FlowVLA: Enabling World Dynamics Understanding through Pluggable 3D Flow Learning

Junhyeong Go[1] **Mingyu Jung**[1] **Woobin Im**[2] **Jongbin Ryu**[1]

[1]Ajou University [2]Samsung Electronics

Abstract: Recent advances in 3D reconstruction enable the extraction of temporally consistent 3D motion from RGB observations; however, current Vision-Language-Action (VLA) models still predominantly rely on 2D image-text priors. Therefore, they do not accurately represent the fundamental dynamics of the 3D world associated with physical actions. This over-reliance on static 2D visual cues limits the capacity of VLA models in achieving robust robot manipulation. To address these limitations, we introduce **FlowVLA**, a reconstruction-guided, pluggable 3D flow learning framework aimed at empowering VLA models with an inherent understanding of action-relevant dynamics. Instead of requiring explicit 3D sensor inputs or altering the inference-time architecture, our method is designed to utilize temporally aligned 3D flow only during training. Our method allows the model to learn the dynamic evolution of the physical scene while preserving the computational efficiency of conventional 2D VLA backbones. To bridge scene-level dynamics with robotic manipulation, we propose a modular 3D flow learning framework defined by a dual-level flow supervision design. Specifically, point-level flow guides the model to forecast the future displacements of task-critical 3D points, while TCP-level flow aggregates these motions to align with the robot's Tool-Center-Point (TCP) trajectories. This dual-level flow supervision encourages the model to focus on semantically meaningful dynamic regions, effectively coupling scene-level motion with precise control signals. Extensive evaluations across simulation and real-world experiments demonstrate that FlowVLA consistently improves manipulation performance, zero-shot robustness, and optimizes data efficiency. FlowVLA adds only lightweight auxiliary heads during training and discards them at inference. Therefore, FlowVLA introduces no additional 3D sensor inputs, architectural modifications, or inference-time computation.

Keywords: Motion-relevant 3D Dynamics, 3D Scene Understanding, Training-only Supervision

1 INTRODUCTION

Reliable robotic manipulation should consider interaction within a dynamic, three-dimensional physical world. This dynamic interaction is addressed through the emergence of VLA models as a strong paradigm, transferring the rich semantic knowledge of pre-trained LLMs and VLMs to physical robot controls [1, 2, 3, 4, 5]. By integrating large-scale multimodal backbones with an action decoder, recent VLAs have achieved remarkable performance across diverse robotic tasks, including complex scene reasoning [6, 7], cross-embodiment generalization [8, 4, 9], and language-instructed manipulation [10, 3, 11].

Despite their impressive capabilities, most existing VLA models remain constrained by their dependence on multimodal representations derived from 2D image-text pretraining. Although these 2D representations provide rich semantic priors, they are not inherently designed to capture action-relevant 3D dynamics required for real-world robot manipulation. Consequently, action policies

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

often overfit to static visual cues—such as backgrounds, static camera viewpoints, or absolute global geometry—rather than prioritizing action-relevant 3D interactions and dynamics. To mitigate these limitations, recent approaches [12, 13, 14, 15] have explored 3D-aware VLAs that integrate explicit 3D signals, including depth maps or point clouds. Although these approaches improve physical manipulation by leveraging spatial understanding, they also present new challenges. In their approaches, auxiliary 3D sensors are required, and costly retraining on paired 3D-action datasets is demanded. More crucially, these approaches emphasize the extraction of static 3D geometry from single observations; as a result, they still overlook the dynamic evolution of the scene that is directly tied to physical manipulation.

To address these challenges, we propose **FlowVLA**, a reconstruction-guided, pluggable 3D flow learning framework that injects action-relevant 3D motion cues into VLA representations. Our core insight is to leverage 3D flow exclusively as a privileged signal during training to inject 3D dynamics directly into the model's latent representations. Instead of providing static snapshots, 3D flow provides direct supervision on how the physical environment moves and evolves under robot actions. Therefore, we employ 3D flow not as an input modality, but as a training-only module, while preserving the architecture of VLAs during inference. To achieve this, we derive temporally aligned 3D point trajectories from consecutive frames [16]. By computing the positional differences between adjacent frames, we formulate 3D flow that highlights regions of real physical movement within the scene. To effectively bring this spatial motion into the learning process, we establish a dual-level scene-to-control 3D flow learning framework comprising scene-level and control-level modules. The scene-level module captures dense 3D scene dynamics by supervising the future displacements of motion-salient 3D points, while the control-level module distills these dense motions into TCP-aligned flow that reflects the robot end-effector trajectory. By jointly optimizing these two modules, our framework bridges holistic environmental changes in 3D space with control-relevant robot motion.

We comprehensively validate FlowVLA through extensive evaluations on standard benchmarks, such as LIBERO and LIBERO-Plus, and real-world robot experiments. Empirical results demonstrate that our 3D sensor-free framework improves diverse VLA models with zero inference overhead. By improving task success rates in both simulated and real-world environments, the proposed approach highlights the potential of FlowVLA as a practical framework for 3D-aware policy learning.

2 RELATED WORK

2.1 Vision-Language-Action Models

VLAs integrate visual observations, proprioceptive states, and language instructions to generate robot actions. The viability of language-model-based sequence modeling for robot control was established by early studies [17, 18, 19], leading to RT-1 [1]. RT-2 [2] further showed that the semantic knowledge acquired from internet-scale pretraining can transfer effectively to robot manipulation. Recent advancements have concentrated on scaling, generalization, and expressive action generation. Octo [11] and OpenVLA [10] utilize large-scale robot datasets [8, 20] to improve adaptation to novel environments. Other works extend VLA models with continuous action generation via flow matching or diffusion [3, 4, 21], internal reasoning mechanisms [6, 7], and world modeling [22, 23]. Even with these advances, the current VLA paradigm still relies on representations inherited from 2D image-text pretraining. This makes existing models highly effective at semantic grounding, but leaves action-relevant 3D dynamics insufficiently modeled [15, 24, 25].

