VIEW-ROBUST VISION-LANGUAGE-ACTION MODEL VIA CAMERA-CENTRIC REPARAMETERIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world robot deployment rarely maintains the training-stage camera setup, where cameras often experience repositioning or remounting depending on actual scenarios. Existing view-robust Vision-Language-Action (VLA) policies tolerate such camera variations only when the camera extrinsics are explicitly provided, making them fragile and hard to use especially when view robustness is critical. We argue that the policy should not be told where the camera is, but rather figure it out by itself. To this end, we introduce Camera-Centric VLA (CamVLA), a new VLA model that decouples manipulation controls from camera geometry by predicting (i) a camera-centric end-effector action expressed in the local camera frame, and (ii) a 6-DoF hand-eye matrix relating cameras to the robot base. A deterministic geometric transformation composes the two predictions into a robot base-frame action. This disentangles *how I should move* in pose-independent camera-centric action generation from *where I am looking from* in camera-perspective geometric grounding. The resulting policy is calibration-free, depth-free, and single-view, requiring only a single monocular RGB image as the visual observation and task instruction at deployment. Evaluations in both simulation and real-world robot data show that CamVLA consistently improves success rates across diverse unseen viewpoints. Code will be released.

1 INTRODUCTION

Vision-Language-Action (VLA) models (Brohan et al., 2022; Kim et al., 2025; Intelligence et al., 2026; Bjorck et al., 2025) have rapidly progressed toward generalist robot policies, leveraging internet-scale vision-language data and diverse robotic demonstrations to ground broad semantic knowledge into directly executable manipulation. Yet despite their semantic competence, state-of-the-art VLAs exhibit a sharp and unexpected brittleness to camera viewpoint shifts.

As illustrated in Figure 1, π_0 (Black et al., 2024) trained on a single canonical perspective achieves a $\sim$65.3% success rate under its training view on RLBench (James et al., 2020), yet collapses to a mere 6.3% under a 15° camera rotation. This failure persists even when the scene remains fully observable and the semantic goal is unchanged. Although large-scale multi-view training could mitigate this, acquiring such data is prohibitively expensive and hard to scale. In practice, real-world robot deployment rarely matches the controlled camera setup of training time: sensors get bumped during operation, mounted on different platforms, hand held by operators, or affixed to mobile bases whose pose drifts continuously. Consequently, without inherent view robustness, VLAs remain tethered to static laboratory setups, failing to generalize to the dynamic and unconstrained configurations of real-world deployment.

This brittleness has a structural origin. Standard VLAs (Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025) rigidly predict actions in the robot base frame from camera-perspective visual observations. However, this base-frame parameterization misaligns action outputs with camera-frame inputs, requiring the network to implicitly resolve the spatial transformation from the camera to the robot base (hand-eye transformation). Without explicit geometric constraints, this hand-eye transformation remains a hidden variable, forcing the policy to memorize coordinate mappings by coupling manipulation control with camera geometry, which easily collapses under minor viewpoint shifts. Recent works converge on a single recipe to fix this by *telling the policy where the camera is*. For instance, OC-VLA (Zhang et al., 2026) bypasses this frame-of-reference gap by re-expressing targets

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.

in calibrated camera coordinates; Jiang et al. (2025) rely on ray embeddings derived from known camera parameters; 4D-VLA (Zhang et al., 2025a) back-projects pixels into the robot frame using known intrinsics and extrinsics; and AnyCamVLA (Heo et al., 2026) synthesizes canonical views given both source and target camera poses. Despite their architectural diversity, all of these methods require *known and accurate camera extrinsics at deployment*, which is precisely the assumption that breaks under hand-held, drifting, or remounted cameras. Existing view-robust VLAs are therefore most fragile in the deployment regimes that motivate view robustness in the first place.

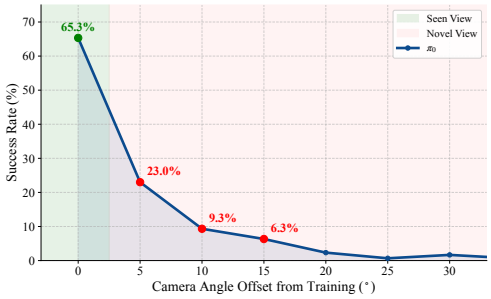


Figure 1: **The Viewpoint Trap in VLAs.** Conventional VLA (*e.g.*, π_0) trained on a single viewpoint exhibits extreme spatial brittleness, where a mere 15° camera shift drops success rates to 6.3%.

In this work, we argue that the policy should not be *told* where the camera is, but rather should *figure it out by itself*. This self-localizing capability finds a natural analogue in human cognition: human vision-guided manipulation operates natively in an egocentric reference frame, while a complementary allocentric system maintains an implicit understanding of head pose relative to the torso (Goodale & Milner, 1992; Crawford et al., 2004). Inspired by this biological factorization, we decouple the VLA policy into two distinct subproblems, both inferable from a single monocular RGB image: *(i) Camera-Centric Action Generation* answers *"How should I move?"* by predicting end-effector actions natively in the local camera coordinate frame, which aligns directly with visual observations, making the mapping inherently independent of the external camera pose. *(ii) Camera-Perspective Geometric Grounding* answers *"Where am I looking from?"* by regressing the 6-DoF hand-eye matrix that relates the observing camera to the robot base, explicitly modeling the relative spatial geometry and isolating viewpoint-dependent variation within a single relative pose.

We instantiate this factorization as CamVLA (see Figure 2), a calibration-free, camera-centric VLA model. From the extracted visual representations, two specialized heads predict the camera-centric action and the hand-eye matrix in parallel, and a deterministic geometric transformation composes them into a robot base-frame action. Parameterizing actions natively in the camera frame aligns them directly with visual observations, avoiding the geometric entanglement of base-frame policies that map conflicting visual flows to base coordinates. By absorbing all viewpoint variability into a single learned 6-DoF pose, CamVLA presents a simpler regression task than memorizing view-dependent coordinate mappings. Crucially, it requires no external camera information at deployment (e.g., calibrated extrinsics, depth sensors, or view synthesis), relying exclusively on the same monocular RGB image and language command that standard VLA already receives. By replacing the assumption of *given* geometry with the discipline of *learned* geometry, CamVLA closes the deployment gap left open by prior view-robust methods, enabling calibration-free manipulation under uncalibrated viewpoint shifts on real hardware.

Our contributions can be summarized as threefold. *First*, we identify that existing view-robust VLA approaches share a common, deployment-fragile assumption of known camera extrinsics, and we argue for a self-localizing alternative that infers camera geometry directly from RGB. *Second*, we propose CamVLA, a calibration-free, depth-free, single-view VLA framework that decouples camera-centric action generation from camera-perspective geometric grounding and recombines them via a deterministic geometric transformation. *Third*, we comprehensively evaluate CamVLA in both simulation and real-world deployment, demonstrating substantial success rate improvements over strong VLA baselines (e.g., π_0 (Black et al., 2024) and GR00T N1.7 (Bjorck et al., 2025)) across diverse unseen camera configurations.

2 Related Work

Vision-Language-Action Models. The pursuit of generalist robot policies has catalyzed the development of VLAs (Zitkovich et al., 2023; Driess et al., 2023; Ahn et al., 2022; Collaboration et al., 2023; Intelligence et al., 2025a), which adapt VLMs (Liu et al., 2023; Li et al., 2023; Zhu et al., 2024;

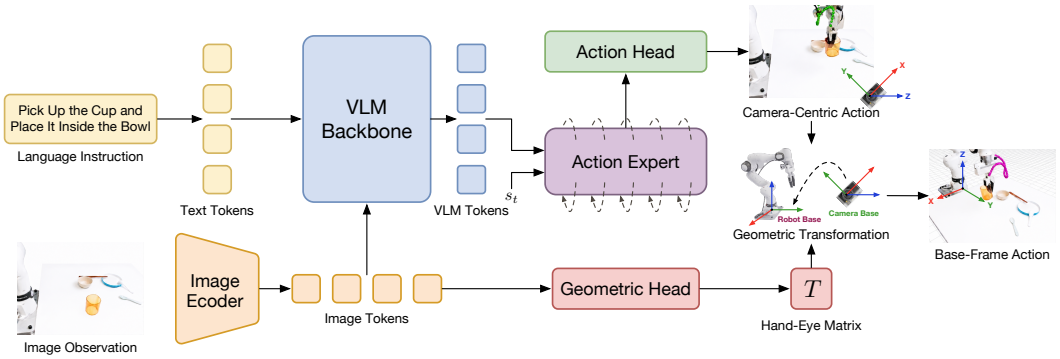


Figure 2: **Overview of the CamVLA Architecture.** Our CamVLA predicts the local camera-centric action and the 6-DoF hand-eye pose in parallel, which are then combined via a deterministic geometric transformation to execute the base-frame action.

