

7 Appendix section

7.1 Coordinate system chaos impact analysis

To assess the impact of coordinate system chaos on VLA’s performance, we conduct a controlled experiment. The experiment consists of two main steps: First, we deliberately introduce controlled chaos into the LIBERO environment. Then, we compare the performance of models with and without 3D information. Our findings demonstrate that chaotic coordinate transformations significantly degrade model performance while incorporating 3D information effectively alleviates it.

Chaos generation. To simulate the diverse viewpoints in the pretraining dataset—where the robot’s body is absent from the image, and its coordinate system varies unpredictably—we select trajectories from a specific task in LIBERO-SPATIAL and render each trajectory from 30 distinct viewpoints. For each trajectory and its corresponding viewpoint, we introduce coordinate system chaos by applying a random translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $\mathbf{q} \in \mathbb{SO}(3)$ to the robot’s coordinate frame.

After applying the coordinate transformation, the gripper’s grasping state in the ground-truth action remains unchanged. However, the rotation offset $\Delta\theta$ and translation $\Delta\mathbf{x}$, along with the proprioceptive and camera pose information, undergo the following transformation:

$$\begin{aligned}\Delta\theta' &= \psi^{-1}(\mathbf{q}\psi(\Delta\theta)\mathbf{q}^\top), & \Delta\mathbf{x}' &= \mathbf{q}\Delta\mathbf{x}, \\ \theta' &= \psi^{-1}(\mathbf{q}\psi(\theta)), & \mathbf{x}' &= \mathbf{q}\mathbf{x} + \mathbf{t}, \\ \mathbf{R}' &= \mathbf{q}\mathbf{R}, & \mathbf{T}' &= \mathbf{q}\mathbf{T} + \mathbf{t}.\end{aligned}\tag{5}$$

Here, the function $\psi : \mathbb{R}^3 \rightarrow \mathbb{SO}(3)$ maps an Euler angle to its corresponding rotation matrix. The terms $\Delta\theta'$ and $\Delta\mathbf{x}'$ denote the transformed action values, while θ' , θ and \mathbf{x}' , \mathbf{x} represent the transformed and original rotation and position, respectively. Additionally, $[\mathbf{R}'|\mathbf{T}']$ and $[\mathbf{R}|\mathbf{T}]$ correspond to the transformed and original camera poses.

In the subsequent training process, we utilize the transformed action values $\Delta\theta'$ and $\Delta\mathbf{x}'$, along with the transformed camera parameters, for model training.

Implementation details. We employ a simple baseline, controlling for the involvement of 3D information. The baseline model extracts tokens from a single RGB view, while the alternative model converts an RGB-D frame into spatial vision tokens as input for the LLM Transformer. During testing, we do not apply random rotations or translations to the world coordinate system.

Experimental results. We control chaos levels by adjusting the magnitude of random rotations. Level 0 applies no rotation, while levels 1–3 introduce random z-axis rotations of 15°, 30°, and 90°, respectively. Translation \mathbf{t} is randomly set within a range of 0.5. As shown in Fig. 6, without chaos, both models perform well, with 3D information further boosting success rates. Notably, the 3D model shows lower variance across viewpoints. As chaos increases, the non-3D model’s performance drops sharply, while the 3D model maintains relatively high success—highlighting the value of 3D cues in handling coordinate system chaos.

7.2 Multi-view real-world evaluation

In this section, we conduct additional real-world experiments under a multi-view camera setup. We design two more challenging tasks to evaluate the model’s generalization ability with respect to: (i) variations in object locations together with the changed background environments; (ii) inputs from novel camera viewpoints. We use 4 fixed cameras to capture each demonstration from different angles, collecting 50 trajectories per task per camera—resulting in a total of 200 trajectories per task for training. All models are trained for 20 epochs, and performance is measured by success rate.

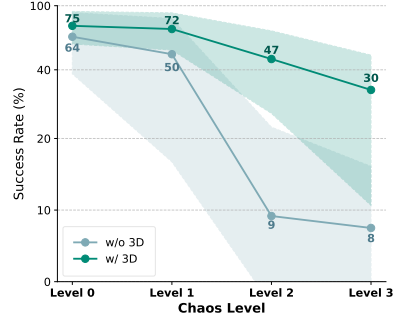


Figure 6: Success rates under varying coordinate chaos levels.