2.2 3D-Aware VLA Models

To improve spatial grounding in VLA models, recent works have incorporated explicit 3D information into their pipelines. Models such as LLaVA-3D [26], SpatialVLA [12], and DM0 [27] exploit 3D representations to enhance spatial understanding, whereas 3DVLA [15] injects 3D features into the VLM embedding space for improved reasoning capabilities. It is demonstrated by these methods that useful physical grounding for robot manipulation can be provided by 3D structures. However,

incorporating 3D information into 2D-pretrained backbones poses significant challenges. Dedicated alignment mechanisms are required due to the domain gap between pixel-based and geometric representations [14, 28, 13, 29]. Furthermore, high-quality 3D-action paired datasets are remarkably scarce compared to large-scale 2D corpora, making the computational cost of retraining prohibitively expensive [15, 30]. Several recent methods therefore explore more efficient ways to integrate 3D information. PointVLA [14] introduces lightweight modules for injecting 3D information without retraining the entire backbone, while GeoVLA [31] employs a parallel 3D embedding network to preserve multimodal alignment. BridgeVLA [13] instead projects 3D point clouds into multiple 2D views so that they can be processed within a unified representation space. Despite their effectiveness, these approaches typically rely on 3D sensor-derived inputs or 3D representations, and primarily emphasize static 3D geometry while providing limited supervision on how motion-salient regions physically evolve under robot actions. In contrast, **FlowVLA** uses reconstruction-derived 3D flow as a training-only privileged signal, encouraging existing 2D VLA policies to encode action-relevant 3D motion cues without the need for auxiliary 3D sensors or additional inference-time computation.

3 METHOD

As shown in Fig. 1, we present **FlowVLA**, a reconstruction-guided 3D flow learning framework that injects action-relevant 3D dynamics into VLA models through training-only supervision. Temporally aligned 3D point trajectories are derived from RGB sequences by our FlowVLA using a feed-forward dynamic reconstruction model [16]. Then, it converts 3D point trajectories into future 3D flow targets. These targets supervise the VLA with a dual-level scene-to-control objective: the action latent is encouraged by point-level flow to encode dense 3D scene motion, while motion cues are distilled into a control-oriented representation aligned with end-effector displacement by TCP-level flow.

3.1 Preliminaries

VDPM [16]. We adopt VDPM, a model intended to recover temporally consistent 3D point trajectories from RGB observations, to obtain a 3D motion signal without any other sensors. The core advantage of VDPM is its ability to consistently track points defined in a reference frame over time. Specifically, given an observation $o_{t_{\text{ref}}}$ at reference time step t_{ref}, VDPM represents the 3D location of each point on the reference pixel grid $u \in \{0, \dots, H-1\} \times \{0, \dots, W-1\}$ at an arbitrary time step t' as $P_{t_{\text{ref}}}(t', \pi) \in \mathbb{R}^{H \times W \times 3}$, where π denotes the coordinate system in which the 3D points are expressed. Thus, $P_{t_{\text{ref}}}(t', \pi)(u)$ is the 3D position of the point corresponding to pixel u in the reference frame, evaluated at time step t'. For fixed t_{ref} and π, the 3D trajectory of a reference point over a sequence is defined as $\{P_{t_{\text{ref}}}(t', \pi)(u)\}_{t'=0}^{T-1}$.

Coordinate Alignment. We set the first frame as the reference frame, i.e., $t_{\text{ref}} = 0$, and define the corresponding reference camera coordinate as π_0. VDPM initially generates reference-anchored point trajectories within this reference camera coordinate, denoted as $P_0(t', \pi_0)(u)$. However, robot motion, including TCP displacement, is defined in the robot base coordinate. Before constructing flow targets, all 3D points are expressed in the robot base coordinate π_{base} to ensure that the reconstruction-derived point trajectories are directly comparable with robot motion. For each task, we compute a rigid transform $T_{\text{base}\leftarrow 0} = [R_{\text{base}\leftarrow 0} \mid \mathbf{t}_{\text{base}\leftarrow 0}] \in SE(3)$ from the reference camera frame to the robot base frame. Subsequently, we proceed to transform each reference-anchored 3D point as follows:

$$P_0(t', \pi_{\text{base}})(u) = R_{\text{base}\leftarrow 0} P_0(t', \pi_0)(u) + \mathbf{t}_{\text{base}\leftarrow 0}. \tag{1}$$

For a trajectory of length N, we define the 3D point flow at time t' in the robot base coordinate as:

$$\Delta P_0(t', u) = P_0(t'+1, \pi_{\text{base}})(u) - P_0(t', \pi_{\text{base}})(u), \quad t' = 0, \dots, N-2. \tag{2}$$

Here, $\Delta P_0(t', u) \in \mathbb{R}^3$ is referred to as the 3D displacement of reference point u from t' to $t'+1$ in the robot base coordinate. This alignment assures that the resulting 3D point flows are defined in the same coordinate system as the TCP displacements used in the TCP-level objective.

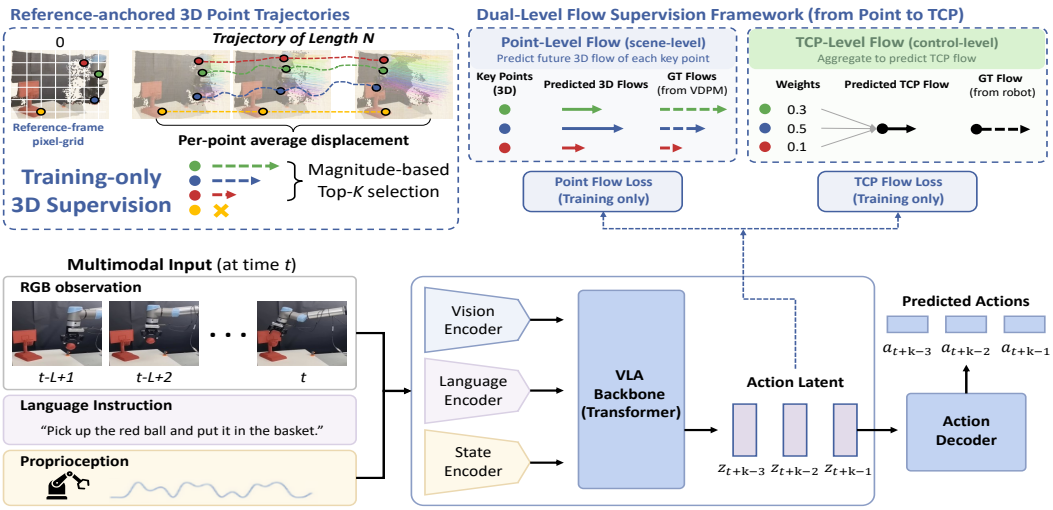


Figure 1: **Overview of Training-only 3D flow supervision.** Reference-anchored 3D points are tracked over time to construct 3D flow targets, from which top-K motion-salient points are selected. Point-Level Flow predicts scene-level motion, while TCP-Level Flow aggregates it into end-effector motion, bridging scene-level dynamics and control-relevant motion.