Bai et al., 2025) by predicting continuous robot actions via large-scale transformers. Octo (Team et al., 2024) utilizes a Transformer backbone to handle multi-modal observations and a diffusion head (Chi et al., 2025) for robust action generation. Recent flow-matching architectures such as π_0 (Black et al., 2024) and $\pi_{0.5}$ (Intelligence et al., 2025b) have advanced high-frequency continuous control, open-world generalization, and learning from real-world experience. GR00T N1 (Bjorck et al., 2025) employs a dual-system architecture to coordinate high-level reasoning with low-level motor commands for humanoid control. Fast-in-Slow (Chen et al., 2025) further explores the unification of fast manipulation within a slow reasoning system to improve real-time responsiveness. While these models possess strong semantic knowledge, their generalization capabilities remain limited under novel camera viewpoints.

Viewpoint Generalization in Robotics. Achieving viewpoint robustness remains a critical bottleneck for robot policies, whose performance can degrade sharply when the deployment camera pose differs from the training setup (Li et al., 2025). One line of work introduces explicit 3D structure to improve geometric consistency. For example, Perceiver-Actor (Shridhar et al., 2023) builds voxelized RGB-D scene representations, and recent VLA models incorporate 3D features, depth/point-cloud priors, or spatiotemporal 3D representations (Zhen et al., 2024; Jia et al., 2024; Zhang et al., 2025b; Ke et al., 2024; Deng et al., 2025). However, these approaches often introduce computational overhead and depend on depth sensing, multi-view observations, or calibrated camera geometry. Alternative approaches improve robustness via novel-view synthesis or image-level augmentation (Tian et al., 2024; Coholich et al., 2026; Xu et al., 2025), viewpoint selection (Vasudevan et al., 2025), and view-invariant representations or latent actions (Pang et al., 2025; Jeong et al., 2026; Luo et al., 2025; Li et al., 2026; Sun et al., 2026; Abouzeid et al., 2025). Several concurrent works mitigate viewpoint shifts via camera-aware policy conditioning, including grounding actions in the camera space (Zhang et al., 2026), conditioning on camera parameters via ray embeddings (Jiang et al., 2025), back-projecting pixels into the robot base frame using known camera parameters (Zhang et al., 2025a), or synthesizing canonical views based on relative camera poses (Heo et al., 2026). Unlike these camera-aware methods that all heavily rely on known and accurate camera intrinsics or extrinsics at deployment, CamVLA self-predicts the hand-eye matrix directly from monocular RGB, achieving view-robust manipulation without external calibration, depth sensing, novel-view synthesis, or 3D reconstruction.

3 Methodology

3.1 Overview

As shown in Figure 2, CamVLA achieves viewpoint robustness by decoupling the policy into two parallel components: (i) an Action Head that predicts end-effector actions natively in the camera coordinate frame, and (ii) a Geometric Head that regresses the 6-DoF hand-eye matrix to estimate

the camera pose relative to the robot base. A deterministic transformation then composes these two outputs into a base-frame action for execution. This decoupling design factorizes the VLA policy into pose-independent action generation and viewpoint-dependent hand-eye grounding, isolating viewpoint variations from the core visual-action mapping. In the following subsections, we review the standard VLA formulation and detail our decoupling pipeline.

3.2 Standard VLA Action Formulation

At each timestep t, the VLA model receives a visual observation $I_t \in \mathbb{R}^{H\times W\times 3}$, a proprioceptive state $s_t = [p_{b,t}, r_{b,t}]$, and a natural language goal L. Here, $p_{b,t} \in \mathbb{R}^3$ and $r_{b,t} \in \mathbb{R}^3$ denote the base-frame position and axis-angle orientation of the robot end-effector, respectively. Standard VLAs (Kim et al., 2025; Black et al., 2024; Bjorck et al., 2025) predict the base-frame delta action $\Delta A_{b,t} = [\Delta p_{b,t}, \Delta r_{b,t}, g_t]$, where $\Delta p_{b,t}, \Delta r_{b,t} \in \mathbb{R}^3$ denote the translational displacement and axis-angle delta rotation, and $g_t \in [0, 1]$ is the gripper state. The network parameters θ are optimized with the action prediction objective $\mathcal{L}_{\text{act}}$; for regression-style policies, this can be written as $\sum_t \| f_\theta(I_t, s_t, L) - \Delta A_{b,t}\|_2^2$.

3.3 Camera-Centric Action Generation

When f_θ is trained to output $\Delta A_{b,t}$, it learns a difficult cross-frame mapping $F : (I_t, s_t, L) \mapsto \Delta A_{b,t}$ from camera-perspective visual observations to a base-frame action. The bridge between these two frames is the hand-eye matrix T_t, a rigid transform relating the camera and robot coordinate systems (Tsai et al., 1989; Shiu & Ahmad, 1987). Absent from the input, T_t is implicitly entangled in the weights of f_θ. While training on multi-view datasets commonly improves viewpoint generalization (Qian et al., 2025; Cai et al., 2026), a base-frame policy forces the network to map conflicting visual flows to the same base-frame action, leading to geometric entanglement. Consequently, the mapping F collapses once the deployment T_t diverges from the training distribution.

Rather than relying on this cross-frame mapping to absorb T_t implicitly, our approach decouples the visual-to-action mapping from T_t by letting the policy learn a simpler same-frame form in which both visual observations and actions are natively expressed in the local camera frame, where this mapping remains independent of how the camera is mounted. We define the camera-centric delta action as:

$$\Delta A_{c,t} = [\Delta p_{c,t}, \Delta r_{c,t}, g_t], \tag{1}$$

where $\Delta p_{c,t} \in \mathbb{R}^3$ and $\Delta r_{c,t} \in \mathbb{R}^3$ are the relative translation and axis-angle rotation expressed in the camera frame. Because both visual representations and actions share the same egocentric perspective, visual flows are naturally aligned with physical movement in the camera frame. For instance, a leftward visual translation in the image consistently corresponds to a negative displacement along the local X-axis of the camera. By resolving this conflict, this consistent spatial relationship prevents visual-action confusion, helping the policy generalize to unseen viewpoints.

3.4 Camera-Perspective Geometric Grounding

While predicting $\Delta A_{c,t}$ ensures robust visual alignment, robotic arms fundamentally operate based on kinematic models and inverse kinematics defined in their grounded base frame (Siciliano et al., 2009). The hand-eye transformation defines the rigid-body mapping between the observing camera and the robot base coordinate system, which is essential for bridging the visual and physical action spaces. Consequently, we cannot send $\Delta A_{c,t}$ directly to the low-level controller without first applying this transformation.

To bridge this gap, our architecture incorporates a specialized auxiliary Geometric Head that regresses the 6-DoF hand-eye matrix $T_t \in SE(3)$ from visual features. This makes T_t an explicit network output, whereas in the cross-frame baseline, it remains implicitly entangled in the weights. We parameterize T_t with a translation vector $\tau_t \in \mathbb{R}^3$ and an axis-angle rotation vector $\omega_t \in \mathbb{R}^3$, where ω_t is converted to a rotation matrix $R_t \in SO(3)$ during geometric fusion. The network f_θ is therefore redefined to jointly output the camera-centric action $\Delta A_{c,t}$ from the Action Head and the hand-eye matrix T_t from the Geometric Head:

$$(\Delta A_{c,t}, T_t) = f_\theta(I_t, s_t, L). \tag{2}$$

3.5 Deterministic Geometric Transformation

To execute actions on the physical robot, we combine the parallel predictions from the Action Head and the Geometric Head using a deterministic geometric transformation. Specifically, since the relative translation $\Delta p_{c,t}$ and axis-angle rotation $\Delta r_{c,t}$ are free vectors, both transform linearly under the predicted rotation $R_t \in SO(3)$, independently of the translation vector τ_t (Murray et al., 2017):

$$\Delta p_{b,t} = R_t \Delta p_{c,t}, \tag{3}$$

$$\Delta r_{b,t} = R_t \Delta r_{c,t}. \tag{4}$$

The final base-frame delta action is then assembled as $\Delta A_{b,t} = [\Delta p_{b,t}, \Delta r_{b,t}, g_t]$. Despite this mathematical independence, we still regress τ_t to enhance the geometric grounding of the visual backbone and to support potential absolute-action variants. Consequently, test-time drift in τ_t has zero physical impact on execution, confining viewpoint errors exclusively to R_t.

3.6 Optimization

We optimize the model end-to-end with a joint objective:

$$\mathcal{L} = \mathcal{L}_{act} + \lambda \mathcal{L}_{ext}, \tag{5}$$

where $\mathcal{L}_{act}$ inherits the action prediction loss of the underlying baseline (e.g., flow-matching in π_0 (Black et al., 2024)), and the geometric grounding loss $\mathcal{L}_{ext}$ is defined as the mean squared error:

$$\mathcal{L}_{ext} = \sum_t \left(\|\tau_t - \hat{\tau}_t\|_2^2 + \|\omega_t - \hat{\omega}_t\|_2^2 \right), \tag{6}$$

with $\hat{\tau}_t, \hat{\omega}_t$ denoting the ground-truth translation and axis-angle rotation vectors. We set the balancing hyperparameter $\lambda = 0.1$.

