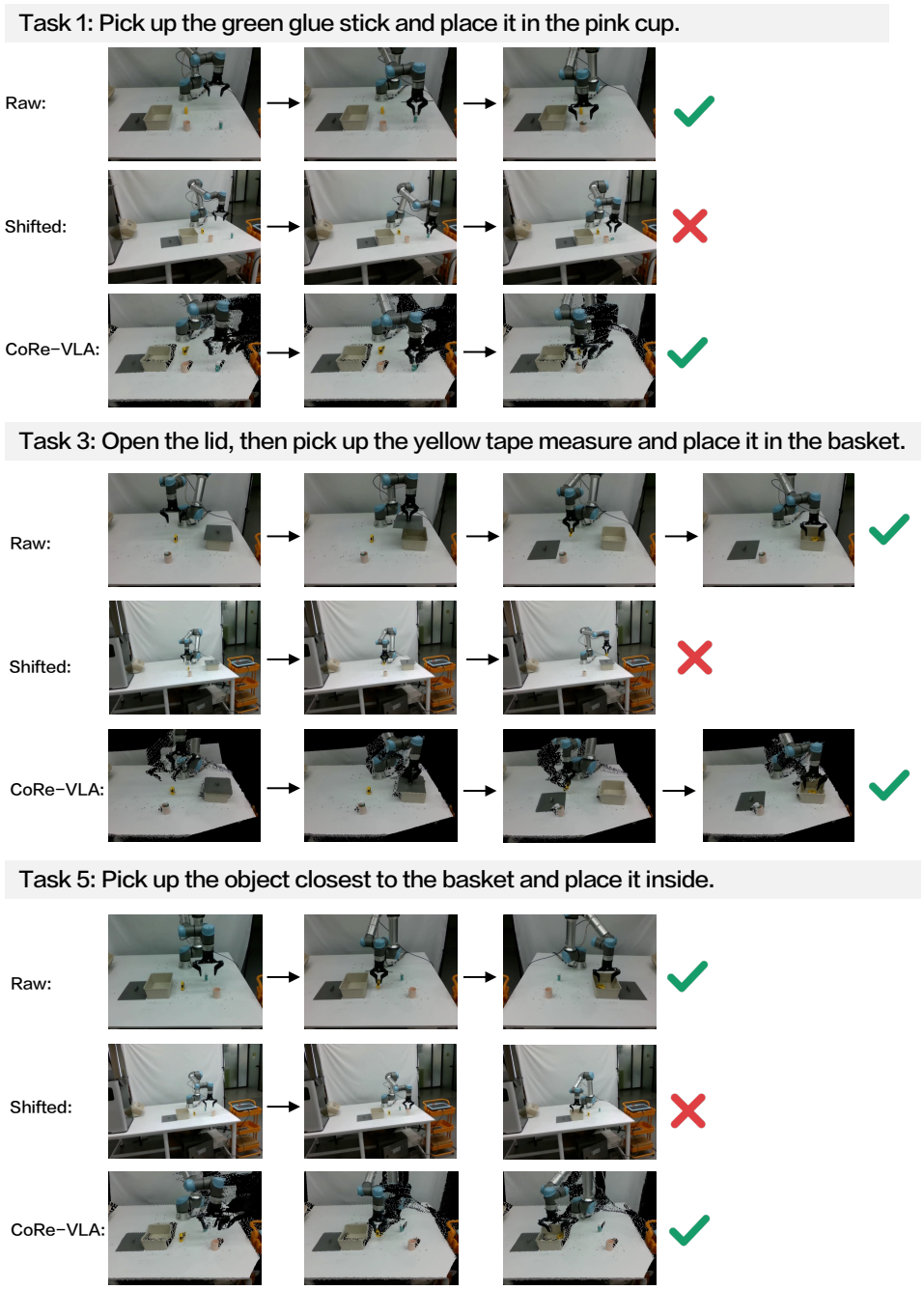


Figure 8: Representative real-robot execution sequences under nominal (Raw), shifted, and CoRe-VLA-restored external views. Arrows indicate action order; check marks and crosses denote success and failure, respectively.

on the locally visible tape measure without maintaining this order. In Task 5, selecting the object closest to the basket requires the global spatial relationships provided by the external view. Under the shifted view, the robot instead selects the cup visible in its wrist view. These examples show that local cues can remain actionable while the global context needed for task-state interpretation and target selection is no longer used effectively. By restoring an external view familiar to the VLA from training, CoRe-VLA helps align wrist-level cues with the intended task order and spatial constraints, despite residual artifacts in the restored images.

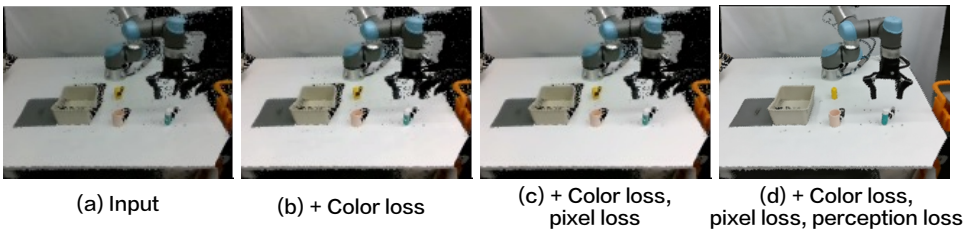


Figure 9: Qualitative ablation of the R2C training losses. From left to right: original image, degraded input, and restorations using color loss, color and pixel losses, and the full objective including perceptual loss.

E Visual Ablation of R2C Restoration

Figure 9 compares restorations from the same degraded input as the loss terms are added. The input contains missing regions around the robot and basket, speckled artifacts, and unclear object boundaries. Color loss alone, or combined with pixel loss, leaves much of this corruption visible. Adding the perceptual loss produces the closest match to the original image in this example, with a more coherent robot appearance and clearer tabletop objects. This visual comparison complements the execution-outcome ablation in Sec. 4.2.

F Comparison with Asynchronous Execution Methods

Asynchronous execution allows the robot to act while the next action chunk is being inferred, but the incoming chunk must be reconciled with motion already performed. Real-Time Chunking (RTC) (Black et al., 2026) addresses this for diffusion- and flow-based policies by freezing committed actions and inpainting the remaining chunk during sampling. REMAC (Wang et al., 2026) learns corrective adjustments through masked action chunking to handle inconsistencies during asynchronous execution. ETCA instead operates on an already predicted chunk: it matches candidate prefixes to the robot's measured joint trajectory and executes the corresponding suffix only when the match passes quality checks. This makes ETCA particularly suited to CoRe-VLA's variable observation-processing latency without changing the frozen VLA or its action-generation procedure.