3.2 Reconstruction-Guided 3D Flow Construction

Given visual observations $o_{t-L+1:t}$, a language instruction l, and proprioceptive states $s_{t-L+1:t}$, where L denotes the historical observation length and k denotes the future prediction horizon, the VLA backbone generates action latent representations $z_{t:t+k-1}$ as follows:

$$z_{t:t+k-1} = f_\theta(f_v(o_{t-L+1:t}), f_l(l), f_s(s_{t-L+1:t}))\,, \tag{3}$$

$$a_{t:t+k-1} = p_\theta(z_{t:t+k-1})\,, \tag{4}$$

where f_θ and p_θ denote the VLA backbone and action policy, respectively, and f_v, f_l, and f_s act as modality-specific encoders. While rich semantic priors are provided by these 2D-centric representations, they often struggle to capture the underlying 3D dynamics essential for complex manipulation. The reference-anchored 3D flow defined in Eq. 2 is utilized as a reconstruction-derived, training-only supervision signal to bridge this gap.

Magnitude-based Selection. Since not all points contribute equally to action-relevant dynamics, we prioritize the selection of dynamically informative points according to their motion magnitude. This filtering directs the model's attention towards dynamically informative regions rather than static backgrounds or reconstruction artifacts. Specifically, for each point u on the reference-image pixel grid, we compute its instantaneous flow magnitude at time τ within an observation window of length $L+k$ using the formula $m_\tau(u) = \|\Delta P_0(\tau, u)\|_2$, where τ ranges from $t-L+1$ to $t+k-1$. The average flow magnitude over the sequence window is calculated as follows:

$$\bar{m}_t(u) = \frac{1}{L+k-1} \sum_{\tau=t-L+1}^{t+k-1} m_\tau(u). \tag{5}$$

By selecting the top-K points that exhibit the highest $\bar{m}_t(u)$ values, we construct a keypoint set $\mathcal{U}_t = \mathrm{TopK}\left(\{\bar{m}_t(u)\}_u, K\right)$, which identifies regions with dominant 3D motion. For each selected point $u \in \mathcal{U}_t$, the future 3D flow target at prediction step j is defined as:

$$\Delta P_0^{(j)}(t,u) := \Delta P_0(t+j, u) = P_0(t+j+1, \pi_{\text{base}})(u) - P_0(t+j, \pi_{\text{base}})(u), \tag{6}$$

where $j \in \{0, \ldots, k-1\}$ and $t+k \leq N-1$. These selected points and their corresponding future displacements serve as ground-truth supervision signals for the dual-level flow learning objective, guiding the VLA action latent to encode the 3D evolution of the workspace.

3.3 Dual-Level 3D Flow Learning

Point-Level Flow. The point-level flow branch facilitates the action latent representation in effectively capturing the future motion of dynamically informative 3D points. With the selected keypoint set $\mathcal{U}_t$ and the future flow targets $\Delta P_0^{(j)}(t,u)$ from Sec. 3.2, the action latent token z_{t+j} is first mapped to a higher-dimensional space and broadcast across the $K = |\mathcal{U}_t|$ points as follows:

$$\mathbf{h}_t^{(j)} = \text{Broadcast}\left(g_\phi(z_{t+j}), K\right) \in \mathbb{R}^{K\times d_h}, \quad j \in \{0,\dots,k-1\}. \tag{7}$$

Here, g_ϕ is a 2-layer MLP and d_h represents the latent dimension. To provide spatial context during training, we collect the 3D coordinates of the selected points at time step $t+j$ as:

$$\mathbf{P}_t^{(j)} = \left[P_0(t+j, \pi_{\text{base}})(u)\right]_{u\in\mathcal{U}_t} \in \mathbb{R}^{K\times 3}. \tag{8}$$

We fuse the broadcast action feature with the selected 3D coordinates to obtain $\mathbf{F}_t^{(j)} = q_\phi([\mathbf{h}_t^{(j)}; \mathbf{P}_t^{(j)}])$, where q_ϕ is a point-wise feature extractor. The flow head h_{flow} predicts the future displacement of each selected point as $\Delta\hat{\mathbf{P}}_0^{(j)}(t) = h_{\text{flow}}(\mathbf{F}_t^{(j)}) \in \mathbb{R}^{K\times 3}$. These point-wise predictions are supervised by the point-level flow loss:

$$\mathcal{L}_{\text{point}} = \frac{1}{kK}\sum_{j=0}^{k-1}\sum_{u\in\mathcal{U}_t} \rho\Big(\Delta\hat{P}_0^{(j)}(t,u), \Delta P_0^{(j)}(t,u)\Big), \tag{9}$$

where $\rho(\cdot,\cdot)$ denotes the Smooth L1 loss. This objective encourages the action latent to reflect scene-level 3D dynamics by predicting the future motion of selected points.

TCP-Level Flow. While point-level supervision grounds the model in scene dynamics, not all motions are equally relevant to the control task. We aggregate the predicted point flows into a TCP flow estimate to focus on control-relevant dynamics. Importance weights $\mathbf{r}_t^{(j)}$ are predicted through a softmax-normalized weighting head h_{weight} using the fused representation $\mathbf{F}_t^{(j)}$:

$$\mathbf{r}_t^{(j)} = \text{Softmax}_u\Big(h_{\text{weight}}\Big(\mathbf{F}_t^{(j)}\Big)\Big) \in \mathbb{R}^K. \tag{10}$$

The predicted TCP-level flow is then computed as the weighted sum of point-wise displacements:

$$\Delta\hat{\mathbf{x}}_{\text{tcp}}^{(j)}(t) = \sum_{u\in\mathcal{U}_t} \mathbf{r}_t^{(j)}(u)\,\Delta\hat{P}_0^{(j)}(t,u) \in \mathbb{R}^3. \tag{11}$$

The TCP-level flow loss is then defined as:

$$\mathcal{L}_{\text{tcp}} = \frac{1}{k}\sum_{j=0}^{k-1} \rho\Big(\Delta\hat{\mathbf{x}}_{\text{tcp}}^{(j)}(t), \Delta\mathbf{x}_{\text{tcp}}^{(j)}(t)\Big), \tag{12}$$

where $\Delta\mathbf{x}_{\text{tcp}}^{(j)}(t) = \mathbf{x}_{\text{tcp}}(t+j+1) - \mathbf{x}_{\text{tcp}}(t+j)$ is the ground-truth TCP flow. This scene-to-control distillation encourages the action latent to capture control-relevant dynamics in addition to scene-level motion. The final objective combines the action loss with the auxiliary flow losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \lambda_{\text{point}}\mathcal{L}_{\text{point}} + \lambda_{\text{tcp}}\mathcal{L}_{\text{tcp}}, \tag{13}$$

where λ_{point} and λ_{tcp} are the loss weights. Critically, all auxiliary heads and 3D geometric inputs are discarded at inference. The policy operates through its original 2D-based pathway, benefiting from 3D-aware representation with zero inference-time overhead.