4 Experiments

4.1 Architecture and Training

CamVLA is trained in a multi-task manner on 8 NVIDIA H100 80GB GPUs, and we subsequently evaluate its performance across different tasks. We instantiate CamVLA upon foundational VLA architectures, specifically π_0 (Black et al., 2024) and GR00T N1.7 (Bjorck et al., 2025), adhering to their respective training recipes and configurations. Visual inputs are restricted to single third-person monocular RGB images resized to 224×224 pixels. To evaluate robustness to third-person viewpoint changes and prevent the model from relying on the unaffected wrist view to bypass them, we exclude wrist cameras, which also simplifies hardware routing across robot joints. The predicted robot actions are parameterized as delta 6-DoF end-effector poses relative to the current state, where the rotation component is represented as a 3D axis-angle vector. The auxiliary Geometric Head is implemented as a lightweight three-layer Multi-Layer Perceptron (MLP) with GELU activations and a hidden dimension of 1024. It operates on the high-level semantic features extracted from the visual tokens of the backbone, which are aggregated via mean pooling before being fed into the MLP.

4.2 Simulation Experiments

Experimental Setup. We evaluate CamVLA on the RLBench benchmark (James et al., 2020). The simulation is powered by the CoppeliaSim engine, with expert demonstrations generated by the Open Motion Planning Library (OMPL) (Sucan et al., 2012). In simulation, the control frequency is 20 Hz and the model executes the first 5 steps of the predicted action trajectory per inference. To evaluate viewpoint robustness, we utilize the front camera and rotate it around the robot base from $-90°$ to $90°$ at $5°$ intervals to

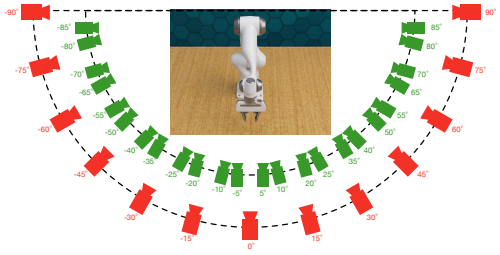


Figure 3: **Simulated camera configuration.** The training set (red cameras) covers discrete viewpoints, while evaluation (green cameras) is conducted on a dense set of unseen viewpoints.

Table 1: **Zero-shot viewpoint generalization on RLBench simulation experiments.** Success rate (%) comparison between VLA baselines and CamVLA across six tasks under unseen viewpoints. All models are trained on the same multi-viewpoint dataset.

Model	Slide Block	Push Buttons	Take Umbrella	Close Laptop	Lamp Off	Put Knife	**Mean**
π_0	18.3	51.5	32.3	57.0	29.8	10.0	33.2
π_0 + CamVLA (Ours)	44.5	72.3	39.2	69.0	58.0	25.3	51.4
GR00T N1.7	27.5	13.5	41.8	47.7	28.2	11.5	28.4
GR00T N1.7 + CamVLA (Ours)	44.7	30.5	50.3	56.0	35.0	14.0	38.4

generate a diverse set of viewpoints as shown in Figure 3. For fair comparisons, all baseline models and CamVLA are trained on the same multi-viewpoint dataset, comprising a discrete subset of viewpoints with 15° intervals ($0^\circ, \pm 15^\circ, \pm 30^\circ, \ldots, \pm 90^\circ$) under identical training budgets. Zero-shot generalization is then evaluated on the remaining unseen viewpoints on the 5° grid. We evaluate CamVLA on six representative manipulation tasks: *slide block to target*, *push buttons*, *take umbrella out of umbrella stand*, *close laptop lid*, *lamp off*, and *put knife on chopping board*. For each task and viewpoint, we collect 100 expert demonstrations for training and 50 episodes for evaluation.

Zero-Shot Viewpoint Generalization. As shown in Table 1, on RLBench simulation experiments, we integrate our CamVLA framework with two foundational VLA architectures, π_0 (Black et al., 2024) and GR00T N1.7 (Bjorck et al., 2025), evaluating zero-shot generalization across six tasks under unseen camera configurations. The reported success rates are averaged across all unseen viewpoints. Notably, even when trained on diverse multi-viewpoint demonstrations, standard end-to-end VLAs struggle to generalize to unseen intermediate viewpoints (achieving only 33.2% for π_0 and 28.4% for GR00T N1.7). This demonstrates that simply increasing viewpoint diversity during training cannot resolve the fundamental base-frame geometric entanglement. In contrast, by explicitly decoupling camera geometry, CamVLA achieves substantial gains of +18.2% (from 33.2% to 51.4%) for π_0 and +10.0% (from 28.4% to 38.4%) for GR00T N1.7. On challenging tasks where vanilla policies severely degrade, such as Slide Block and Lamp Off, CamVLA boosts success rates from 18.3% to 44.5% and 29.8% to 58.0%, respectively. These results confirm that CamVLA generalizes zero-shot to unseen viewpoints, effectively overcoming the discrete viewpoint memorization that plagues standard base-frame VLAs.

4.3 Real-World Experiments

Experimental Setup. Our real-world setup uses a Franka Research 3 robot arm with a parallel gripper. As shown in Figure 4, we use a multi-view camera setup where third-person perspectives are captured by calibrated Intel RealSense D435i cameras. We collect training demonstration data from five different camera perspectives, which provides sufficient viewpoint diversity for VLA learning while avoiding the excessive setup complexity of denser arrays to facilitate easy real-world deployment. To ensure fair comparisons, both standard end-to-end baselines and our CamVLA are trained on the same multi-view datasets under identical training budgets.

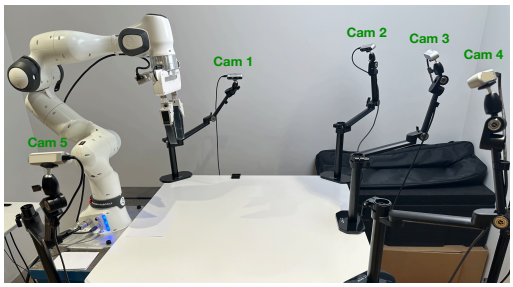


Figure 4: **Real-world experimental setup.** Multi-view camera configuration used to verify viewpoint robustness in physical environments.

For evaluation, we employ three testing cameras (Cam 2, Cam 3, and Cam 4). To evaluate viewpoint robustness, we rotate each testing camera by 5°, 10°, and 15° around the robot base from its default position (0°). We limit the viewpoint offsets up to 15° to cover typical camera perturbations (e.g., sensor drifts) in daily operations, beyond which both the baseline and our model perform poorly, rendering further evaluation impractical. We evaluate on five common household tasks (see Figure 5): *put the basket upright*, *pick up the banana and place to the circular basket*, *push the cabbage near the pineapple*, *wipe the table with the cloth*, and *pick up the cup and place it inside the bowl*. For each task, we collect 100 demonstrations per training viewpoint. During evaluation, we run 20 episodes per task and viewpoint offset on each testing camera. Raw demonstrations are collected at

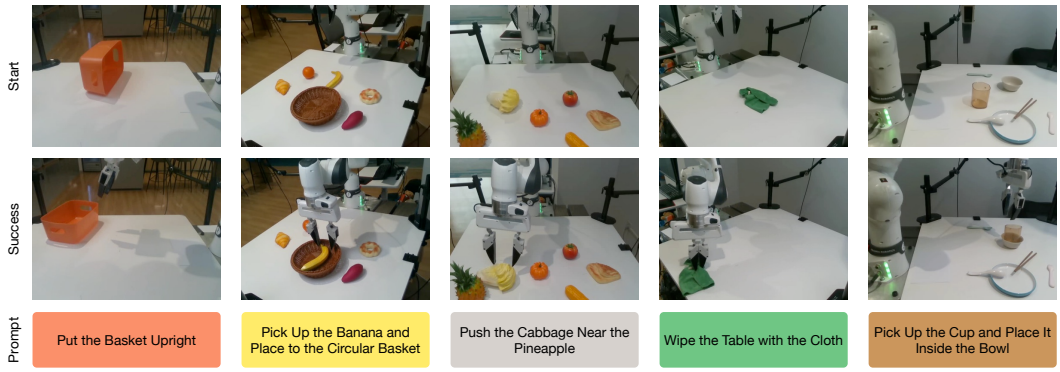


Figure 5: **Real-world evaluation tasks.** Five representative manipulation tasks involving diverse objects and interaction requirements.

Table 2: **Generalization to repositioned cameras on real-world robot experiments.** Success rates (%) across five manipulation tasks under calibration-free viewpoint shifts relative to the canonical view (0°). All policies are evaluated under identical training budgets.

Task	π_0				π_0 + CamVLA (Ours)			
	0°	5°	10°	15°	0°	5°	10°	15°
Basket Upright	75.0	58.3	31.7	16.7	88.3	80.0	68.3	48.3
Pick & Place Banana	53.3	45.0	33.3	0.0	80.0	68.3	50.0	18.3
Push Cabbage	45.0	50.0	33.3	16.7	81.7	70.0	43.3	31.7
Wipe Table	75.0	51.7	41.7	23.3	68.3	71.7	51.7	18.3
Pick & Place Cup	68.3	61.7	56.7	23.3	76.7	50.0	63.3	30.0
Mean	63.3	53.3	39.3	16.0	79.0	68.0	55.3	29.3

Task	GR00T N1.7				GR00T N1.7 + CamVLA (Ours)			
	0°	5°	10°	15°	0°	5°	10°	15°
Basket Upright	95.0	88.3	36.7	0.0	96.7	100.0	73.3	45.0
Pick & Place Banana	48.3	28.3	13.3	0.0	70.0	53.3	30.0	8.3
Push Cabbage	65.0	53.3	65.0	30.0	73.3	81.7	66.7	50.0
Wipe Table	68.3	56.7	35.0	26.7	86.7	63.3	48.3	36.7
Pick & Place Cup	46.7	33.3	28.3	16.7	76.7	63.3	46.7	25.0
Mean	64.7	52.0	35.7	14.7	80.7	72.3	53.0	33.0

30 Hz, while training and deployment are conducted at 10 Hz to ensure stable hardware response, with the model executing the first 20 steps of the predicted trajectory per inference. In both settings, ground-truth hand-eye matrices are used only as training supervision, and the model receives no extrinsic information at deployment.