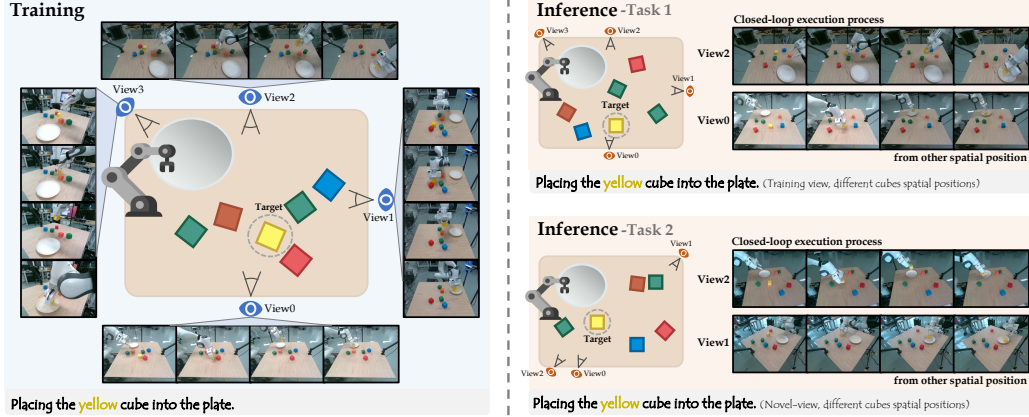


Figure 7: **Our multi-view real-world experiment settings.** These settings aim to evaluate the model’s out-of-distribution and novel-view generalization ability.

Method	In-View				Cross-View			Avg.
<i>Angle</i>	0°	90°	180°	225°	$\Delta 15^\circ (-15^\circ)$	$\Delta 25^\circ (-25^\circ)$	$\Delta 45^\circ (135^\circ)$	
OpenVLA [1]	25	15	30	10	30	10	5	18
4D-VLA (Ours)	60	50	65	65	50	55	40	55

Table 6: **Real-world multi-view evaluation.** We test our model’s spatial generalization across varying viewpoints and object layouts. 4D-VLA shows strong in-view and cross-view performance, highlighting its robustness under real-world distribution shifts.

Task descriptions. These two more challenging tasks are shown in Fig. 7. **Task 1: Out-of-distribution generalization.** The robot is tasked with placing a yellow cube into a plate under conditions where both the spatial configuration and the surrounding environment differ from those seen during training. These variations include changes in the plate’s position, the presence and location of distractor objects (e.g., other cubes). This task evaluates the model’s ability to generalize to unseen object arrangements and background contexts, testing its robustness in real-world deployments beyond the training distribution. **Task 2: Novel-view generalization.** Similar to Task 1, the robot is asked to place a yellow cube into a plate, with spatial setup and surroundings differing from training. However, during inference, data input is captured exclusively from an additional novel, unseen camera viewpoint that was not used during training. This task evaluates the model’s viewpoint robustness—its ability to generalize across camera perspectives and accurately interpret the scene from unfamiliar angles. Success in this task reflects strong spatial understanding and invariance to viewpoint changes, both critical for real-world multi-camera deployment. To simplify the setup, the target block is only moved within a small spatial range, while the background is fully randomized.

Evaluation metrics. Each multi-view task is evaluated over 20 trials. In every trial, both the background and object positions are randomly shuffled to assess the model’s robustness and generalization. The evaluation metric is the task success rate, computed as the ratio of successful trials to the total number of trials.

Experiments results. As shown in Tab. 6, 4D-VLA significantly outperforms OpenVLA in both in-view and cross-view settings, demonstrating strong generalization to viewpoint shifts and layout variations. In the in-view setting, where the camera is fixed but object layouts change, our model maintains consistently high success rates, indicating robustness to spatial perturbations. In the more challenging cross-view setting involving unseen viewpoints, 4D-VLA continues to perform stably across different angles. Although performance slightly drops at larger viewpoint shifts (e.g., $\Delta 45^\circ$), it remains stable compared to OpenVLA, whose success rate fluctuates more severely under such conditions. These results suggest that our model effectively captures spatial consistency across views, leading to more reliable visuomotor control in real-world environments.