4 SIMULATION EXPERIMENTS

Settings. The proposed 3D flow supervision is evaluated across heterogeneous policy designs to confirm the generalization ability of FlowVLA. We instantiate FlowVLA on two representative VLAs with substantially different action-generation policies: Seer [22] and $\pi_{0.5}$ [4]. Seer adopts a two-layer MLP-based policy that reconstructs future observations and predicts actions, whereas $\pi_{0.5}$ employs a PaliGemma-based [32] VLM backbone with a flow-matching action policy. For $\pi_{0.5}$, we follow the official fine-tuning recipe and perform LoRA adaptation [33]. For both backbones, we attach 3D flow objectives only during training while preserving their model architectures and inference pathways. Further details on the simulation settings and datasets are provided in Appendix A.

Table 1: **Results on LIBERO Benchmark.** For Seer [22], we follow the original LIBERO-Long evaluation protocol, averaging the top-3 checkpoints over 20 rollouts per task.

Method	Spatial (%)	Object (%)	Goal (%)	Long (%)	Avg. (%)
Diffusion Policy *(RSS'23)*	78.3	92.5	68.3	50.5	72.4
OpenVLA *(CoRL'24)*	84.7	88.4	79.2	53.7	76.5
π_0 *(RSS'25)*	96.8	98.8	95.8	85.2	94.2
CoT-VLA *(CVPR'25)*	87.5	91.6	87.6	69.0	83.9
π_0-FAST *(RSS'25)*	96.4	96.8	88.6	60.2	85.5
SpatialVLA *(RSS'25)*	88.2	89.9	78.6	55.5	78.1
Octo *(RSS'24)*	78.9	85.7	84.6	51.1	75.1
UniVLA *(RSS'25)*	96.5	96.8	95.6	92.0	95.2
Seer *(ICLR'25)*	84.2	94.8	89.8	80.4	87.3
Seer + FlowVLA (Ours)	93.5	96.5	91.8	87.8	92.4
$\pi_{0.5}$ *(CoRL'25)*	98.8	98.2	98.0	92.4	96.8
$\pi_{0.5}$ **+ FlowVLA (Ours)**	99.0	98.8	97.4	94.8	97.5

Table 2: **Zero-shot evaluation on LIBERO-Plus.** Results demonstrate the robustness of policies trained on LIBERO benchmarks under unseen environmental perturbations.

	Camera	Robot	Language	Light	Background	Noise	Layout	Avg. (%)
OpenVLA	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
WorldVLA	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
UniVLA	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
Seer	22.1	47.7	86.3	12.3	2.0	22.9	50.0	36.1
Seer + FlowVLA	24.6	54.6	87.0	25.1	8.1	25.5	57.2	41.1
$\pi_{0.5}$	21.4	39.1	68.4	86.7	83.3	36.1	84.6	57.5
$\pi_{0.5}$ **+ FlowVLA**	24.1	39.9	68.8	90.4	87.1	37.7	84.7	59.2

4.1 Simulation Results

LIBERO. Tab. 1 summarizes the results on the LIBERO benchmark. Our scene-to-control 3D flow supervision yields consistent improvements on recent VLA models, with particularly large gains in long-horizon manipulation. It is notable that FlowVLA achieves particularly large gains on **LIBERO-Long**, which requires sustained spatial-temporal reasoning over extended horizons. On Seer [22], our method improves the success rate from 80.4% to 87.8%, a substantial 7.4%p increase. Even on the state-of-the-art $\pi_{0.5}$, FlowVLA improves the success rate from 92.4% to 94.8%. Importantly, these performance gains are achieved with *zero inference-time overhead*, as the auxiliary flow heads are entirely decoupled after training.

LIBERO-Plus. We conduct an additional evaluation of zero-shot robustness on LIBERO-Plus, and report the results in Tab. 2. The average success rate is improved from 36.1% to 41.1% on Seer [22] model. The performance improvements are particularly noticeable for perturbation types such as *Camera*, *Robot*, *Light*, *Noise*, and *Layout*. This suggests that reconstruction-derived 3D flow supervision guides the policy to focus on action-relevant 3D dynamics, rather than overfitting to superficial visual appearances. Notably, we also observe a modest gain under the *Language* perturbation, indicating that the proposed method enhances dynamic motion grounding without compromising the semantic priors acquired from image-text pretraining.

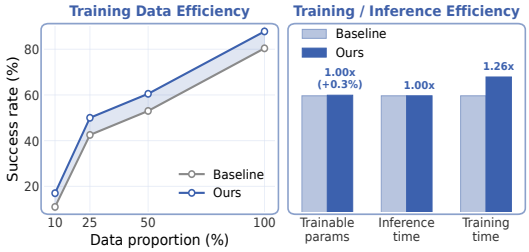
4.2 Ablation Studies

Component Analysis. We verify the contribution of each design choice through a component ablation study performed on **LIBERO-Long** (Tab. 3). ❶ **Coordinate alignment:** Aligning 3D point trajectories across different camera coordinate systems improves performance, indicating that robot-aligned flow targets are important for consistent supervision. ❷ **Reference-frame alignment:** Using reference-anchored, temporally trackable 3D points further improves performance, supporting

Table 3: **Component analysis on LIBERO-Long.**

Method	Avg. (%)
FlowVLA (Ours)	**87.8**
– w/o coordinate alignment	86.3
– w/o reference-frame alignment	85.2
– w/o magnitude-based selection	84.0
– w/o TCP-level loss	83.8
– w/o point-level loss	84.8
None (Baseline)	80.4

Figure 2: **Analysis of training data efficiency and computational cost.**



their effectiveness for defining reliable 3D flow targets. ❸ **Magnitude-based selection:** Replacing magnitude-based point selection with *random-K selection* reduces performance, indicating that directing supervision towards dynamic, motion-salient regions is important for efficient action-relevant learning. ❹ **Dual-level flow supervision:** Removing either *TCP-level flow* or *point-level flow* degrades performance, demonstrating that FlowVLA benefits from both scene-level motion grounding and end-effector-aligned control supervision. Additional qualitative visualizations of the magnitude-based point selection and TCP-level aggregation weights are provided in Appendix C.

Data Efficiency and Computational Cost. In Fig. 2, we analyze FlowVLA in terms of data efficiency and computational cost. FlowVLA consistently outperforms the baseline across all data regimes when trained with 10%, 25%, and 50% of the full dataset. This suggests that utilizing training-only 3D flow supervision provides a useful inductive bias under limited data, allowing for less reliance on superficial visual correlations and better capture of control-relevant scene dynamics by the policy. It is also evident that the computational overhead is kept at a manageable level: the proposed flow heads increase the total number of parameters by only 0.34% and add no inference-time computation. The only cost occurs during training, where total training time increases by approximately 1.26×. This trade-off is favorable given the consistent gains across evaluated tasks and backbones.