Generalization to Repositioned Cameras. Table 2 reports the results of our real-world robot experiments, evaluating the generalization performance of CamVLA under repositioned cameras with calibration-free viewpoint shifts. The reported success rates are averaged across all three cameras under the corresponding offset. We observe that under the canonical viewpoint (0°), CamVLA substantially outperforms the standard baselines, achieving average success rates of 79.0% (vs. 63.3% for π_0) and 80.7% (vs. 64.7% for GR00T N1.7). Under the extreme 15° offset, while the baseline success rates collapse to 16.0% (π_0) and 14.7% (GR00T), CamVLA maintains robust performance at 29.3% and 33.0%, respectively, demonstrating exceptional zero-shot generalization to uncalibrated viewpoint shifts. This is enabled by our CamVLA framework, which predicts camera-centric actions and utilizes the auxiliary Geometric Head to dynamically calibrate action outputs from monocular inputs during inference, highlighting the practicality of CamVLA in unstructured settings where camera positions may shift.

To evaluate the accuracy of the Geometric Head in physical environments, we report the hand-eye errors across viewpoints in Table 3. We measure these errors against the ground-truth extrinsics from hand-eye calibration, showing that CamVLA maintains stable geometric grounding even under

Table 3: **Real-world hand-eye matrix estimation errors.** Translation and rotation errors under different viewpoint offsets.

Metric	0°	5°	10°	15°
Trans. (cm)	1.35	2.12	7.91	27.16
Rot. (°)	2.49	4.73	5.98	9.39

Table 4: **Computational Efficiency.** Parameter count, FLOPs, and inference time on RTX 4090.

Model	Params (M)	FLOPs (G)	Inference (ms)
π_0	3238.1	660.9	61
CamVLA	3244.4	661.9	62

Table 5: **Generalization and hand-eye error across training distributions.** Success rates (%) and pose estimation errors under different training intervals. CamVLA† uses ground-truth extrinsics.

Interval	Success Rate (%)			Estimation Error	
	π_0	CamVLA	CamVLA†	Trans. (cm)	Rot. (°)
15°	33.2	51.4	52.3	4.69	1.41
30°	25.5	34.0	40.0	19.71	4.77
45°	16.8	21.2	26.3	34.83	8.28

rotated test viewpoints. Despite the large errors at $15°$, CamVLA outperforms the baseline because our relative-action formulation physically isolates translation errors, while the rotation error (under $10°$) remains within the tolerance of the closed-loop policy (as detailed in Sec. B of Supp.).

Computational Efficiency. We report the parameter count, FLOPs, and inference time of CamVLA in Table 4. To evaluate its real-world deployment feasibility, we measure the inference speed on a single consumer-grade NVIDIA RTX 4090 GPU. As shown, our auxiliary Geometric Head introduces only an extra 6.30 M parameters (0.19% overhead), 1.0 G FLOPs (0.15% overhead), and increases inference latency by only 1 ms (62 ms vs. 61 ms). With a 20-step trajectory executed at 10 Hz, the 62 ms inference easily runs in parallel with robot motion, presenting no bottleneck for real-time deployment.

4.4 Ablation Studies

We ablate key design choices of CamVLA on RLBench (James et al., 2020) with π_0 (Black et al., 2024) as the baseline model, reporting the average success rate across unseen viewpoints for all evaluated tasks.

Density of Training Viewpoints. We examine how the density of camera viewpoints in the training data affects generalization. We compare three training configurations with viewpoint sampling intervals of 15°, 30°, and 45°. As shown in Table 5, denser viewpoint sampling leads to improved success rates under novel perspectives. As training viewpoints become sparser (30° and 45° intervals), the performance of all configurations inevitably degrades. Nevertheless, CamVLA consistently maintains a much higher unseen success rate (e.g., 34.0% under 30° compared to 25.5% of the baseline). The small performance gap between our method and GT under the $15°$ interval (51.4% vs. 52.3%) confirms the high accuracy of our self-predicted extrinsics ($1.41°$ rotation error). Even under the extremely sparse $45°$ configuration, where self-localization errors increase ($8.28°$), CamVLA still achieves a 21.2% success rate, which is close to the GT performance of 26.3% and outperforms the baseline by 4.4%. These results demonstrate that our method is highly robust to self-localization errors. Notably, even the configuration with ground-truth extrinsics (CamVLA†) suffers similar degradation, confirming that this decline is primarily driven by visual representation shifts under unseen perspectives rather than hand-eye regression errors. Furthermore, while CamVLA operates without calibration, our framework is compatible with incorporating online hand-eye calibration systems to replace predicted extrinsics for better performance.

Viewpoint Intervals. To provide a more granular understanding of the local generalization capabilities, we present the detailed relative drop for each individual unseen viewpoint interval in Table 6. We define the *Relative Drop (%)* as $\text{Drop} = (\max(S_{\theta_1}, S_{\theta_2}) - S_{\text{unseen}}) / \max(S_{\theta_1}, S_{\theta_2})$, where S_{θ_1} and S_{θ_2} denote the success rates at the two adjacent training viewpoints, and S_{unseen} represents the success rate at the unseen viewpoint. CamVLA achieves an average relative drop of 4.1%, which

Table 6: **Relative performance drop across individual viewpoint intervals.** We compare the unseen success rates and their relative drops from adjacent seen viewpoints across unseen intervals.

Viewpoint Interval	π_0			CamVLA		
	Seen (%)	Unseen (%)	Drop (%)	Seen (%)	Unseen (%)	Drop (%)
$[0°, 15°]$	42.0	32.8	21.8	70.0	58.7	16.2
$[15°, 30°]$	42.0	44.0	-4.8	70.0	61.5	12.1
$[30°, 45°]$	37.7	29.8	20.8	57.7	57.2	0.9
$[45°, 60°]$	35.7	37.0	-3.7	50.0	51.5	-3.0
$[60°, 75°]$	35.0	30.0	14.3	43.0	43.8	-1.9
$[75°, 90°]$	26.7	25.3	5.0	35.7	35.7	0.0
Average	36.5	33.2	8.9	54.4	51.4	4.1

Table 7: **State and action space ablations.** Comparison of various proprioceptive state representations, action outputs, and calibration requirements.

Model Variant	State Input	Action Output	Hand-Eye Source	Calibration-Free?	Success Rate (%)
π_0	Base	Base	None	✓	33.2
CamVLA‡	Camera	Camera	Ground-Truth	✗	51.9
CamVLA*	Camera	Camera	Self-Predicted	✗	51.7
CamVLA†	Base	Camera	Ground-Truth	✗	52.3
CamVLA	Base	Camera	Self-Predicted	✓	51.4

outperforms the baseline (8.9%), showing exceptional viewpoint invariance. In the viewpoint interval of $[30°, 45°]$, where the baseline suffers a severe performance drop of 20.8%, ours maintains an exceptionally stable drop of only 0.9%, showing robust invariance to viewpoint shifts.

State and Action Representations. As shown in Table 7, we evaluate the impact of different action spaces, proprioceptive states, and calibration requirements. Shifting the action output from the robot base frame to the camera frame yields a significant performance gain, boosting the success rate from 33.2% (π_0) to 51.4% (CamVLA). Using self-predicted instead of ground-truth extrinsics leads to only a marginal performance drop (e.g., 0.9% gap between CamVLA† and CamVLA, and 0.2% gap between CamVLA‡ and CamVLA*), validating the high accuracy of our auxiliary Geometric Head. Although utilizing the base-frame proprioceptive state instead of the camera-frame state (comparing CamVLA with CamVLA*) incurs a 0.3% performance drop, it represents a key design choice that completely eliminates the need for camera calibration in the state observation phase, enabling true calibration-free deployment. This design crucially bypasses the complex hand-eye calibration procedures traditionally required before every execution, offering substantial practical utility for swift plug-and-play robot deployments in varied environments.

5 Conclusion and Limitations

In this work, we present CamVLA, a test-time calibration-free VLA model that achieves viewpoint robustness without external camera information. By decoupling the policy into an Action Head that predicts the camera-centric action and a Geometric Head that regresses the hand-eye matrix from a single monocular RGB image, CamVLA isolates camera-centric action generation from camera-perspective geometric grounding, composing their outputs into base-frame actions through a deterministic transformation. Experiments in simulation and on real hardware demonstrate substantial gains over strong VLA baselines under unseen camera configurations. By replacing given geometry with learned geometry, CamVLA brings VLA policies one step closer to robust and calibration-free manipulation under unseen camera viewpoints in unstructured environments.