Effect of Supervision Signal. We compare different supervision targets under the same training and evaluation protocol. Tab. 4 shows that dynamic 3D flow supervision outperforms both 2D optical flow and static 3D point supervision. In contrast to 2D optical flow, 3D flow captures motion in physical space instead of image-plane displacement. This allows it to align more closely with scene-level dynamics and end-effector motion. Unlike static 3D point supervision, 3D flow emphasizes state change rather than static geometry. This supports our motivation for using dynamic 3D flow, which captures temporally varying scene dynamics more directly tied to action. For a fair comparison, all supervision baselines are implemented with the same auxiliary-head capacity and training protocol as FlowVLA.

Table 4: **Effect of supervision signal.**

Supervision Signal	Avg. (%)
None (Baseline)	80.4
2D optical flow (CoTracker [34])	84.5
Static 3D point cloud	85.0
Dynamic 3D flow (Ours)	**87.8**

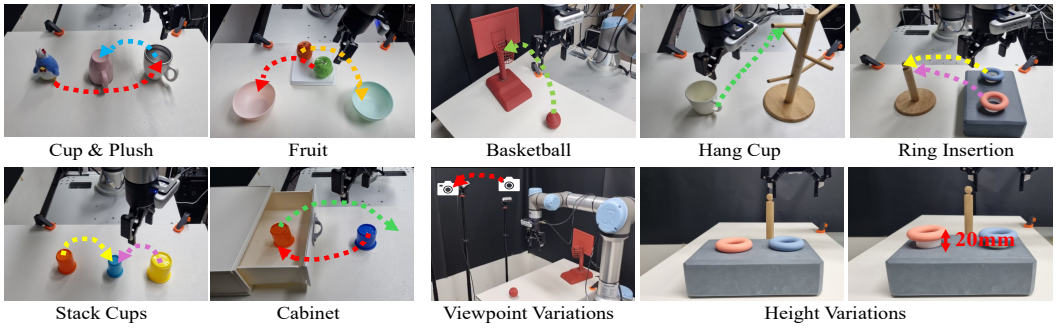
5 REAL-WORLD EXPERIMENTS

Settings. We construct a real-world deployment setup with a UR5e platform equipped with third-person and wrist-view cameras. We evaluate four long-horizon manipulation tasks—*Cup & Plush*, *Fruit*, *Stack Cups*, and *Cabinet*—and three more challenging tasks that require stronger 3D spatial understanding—*Hang Cup*, *Basketball*, and *Ring Insertion*. Zero-shot generalization is also evaluated under **viewpoint** and **height variations**. Further details on the setup are provided in Appendix B.

Real-world Results. To validate real-world manipulation performance, we evaluate FlowVLA on four long-horizon base tasks and three 3D-aware tasks. In Tab. 5, FlowVLA consistently improves baseline performance across the base tasks; notably, on *Stack Cups*, FlowVLA fails only once out of 15 trials, demonstrating reliable multi-step manipulation. The performance gain becomes more pronounced in the 3D-aware tasks that require precise spatial reasoning and action-relevant 3D dynamics.

Table 5: **Real-world results**. We evaluate each task over 15 trials and report success rates (%).

	Base Tasks				3D-aware Tasks		
	Cup & Plush	Fruit	Stack Cups	Cabinet	Basketball	Hang Cup	Ring Insertion
Seer	60.0	66.7	73.3	60.0	60.0	46.7	66.7
Seer + FlowVLA	73.3	73.3	93.3	73.3	80.0	86.7	73.3



(a) Base tasks (b) 3D-aware tasks

Figure 3: **Task visualization.** All policies are trained with only 40 demonstrations per task.

FlowVLA improves *Basketball* by approximately 20%p over the baseline and substantially boosts *Hang Cup* from 46.7% to 86.7%. These results indicate that scene-to-control 3D flow supervision effectively enhances control-relevant 3D scene understanding in real-world manipulation.

Table 6: **Test-time variation results.**

Method	Viewpoint	Height
Seer	0	46.7
Seer + FlowVLA	46.7	73.3

Results under Test-Time Variations. We further evaluate zero-shot robustness under test-time variations. For *Basketball*, we adjust the angle and height of the third-person camera; the initial ring height is modified for *Ring Insertion*. As shown in Tab. 6, the baseline fails under viewpoint variation with 0% success, whereas FlowVLA achieves 46.7%. Under height variation in *Ring Insertion*, FlowVLA improves the success rate from 46.7% to 73.3%, with a reduction in gripper collisions and inaccurate insertions. These results demonstrate that FlowVLA improves both in-distribution performance and zero-shot robustness against real-world spatial variations. A detailed failure case analysis is provided in Appendix C.3.

6 CONCLUSION

We present **FlowVLA**, a reconstruction-guided 3D flow learning framework that injects action-relevant 3D dynamics into existing VLA models through training-only supervision. By deriving temporally aligned 3D point trajectories from RGB observations into future 3D flow targets, FlowVLA couples scene-level point motion with control-relevant TCP displacement through dual-level flow supervision. Experiments on LIBERO, LIBERO-Plus, and real-world tasks show consistent improvements in performance and robustness across different VLA backbones. FlowVLA notably operates without auxiliary 3D sensors, architectural modifications, or inference-time computation, as all 3D flow modules are used only during training. These results suggest that proposed 3D flow supervision serves as an effective approach to impart action-relevant 3D dynamics to 2D VLA models.

Limitations. FlowVLA requires an offline reconstruction step to generate 3D point trajectories for training. Although this does not affect inference-time efficiency, the quality of the auxiliary supervision depends on the reconstruction model. In addition, while the visualization in Appendix C.2 suggests that magnitude-based point selection tends to capture dynamically active regions, it does not explicitly guarantee task relevance. This suggests future extensions with semantic or contact-aware point selection.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.jun:여기 작성해야함.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022.

[2] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Conference on Robot Learning, pages 2165–2183. PMLR, 2023.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. pi_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

[4] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

[5] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. arXiv preprint arXiv:2506.01844, 2025.

[6] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine. Robotic control via embodied chain-of-thought reasoning. arXiv preprint arXiv:2407.08693, 2024.

[7] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 1702–1713, 2025.

[8] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 6892–6903. IEEE, 2024.

[9] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. arXiv preprint arXiv:2503.20020, 2025.

[10] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv. org/abs/2406.09246, 1(2):4, 2024.

[11] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. arXiv preprint arXiv:2405.12213, 2024.

[12] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830, 2025.

[13] P. Li, Y. Chen, H. Wu, X. Ma, X. Wu, Y. Huang, L. Wang, T. Kong, and T. Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. arXiv preprint arXiv:2506.07961, 2025.

[14] C. Li, J. Wen, Y. Peng, Y. Peng, and Y. Zhu. Pointvla: Injecting the 3d world into vision-language-action models. IEEE Robotics and Automation Letters, 11(3):2506–2513, 2026.

[15] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3d-vla: A 3d vision-language-action generative world model. arXiv preprint arXiv:2403.09631, 2024.

[16] E. Sucar, E. Insafutdinov, Z. Lai, and A. Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps. arXiv preprint arXiv:2601.09499, 2026.

[17] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691, 2022.

[18] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. arXiv preprint arXiv:2205.06175, 2022.

[19] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378, 2023.

[20] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In Conference on Robot Learning, pages 1723–1736. PMLR, 2023.

[21] J. Wen, Y. Zhu, M. Zhu, Z. Tang, J. Li, Z. Zhou, X. Liu, C. Shen, Y. Peng, and F. Feng. Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression. In Forty-second International Conference on Machine Learning, 2025.

[22] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. arXiv preprint arXiv:2412.15109, 2024.

[23] J. Cen, C. Yu, H. Yuan, Y. Jiang, S. Huang, J. Guo, X. Li, Y. Song, H. Luo, F. Wang, et al. Worldvla: Towards autoregressive action world model. arXiv preprint arXiv:2506.21539, 2025.

[24] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. arXiv preprint arXiv:2311.12871, 2023.

[25] S. Chen, R. Garcia, C. Schmid, and I. Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. arXiv preprint arXiv:2309.15596, 2023.

[26] C. Zhu, T. Wang, W. Zhang, J. Pang, and X. Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. arXiv preprint arXiv:2409.18125, 2024.

[27] E. Yu, H. Lv, J. Sun, K. Lin, R. Zhang, Y. Shi, Y. Chen, Z. Chen, Z. Zhang, F. Jia, et al. Dm0: An embodied-native vision-language-action model towards physical ai. arXiv preprint arXiv:2602.14974, 2026.

[28] Z. Zhou, P. Wang, Z. Liang, H. Bai, and R. Zhang. Cross-modal 3d representation with multi-view images and point clouds. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 3728–3739, 2025.

[29] Z. Chen, L. Jing, Y. Li, and B. Li. Bridging the domain gap: Self-supervised 3d scene understanding with foundation models. Advances in Neural Information Processing Systems, 36: 79226–79239, 2023.

[30] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin. Pointllm: Empowering large language models to understand point clouds. In European Conference on Computer Vision, pages 131–147. Springer, 2024.

[31] L. Sun, B. Xie, Y. Liu, H. Shi, T. Wang, and J. Cao. Geovla: Empowering 3d representations in vision-language-action models. arXiv preprint arXiv:2508.09071, 2025.

[32] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726, 2024.

[33] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. Iclr, 1(2):3, 2022.

[34] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker: It is better to track together. In European conference on computer vision, pages 18–35. Springer, 2024.

[35] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems, 36: 44776–44791, 2023.

[36] S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626, 2025.

[37] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 5294–5306, 2025.

[38] Y. Xiao, J. Wang, N. Xue, N. Karaev, Y. Makarov, B. Kang, X. Zhu, H. Bao, Y. Shen, and X. Zhou. Spatialtrackerv2: 3d point tracking made easy. arXiv preprint arXiv:2507.12462, 2025.

A SIMULATION ENVIRONMENTS

A.1 Datasets

LIBERO. LIBERO [35] is a standardized benchmark for robot manipulation. It consists of 4 task suites—LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long—where each suite contains 10 tasks and 50 expert demonstrations per task. These suites are designed to evaluate complementary aspects of manipulation capability, including spatial reasoning, generalization to unseen object instances, language-conditioned manipulation, and long-horizon planning.

LIBERO-Plus. LIBERO-Plus [36] extends LIBERO to provide a more fine-grained evaluation of robustness under distribution shifts. It introduces 7 perturbation dimensions and contains 10,030 test episodes organized across 5 difficulty levels. These perturbations include changes in camera viewpoint, robot initialization, language instructions, lighting, background, layout, and sensor noise, thereby enabling a detailed analysis of zero-shot robustness beyond standard in-distribution evaluation.

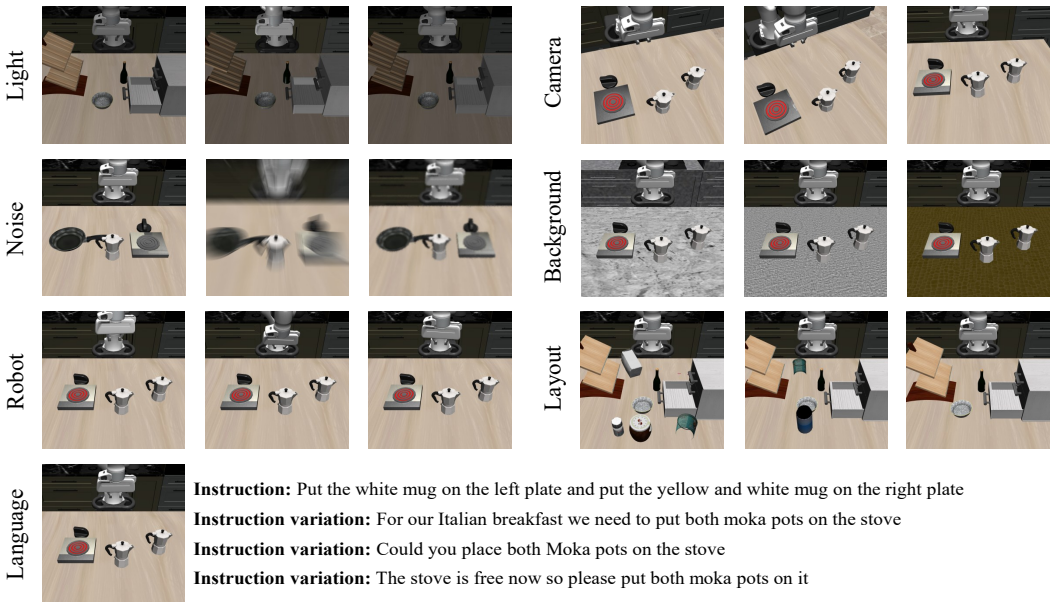


Figure 4: **LIBERO-Plus Benchmark.** LIBERO-Plus evaluates robustness under distribution shifts across seven perturbation categories: lighting, camera viewpoint, sensor noise, background, language instruction, scene layout, and robot initialization.