While CamVLA achieves viewpoint robustness for the third-person perspective, it only utilizes a single third-person camera and does not consider viewpoint perturbations of the wrist-mounted camera in multi-camera systems. Furthermore, although robust to typical shifts, the model struggles under extreme viewpoint changes and high-precision tasks due to out-of-distribution visual features and hand-eye regression errors. In future work, we plan to improve geometric regression robustness under large viewpoint shifts and investigate wrist-mounted camera perturbations.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing. We have not used generative AI tools for generating synthetic data sets, design or provide feedback on research methodology or experiments, and interpret result. Developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, cleaning and reformatting dataset, and supporting qualitative and thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. Code generated by LLM was verified and tested for correctness by authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ali Abouzeid, Malak Mansour, Qinbo Sun, Zezhou Sun, and Dezhen Song. GeoAware-VLA: Implicit geometry aware vision-language-action model. *arXiv preprint arXiv:2509.14117*, 2025.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as I can, not as I say: Grounding language in robotic affordances. In *CoRL*, 2022.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *RSS*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *RSS*, 2022.

Boyang Cai, Qiwei Liang, Jiawei Li, Shihang Weng, Zhaoxin Zhang, Tao Lin, Xiangyu Chen, Wenjie Zhang, Jiaqi Mao, Weisheng Xu, et al. Beyond viewpoint generalization: What multi-view demonstrations offer and how to synthesize them for robot manipulation? *arXiv preprint arXiv:2603.26757*, 2026.

Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953*, 2025.

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*, 44(10-11):1684–1704, 2025.

Jeremiah Coholich, Justin Wit, Robert Azarcon, and Zsolt Kira. Sim2real image translation enables viewpoint-robust policies from fixed-camera datasets. *arXiv preprint arXiv:2601.09605*, 2026.

OX-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 1 (2), 2023.

J Douglas Crawford, W Pieter Medendorp, and Jonathan J Marotta. Spatial transformations for eye–hand coordination. *Journal of neurophysiology*, 2004.

Shengliang Deng, Mi Yan, Yixin Zheng, Jiayi Su, Wenhao Zhang, Xiaoguang Zhao, Heming Cui, Zhizheng Zhang, and He Wang. StereoVLA: Enhancing vision-language-action models with stereo vision. *arXiv preprint arXiv:2512.21970*, 2025.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. In *ICML*, 2023.

Melvyn A Goodale and A David Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992.

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.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025a.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *CoRL*, 2025b.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi^*_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. 2026.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Youngjoon Jeong, Junha Chun, and Taesup Kim. Learning to act robustly with view-invariant latent actions. *arXiv preprint arXiv:2601.02994*, 2026.

Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3D foundation policy: Lifting 2D large-scale pretrained models for robust 3D robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R Walter. Do you know where your camera is? view-invariant policy learning with camera conditioning. *arXiv preprint arXiv:2510.02268*, 2025.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy diffusion with 3D scene representations. In *CoRL*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *CoRL*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *RSS*, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742, 2023.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. VLA models are more generalizable than you think: Revisiting physical and spatial modeling. *arXiv preprint arXiv:2512.02902*, 2025.

Zheng Li, Pei Qu, Yufei Jia, Shihui Zhou, Haizhou Ge, Jiahang Cao, Jinni Zhou, Guyue Zhou, and Jun Ma. Manivid-3D: Generalizable view-invariant reinforcement learning for robotic manipulation via disentangled 3D representations. *IEEE Robotics and Automation Letters*, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36: 34892–34916, 2023.

Mi Luo, Zihui Xue, Alex Dimakis, and Kristen Grauman. Viewpoint rosetta stone: Unlocking unpaired ego-exo videos for view-invariant representation learning. In *CVPR*, pp. 15802–15812, 2025.

Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. 2017.

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *ICLR*, 2025.

Quanhao Qian, Guoyang Zhao, Gongjie Zhang, Jiuniu Wang, Ran Xu, Junlong Gao, and Deli Zhao. GP3: A 3D geometry-aware policy with multi-view images for robotic manipulation. In *ICRA*, 2025.

Yiu Cheung Shiu and Shaheen Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax= xb. 1987.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-Actor: A multi-task transformer for robotic manipulation. In *CoRL*, pp. 785–799, 2023.

Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. 2009.

Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

Josh Qixuan Sun, Huaiyuan Weng, Xiaoying Xing, Chul Min Yeum, and Mark Crowley. View invariant learning for vision-language navigation in continuous environments. *IEEE Robotics and Automation Letters*, 2026.

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. In *RSS*, 2024.

Stephen Tian, Blake Wulfe, Kyle Sargent, Katherine Liu, Sergey Zakharov, Vitor Guizilini, and Jiajun Wu. View-invariant policy learning via zero-shot novel view synthesis. In *CoRL*, 2024.

Roger Y Tsai, Reimar K Lenz, et al. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *IEEE Transactions on robotics and automation*, 5(3):345–358, 1989.

Sreevishakh Vasudevan, Som Sagar, and Ransalu Senanayake. Viewpoint-agnostic manipulation policies with strategic vantage selection. *arXiv preprint arXiv:2506.12261*, 2025.

Yuan Xu, Jiabing Yang, Xiaofeng Wang, Yixiang Chen, Zheng Zhu, Bowen Fang, Guan Huang, Xinze Chen, Yun Ye, Qiang Zhang, et al. EgoDemoGen: Novel egocentric demonstration generation enables viewpoint-robust manipulation. *arXiv preprint arXiv:2509.22578*, 2025.

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, et al. 4D-VLA: Spatiotemporal vision-language-action pretraining with cross-scene calibration. *arXiv preprint arXiv:2506.22242*, 2025a.

Tianyi Zhang, Haonan Duan, Haoran Hao, Yu Qiao, Jifeng Dai, and Zhi Hou. Grounding actions in camera space: Observation-centric vision-language-action policy. In *AAAI*, volume 40, pp. 18782–18790, 2026.

Zhengshen Zhang, Hao Li, Yalun Dai, Zhengbang Zhu, Lei Zhou, Chenchen Liu, Dong Wang, Francis EH Tay, Sijin Chen, Ziwei Liu, et al. From spatial to actions: Grounding vision-language-action model in spatial foundation priors. *arXiv preprint arXiv:2510.17439*, 2025b.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3D vision-language-action generative world model. In *ICML*, 2024.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, volume 2024, pp. 18378–18394, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pp. 2165–2183, 2023.

This supplementary material covers the following details:

- Additional Implementation details (Sec. A).
- Additional Mathematical derivation (Sec. B).
- Additional ablation studies (Sec. C).
- Additional visualization results (Sec. D).

A Additional Implementation Details

Ground-Truth Extrinsics and Calibration Scope. In simulation, ground-truth camera extrinsics are directly queried from the CoppeliaSim simulator API. In real-world experiments, ground-truth extrinsics for both training viewpoints and perturbed testing viewpoints are obtained offline using a multi-marker ArUco calibration board placed on the workspace via standard PnP pose estimation. We explicitly clarify that this calibration is strictly an offline, one-time procedure used only for training data collection and evaluation measurement. At deployment, CamVLA operates completely calibration-free, inferring both actions and camera geometry directly from a single monocular RGB stream without requiring markers, calibration boards, or known extrinsics.

B Additional Mathematical Derivation

We provide the derivation showing why the execution of camera-centric delta actions is independent of the hand-eye translation vector. Let the hand-eye transform $T_t \in SE(3)$ from the camera frame to the robot base frame be parameterized by a translation vector $\tau_t \in \mathbb{R}^3$ and an axis-angle rotation vector $\omega_t \in \mathbb{R}^3$. We convert ω_t to a rotation matrix $R_t \in SO(3)$ (e.g., via Rodrigues' rotation formula) to construct the homogeneous transformation matrix:

$$T_t = \begin{bmatrix} R_t & \tau_t \\ 0 & 1 \end{bmatrix}, \qquad R_t \in SO(3),\ \tau_t \in \mathbb{R}^3. \tag{7}$$

For two end-effector positions expressed in the camera frame, $p_{c,t}^{(0)}$ and $p_{c,t}^{(1)}$ (denoting the initial and target positions of a delta action, respectively), their corresponding base-frame positions are

$$p_{b,t}^{(i)} = R_t p_{c,t}^{(i)} + \tau_t, \qquad i \in \{0, 1\}. \tag{8}$$

The base-frame relative translation is therefore

$$\begin{aligned} \Delta p_{b,t} &= p_{b,t}^{(1)} - p_{b,t}^{(0)} \\ &= \left(R_t p_{c,t}^{(1)} + \tau_t\right) - \left(R_t p_{c,t}^{(0)} + \tau_t\right) \\ &= R_t \left(p_{c,t}^{(1)} - p_{c,t}^{(0)}\right) = R_t \Delta p_{c,t}. \end{aligned} \tag{9}$$

Thus, τ_t cancels exactly for relative translations. Delta rotations are also independent of the hand-eye translation vector. Let $Q_{c,t}^{(0)}, Q_{c,t}^{(1)} \in SO(3)$ denote the two end-effector orientation matrices in the camera frame, with $Q_{b,t}^{(i)} = R_t Q_{c,t}^{(i)}$ in the base frame. The relative rotation in the base frame satisfies

$$\begin{aligned} \Delta Q_{b,t} &= Q_{b,t}^{(1)} \left(Q_{b,t}^{(0)}\right)^\top \\ &= R_t Q_{c,t}^{(1)} \left(Q_{c,t}^{(0)}\right)^\top R_t^\top = R_t \Delta Q_{c,t} R_t^\top. \end{aligned} \tag{10}$$

Using the equivariance of the matrix logarithm under rotation conjugation,

$$[\Delta r_{b,t}]_\times = \log(\Delta Q_{b,t}) = R_t \log(\Delta Q_{c,t}) R_t^\top = [R_t \Delta r_{c,t}]_\times, \tag{11}$$

which gives $\Delta r_{b,t} = R_t \Delta r_{c,t}$. Consequently, both components of the executed delta action depend only on R_t:

$$\Delta A_{b,t} = [R_t \Delta p_{c,t},\ R_t \Delta r_{c,t},\ g_t]. \tag{12}$$

C Additional Ablation Studies

Extrinsic Noise Robustness. To evaluate the robustness of CamVLA to hand-eye matrix estimation errors, we conduct an ablation study with artificial rotation noise during evaluation. At test time, we systematically inject random rotation noise into the ground-truth hand-eye rotation matrix across a wide range of noise levels, which consists of a 0° reference, 1°–20° in 1° increments, and 25°–45° in 5° increments around a random 3D unit axis. To closely simulate the dynamic prediction fluctuations of the geometric head, the rotation noise is independently resampled at each planning step rather than applying a static bias across the entire episode. We explicitly omit translation noise because, as derived above, delta action execution is independent of the hand-eye translation vector under our relative-action formulation. As shown in Figure 6, the policy is highly robust to small rotation perturbations: the success rate decreases only moderately from the 64.0% no-noise reference to 63.3% at 1° and 58.7% at 5°. Notably, even under injected rotation noise magnitudes of up to 12°, CamVLA still outperforms or remains highly comparable to the noise-free baseline π_0 (36.0%). Crucially, our geometric head predicts hand-eye rotation with a mean error of less than 2° (specifically 1.41° as shown in the ablation table of the main text), a high-precision regime where CamVLA suffers almost no performance degradation. This experiment validates the feasibility of our calibration-free framework, demonstrating that while closed-loop VLA policies naturally possess an inherent tolerance to small execution deviations, our decoupled design (which parallelly predicts camera-centric actions and hand-eye poses) successfully extends this feedback robustness to encompass tolerance against hand-eye rotation errors as well, explaining why CamVLA achieves high success rates despite using self-predicted, imperfect extrinsics.

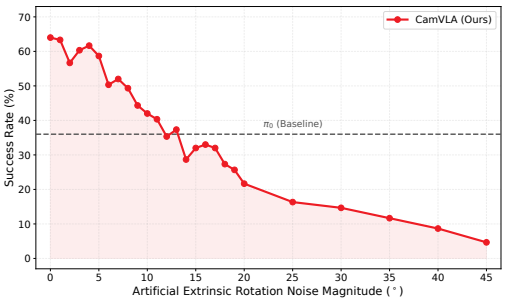


Figure 6: **Robustness under artificial extrinsic rotation noise.** Success rate (%) under varying levels of random rotation noise (ranging from 0° to 45°) applied to the ground-truth hand-eye rotation matrix during execution.

Hand-Eye Pose Representation. We ablate the necessity of predicting the translation component of the hand-eye matrix. Under our relative-action formulation, policy execution depends only on the rotation component. To verify this, we compare the default 6-DoF configuration against a rotation-only variant. As shown in Table 8, additionally predicting translation imposes no extra optimization burden, maintaining rotation accuracy (1.4° vs. 1.6°) and task success (51.4% vs. 51.3%). We therefore retain full 6-DoF pose regression to enhance framework versatility and support potential absolute-action policy variants at negligible parameter cost.

Table 8: **Ablation on hand-eye pose representation.** Comparison of success rates and geometric errors between predicting full 6-DoF extrinsics and regressing rotation only.

Pose Representation	Success (%)	Trans. (cm)	Rot. (°)
Full 6-DoF	51.4	4.7	1.4
Rotation Only	51.3	-	1.6

Visual Feature Sources. We investigate how the visual feature source and gradient propagation affect hand-eye matrix prediction. As shown in Table 9, using features from the Image Encoder yields the most precise geometric localization, with only 4.7 cm translation error and 1.4° rotation error, and achieves a 51.4% unseen success rate. Extracting features from the deeper VLM Backbone slightly improves the task success rate to 53.5%, but substantially worsens geometric estimation (14.7 cm and 3.7°), indicating that high-level semantic features can support action generation while being less spatially precise for camera pose regression. Although the VLM Backbone configuration achieves a slightly higher success rate, we select the Image Encoder as our default configuration. The primary reason is that the VLM Backbone overfits to the discrete training viewpoints, producing geometric discontinuities and large localization spikes at unseen intermediate angles. Such geometric instability poses significant risks during physical robot execution, where smooth and consistent hand-eye estimation is crucial for safety. By contrast, the Image Encoder configuration offers stable and continuous spatial grounding across the entire viewpoint spectrum,

Table 9: **Ablation on the visual feature source for hand-eye matrix prediction.** Comparison of success rates and geometric errors using different visual features.

Feature Source	Success (%)	Trans. (cm)	Rot. (°)
Image Encoder	51.4	4.7	1.4
Image Encoder (Detach)	42.6	11.7	13.2
VLM Backbone	53.5	14.7	3.7
VLM Backbone (Detach)	20.9	45.0	36.0

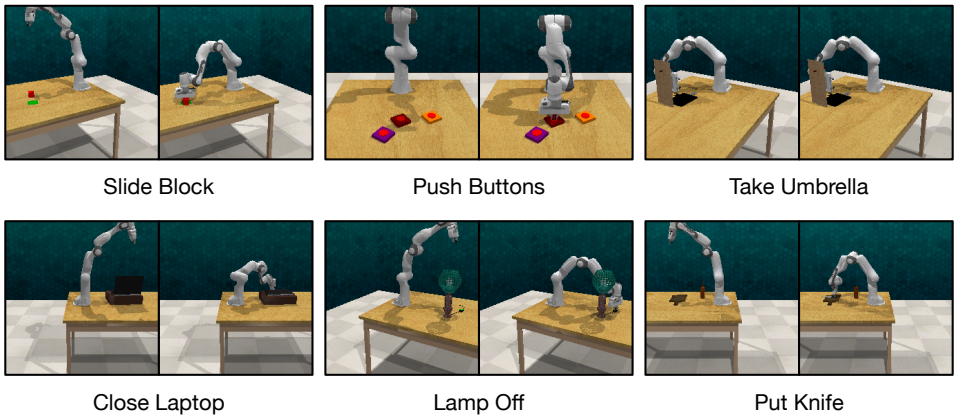


Figure 7: **Simulation tasks on RLBench benchmark.** We evaluate CamVLA across a diverse set of manipulation tasks, requiring both high-level semantic understanding and precise low-level control.

making it a more reliable choice for real-world deployment. Detaching gradients consistently hurts both localization and control. Specifically, Image Encoder detachment reduces success from 51.4% to 42.6%, while VLM Backbone detachment collapses the success rate to 20.9% with severe pose errors (45.0 cm and 36.0°). These results suggest that the Geometric Head benefits from end-to-end feature adaptation, and we therefore use Image Encoder features without detachment as the default configuration for its superior spatial grounding stability.

D Additional Visualization Results

Visualization of Setup and Tasks. Figure 7 provides a visual overview of the 10 RLBench simulation tasks used in our evaluation. Figure 8 visualizes the visual range of both training and testing viewpoints in RLBench. Figure 9 provides a detailed illustration of the camera placement around the robot base for the training viewpoint distribution (ablation study shown in Table 5). Figure 10 illustrates the five training camera viewpoints (Cam 1–Cam 5) used for demonstration collection in our real-world hardware setup. Figure 11 further visualizes the unseen testing viewpoints, where Cam 2, Cam 3, and Cam 4 are perturbed by horizontal rotations of 5°, 10°, and 15° relative to their canonical 0° training position.

Qualitative Comparison. Figure 12 and Figure 13 provide qualitative comparisons of execution trajectories under unseen simulation viewpoints and repositioned real-world camera viewpoints, respectively. We compare the execution trajectories of the baseline π_0 and our CamVLA under unseen viewpoints. CamVLA maintains target-directed behavior and completes the tasks more robustly than the baseline π_0. Additionally, Figure 14 visualizes the real-world deployment under continuously moving and hand-held cameras, echoing the drifting or hand-held camera scenarios and demonstrating the robust viewpoint generalization capability of CamVLA.