A.2 Ablation Study

Magnitude-based Top-K Selection. We ablate the number of selected motion-salient 3D points under the same training setting by varying $K \in \{1024, 2048, 3072\}$. As shown in Tab. 7, $K = 2048$ achieves the best performance. A smaller value, $K = 1024$, may not provide sufficient coverage of scene-level motion, limiting the supervision of diverse motion-relevant dynamics. Conversely, a larger value, $K = 3072$, can include less informative regions such as static backgrounds or reconstruction noise, which weakens the quality of the flow supervision. Nevertheless, FlowVLA maintains strong performance across all tested values of K, indicating that the method is not highly sensitive to this hyperparameter. Thus, $K = 2048$ offers the best trade-off between capturing meaningful 3D motion and filtering out irrelevant points.

Table 7: **Ablation on the number of Top-K.**

Top-K	1024	2048	3072
Success (%)	86.0	**87.8**	85.2

Effect of Reconstruction Model for 3D Flow Supervision. To examine the dependency of FlowVLA on the reconstruction model, we replace VDPM [16] with VGGT [37] and and another temporally consistent dynamics tracker, SpatialTrackerV2 [38], for generating 3D flow supervision while keeping the policy architecture, training protocol, and evaluation setting unchanged. All 3D flow supervision settings improve over the baseline, indicating that FlowVLA is not specific to VDPM. However, VGGT performs lower than VDPM- and SpatialTrackerV2-based supervision. We attribute this gap to the fact that VDPM and SpatialTrackerV2 are explicitly designed for temporally consistent dynamic point trajectories, which provide cleaner motion targets for action-relevant 3D flow learning.

Table 8: **Ablation on reconstruction models for 3D flow supervision.**

Method	Baseline	VGGT	SpatialTrackerV2	VDPM
Success (%)	80.4	86.8	88.0	**87.8**

Sensitivity to Noisy 3D Flow Supervision. To evaluate the robustness of FlowVLA to imperfect reconstruction targets, we conduct a noise sensitivity analysis on LIBERO-Long. After selecting K motion-salient sparse points, we randomly corrupt a fixed percentage of selected point trajectories by adding zero-mean Gaussian noise to their 3D coordinates. As shown in Tab. 9, performance decreases as the corruption ratio increases, from 87.8% without noise to 84.7% and 84.0% under 5% and 20% corruption, respectively. Nevertheless, FlowVLA still outperforms the baseline of 80.4% under both settings, indicating that the proposed supervision is not overly sensitive to moderate reconstruction noise. These results also confirm that the quality of reconstructed 3D trajectories remains an important factor for effective flow supervision.

Table 9: **Noise sensitivity analysis.**

Noise ratio (%)	0	5	20	Baseline
Success (%)	**87.8**	84.7	84.0	80.4

Offline Reconstruction Cost. For Seer [22] experiments on LIBERO-Long, VDPM [16] preprocessing takes approximately 4 wall-clock hours to generate 3D flow supervision, compared to approximately 13 wall-clock hours for policy training excluding this preprocessing. This offline cost is about 30% of the policy training time, is incurred only once before training, and the generated supervision targets are cached for subsequent experiments. Since the 3D flow supervision is training-only, it does not introduce any additional inference-time computation.

B REAL-WORLD ENVIRONMENTS

B.1 Real-World Setups

As illustrated in Fig. 5, our real-world experimental setup consists of a single UR5e robotic arm equipped with two RealSense D435i cameras. The real-world evaluation comprises four long-horizon base tasks and three 3D-aware task scenarios. To assess the zero-shot generalization capability of VLA models, we additionally evaluate the models under viewpoint and height variations. For the Basketball task, we build the task environment using a Bambu Lab P1S 3D printer. For collecting training demonstrations, we perform eye-to-hand camera calibration to obtain calibrated third-person observations and normalize all coordinates with respect to the robot base frame.

B.2 Datasets

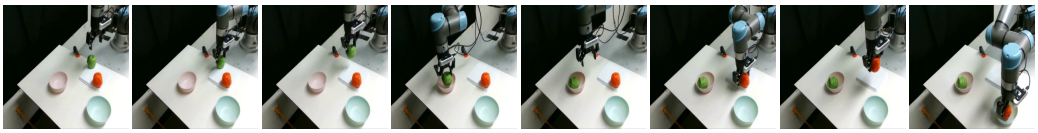
Base Tasks. We define four long-horizon manipulation tasks as base tasks. **(a) Fruit** requires the robot to place a green apple into a pink bowl and an orange into a green bowl. **(b) Cup & Plush** requires the robot to place a white cup on top of an upside-down pink cup, and then put a blue penguin plush toy into the white cup. **(c) Stack Cups** requires the robot to stack cups of different sizes in a specified order. **(d) Cabinet** requires the robot to open the cabinet, swap the object inside with the object outside, and then close the cabinet.



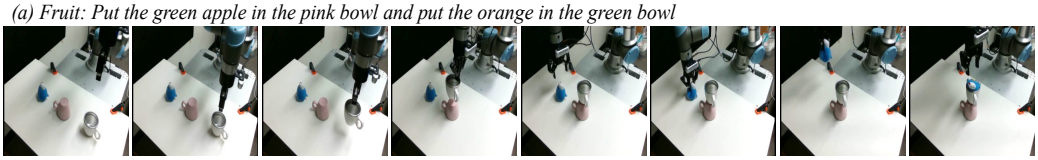
Figure 5: **Real-world deployment setup.**

Table 10: **Experimental protocol and hyperparameter configuration for Seer [22].**

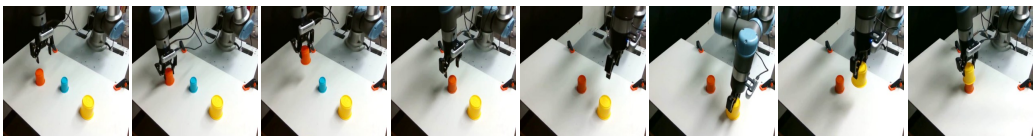
Category	Simulation	Real-world
Evaluation Trials	20 per task	15 per task
Training Demonstrations	50 per task	40 per task
Test-time Randomization	-	position perturbations
Batch Size	512	512
Learning Rate	1e-3	1e-3
LR Schedule	cosine	cosine
Prediction Length k	3	3
Historical Length L	7	7
Selected Points K	2048	2048
λ_{point}	0.005	0.05
λ_{tcp}	0.005	0.05



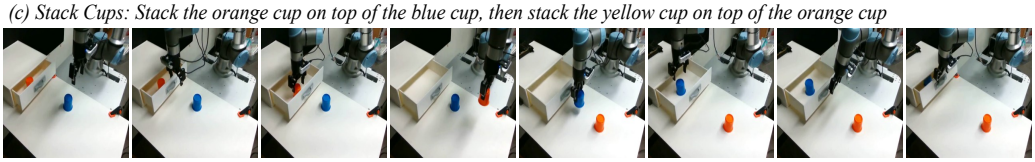
(a) Fruit: Put the green apple in the pink bowl and put the orange in the green bowl