Failure Case Analysis. Figure 15 visualizes representative failure cases of both π_0 and CamVLA. These failures are primarily caused by target objects located at the boundary of the camera's field of view, actions exceeding the physical workspace of the robot arm, or visual self-occlusion. Notably, even in these failed cases, CamVLA's execution trajectories still exhibit more target-directed and reasonable behaviors compared to the baseline π_0.

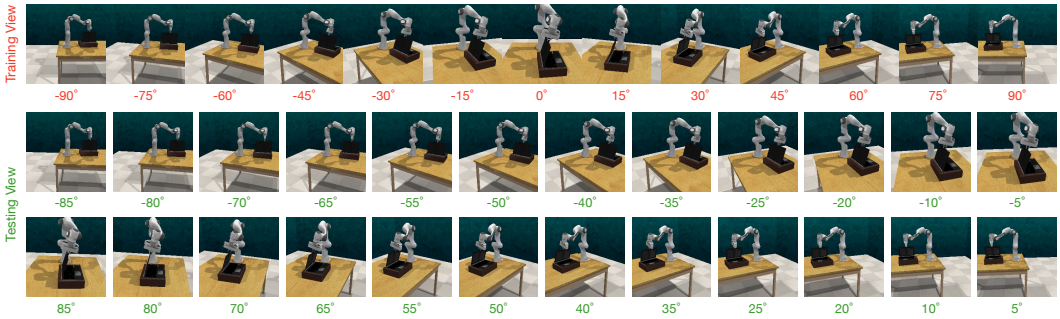


Figure 8: **Visualization of training and testing viewpoints on RLBench.** The training set consists of views sampled at 15° intervals (top row), while the testing set covers a dense range of unseen viewpoints (middle and bottom rows) to evaluate the zero-shot generalization capability of CamVLA.

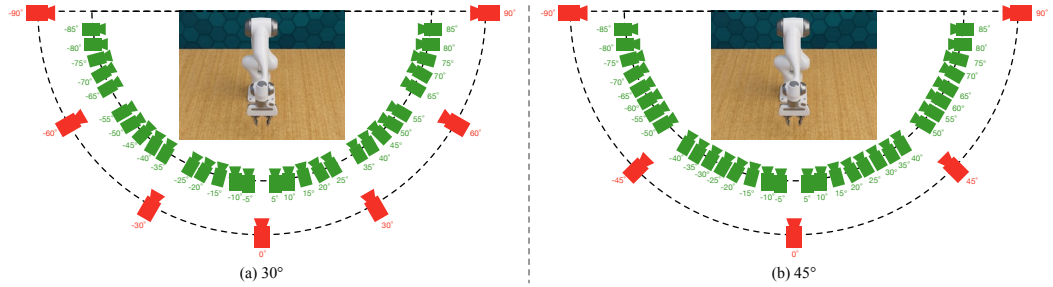


Figure 9: **Detailed range of camera viewpoints in simulation.** (a) Training viewpoints sampled at 30° intervals and (b) training viewpoints sampled at 45° intervals. Red and green cameras represent training and unseen testing viewpoints, respectively.

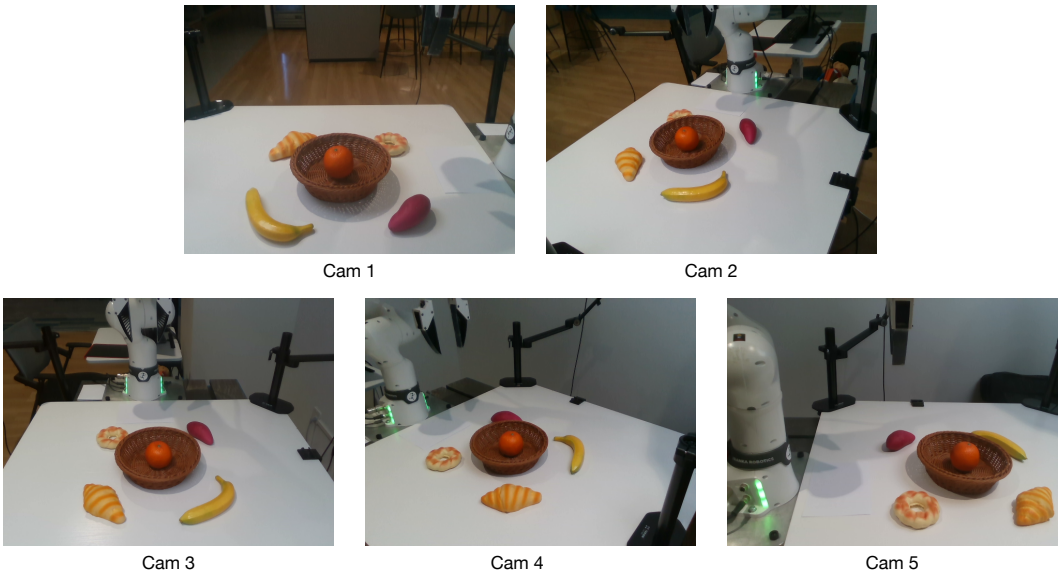


Figure 10: **Real-world training camera viewpoints.** Five third-person training placements (Cam 1–Cam 5) used for demonstration collection.

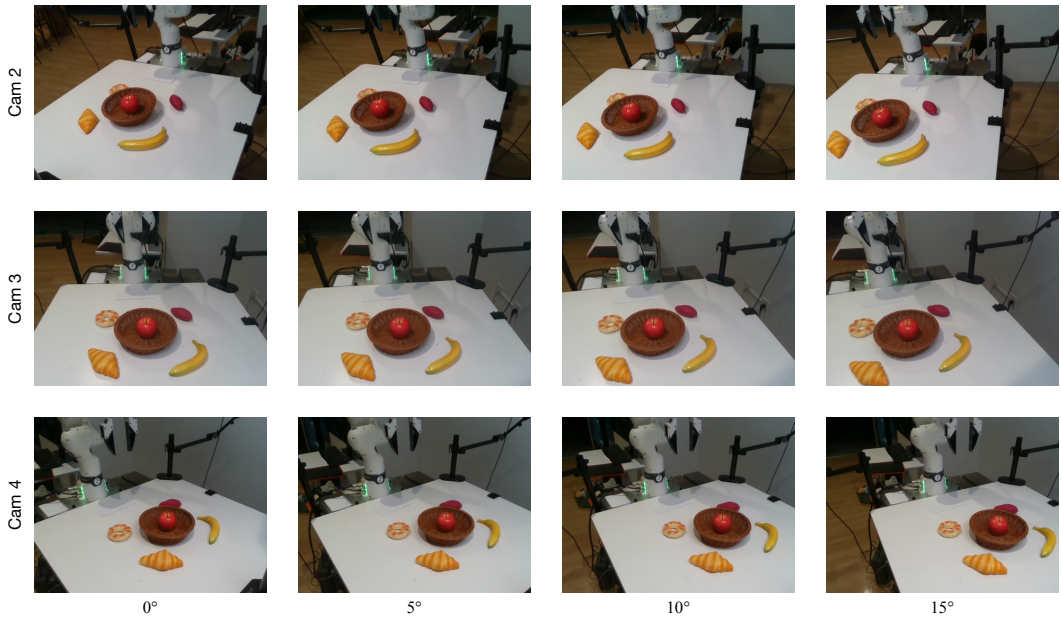


Figure 11: **Real-world unseen testing viewpoints.** Cam 2, Cam 3, and Cam 4 are horizontally rotated by $5°$, $10°$, and $15°$ from their canonical $0°$ training position to form unseen testing views.

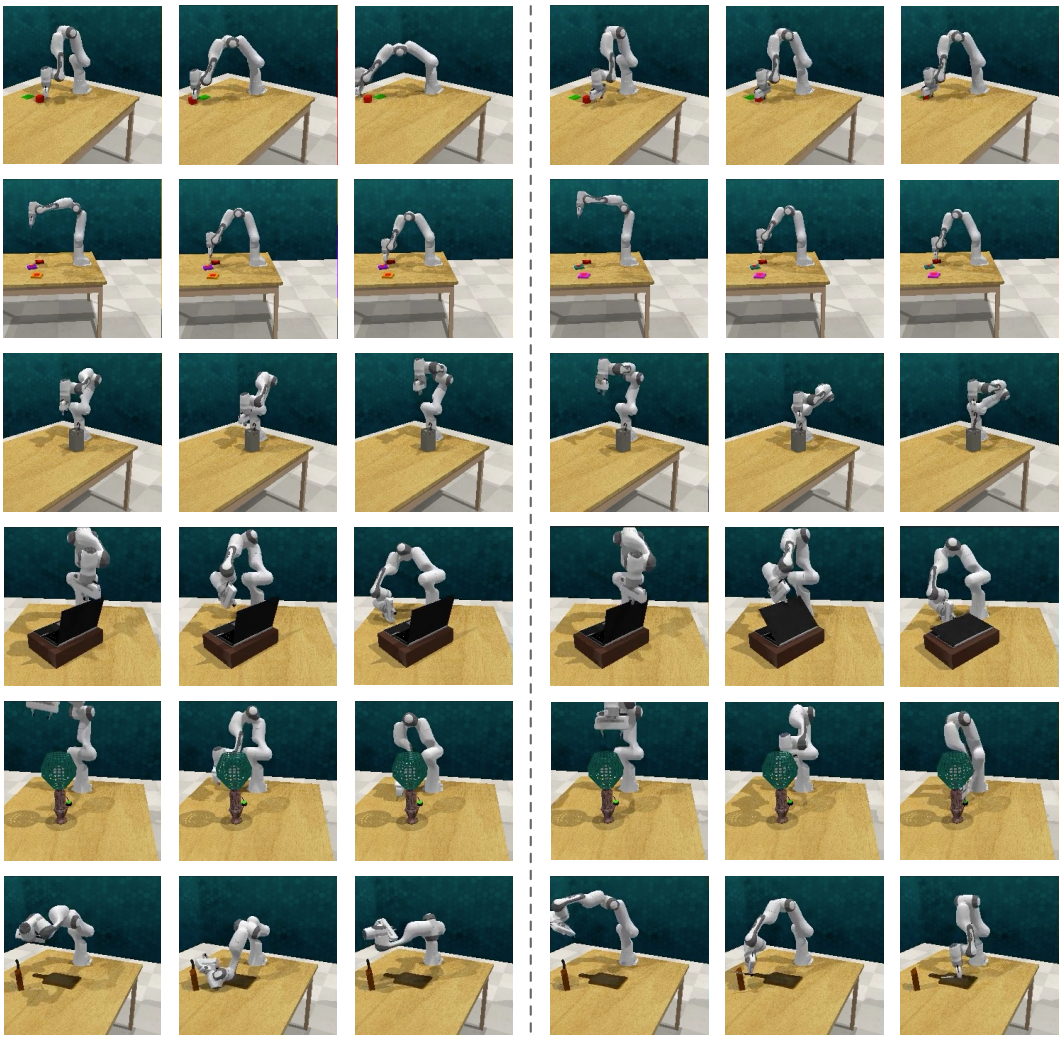


Figure 12: **Qualitative comparison between π_0 and CamVLA on RLBench under unseen cameras.**

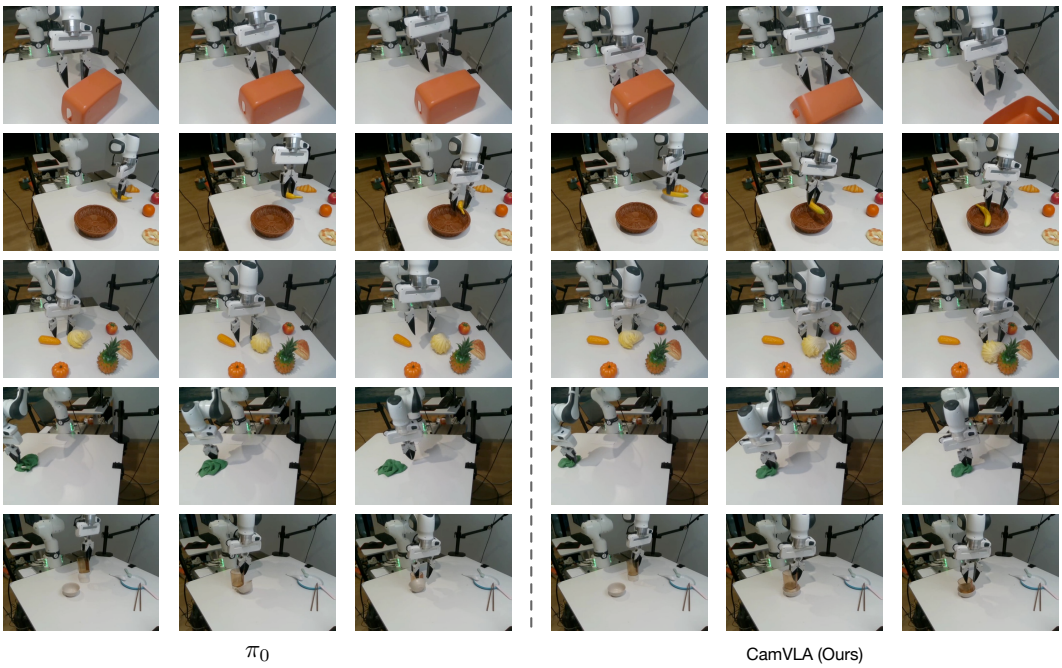


Figure 13: **Qualitative comparison between π_0 and CamVLA on real-world robot experiments under repositioned cameras.**

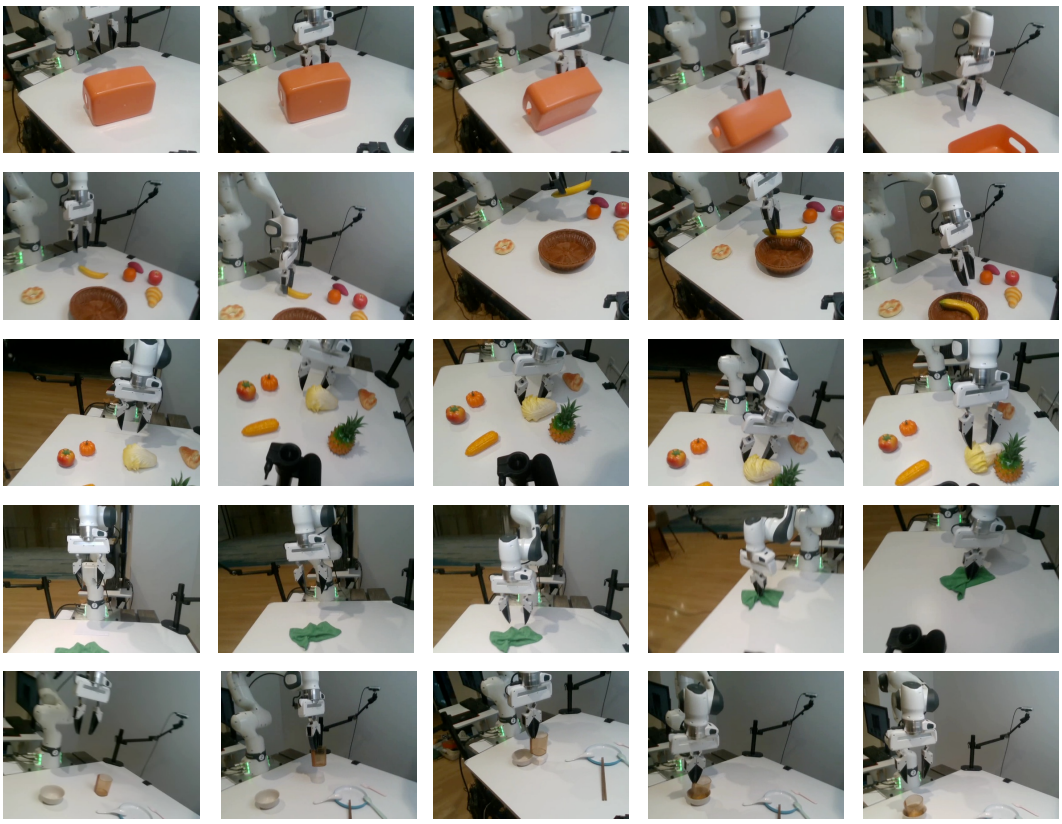


Figure 14: **Visualization of real-world experiments under dynamically hand-held moving cameras.** We visualize the robot's execution and dynamic hand-eye pose tracking when the third-person camera is continuously moved by a human operator during deployment.

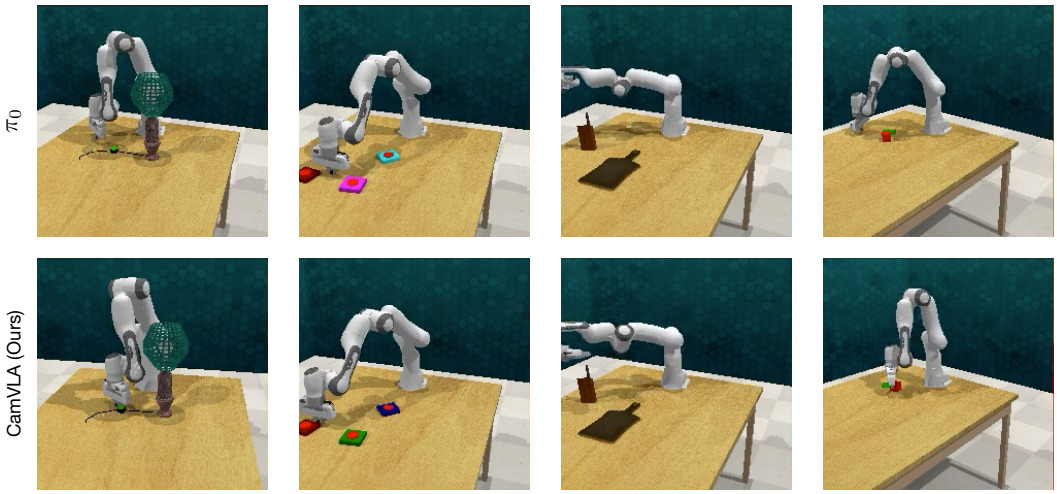


Figure 15: **Failure cases.** We illustrate common failure modes such as boundary objects, actions exceeding the robot's physical workspace, and self-occlusion.