(b) Cup & Plush: Pick up the white cup and place it on top of the upside-down pink cup, then pick up the blue penguin plush toy and put it in the white cup



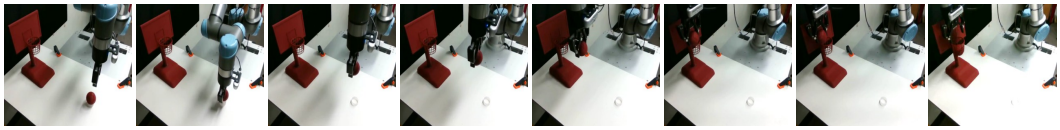
(c) Stack Cups: Stack the orange cup on top of the blue cup, then stack the yellow cup on top of the orange cup



(d) Cabinet: Open the drawer, take the orange cup out and put it on the table, then put the blue cup in the drawer and close the drawer

Figure 6: **Base task visualization.** Overview of the four long-horizon base tasks used in our real-world experiments: Fruit, Cup & Plush, Stack Cups, and Cabinet.

3D-Aware Tasks. We further design three 3D-aware tasks to assess whether the VLA model can capture action-relevant 3D dynamics and spatial relationships beyond conventional pick-and-place skills. **(a) Basketball** requires the robot to grasp a basketball and place it into a basketball hoop. **(b) Hang Cup** requires the robot to grasp a cup and accurately hang its handle on a cup holder. Unlike



(a) Basketball: Pick up the red ball and place it in the basketball hoop



(b) Hang Cup: Pick up the white mug and hang it on the wooden mug rack



(c) Ring Insertion: Put the blue ring on the wooden stand, then put the pink ring on the wooden stand

Figure 7: **3D-aware task visualization.** Overview of the three 3D-aware tasks used in our real-world experiments: Basketball, Ring Insertion, and Hang Cup.

the base tasks, these scenarios require fine-grained 3D understanding beyond basic pick-and-place manipulation. **(c) Ring insertion** requires the robot to pick up two rings and precisely insert them onto a vertical pillar by aligning the ring holes with the pillar. We additionally evaluate zero-shot robustness under *height* and *viewpoint variations*, where the model is tested under shifted object heights and camera viewpoints without additional fine-tuning.

B.3 Further Analysis

Component Analysis. We further conduct a component analysis under the *viewpoint* variation setting. Removing coordinate alignment eliminates the robot-centric alignment of the 3D targets, while removing magnitude-based selection replaces motion-salient point selection with random sampling. We also ablate the point-level and TCP-level losses, which supervise scene-level motion and control-relevant motion, respectively. As shown in Tab. 11, all components contribute to robustness under viewpoint variation. In particular, removing magnitude-based selection or the TCP-level loss reduces the success counts from 7/15 to 0/15 and 2/15, respectively.

Table 11: **Component analysis under viewpoint variation.**

Method	Success
FlowVLA	**7/15**
– w/o coordinate alignment	4/15
– w/o magnitude-based selection	0/15
– w/o TCP-level loss	2/15
– w/o point-level loss	4/15

Real-World Deployment with $\pi_{0.5}$. We further evaluate FlowVLA with the strong baseline [4] on real-world 3D-aware tasks. As shown in Tab. 12, FlowVLA improves the success rate on both Basketball and Hang Cup, from 10/15 to 13/15 and from 9/15 to 14/15, respectively. We additionally evaluate robustness to a dynamic visual environment by introducing a moving background during deployment in the Basketball task. Under this test-time dynamic-background variation, FlowVLA improves the success rate from 5/15 to 8/15, demonstrating improved robustness to dynamic scene changes beyond the static training environment.

Table 12: **Real-world results with $\pi_{0.5}$.** Evaluation on 3D-aware tasks and test-time dynamic-background variation.

Method	Basketball	Hang Cup	Background
$\pi_{0.5}$	10/15	9/15	5/15
$\pi_{0.5}$ **+ FlowVLA**	13/15	14/15	8/15

C Qualitative Results

C.1 3D Point Composition Analysis

Fig. 8 illustrates how our motion-based selection and TCP aggregation change the composition of 3D points in LIBERO-Spatial. Here, object points include robot components and scene objects, whereas non-object points correspond to the background. (**Panel-❶**) shows the 3D points labeled in the MuJoCo simulator, comprising 50k points. (**Panel-❷**) shows that the proportion of 3D object points, such as grippers and robot arms, becomes substantially higher than that of non-object points, such as the background, after applying our first-stage motion-based selection. We also demonstrate that the second selection stage in the TCP branch further filters out non-object 3D points during the attention operation, as shown in (**Panel-❸**). These results illustrate that motion-based selection suppresses largely static background regions, while TCP aggregation further emphasizes points associated with robot-object interactions.

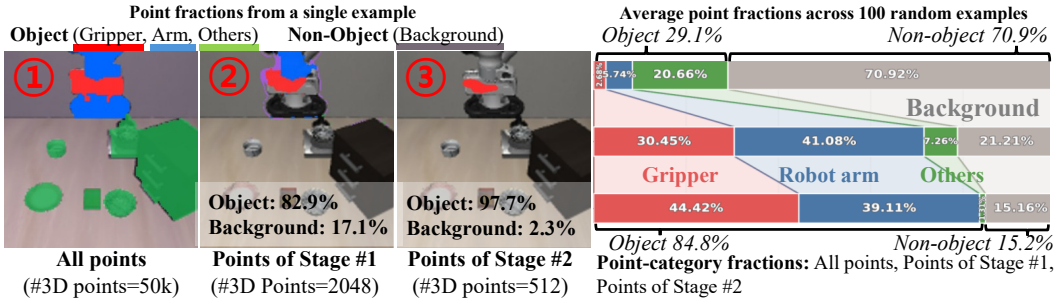


Figure 8: **3D point composition analysis.** Motion-based selection increases the proportion of object points, while TCP aggregation further reduces background points.

C.2 VDPM Visualization

Fig. 9 and Fig. 10 show qualitative visualizations of VDPM [16] 3D point trajectories in LIBERO simulation and real-world scenarios. For visual clarity, we display all reconstructed points in this analysis, whereas FlowVLA uses only the top-K dynamically informative points selected by the flow magnitude during training. The visualization confirms that VDPM can recover temporally consistent 3D point motion from RGB observation sequences. The resulting trajectories highlight motion-salient physical changes, including object displacement and interaction regions around the robot end-effector. These results provide qualitative evidence that reconstruction-derived 3D point trajectories can serve as a useful training-only supervision signal for FlowVLA.

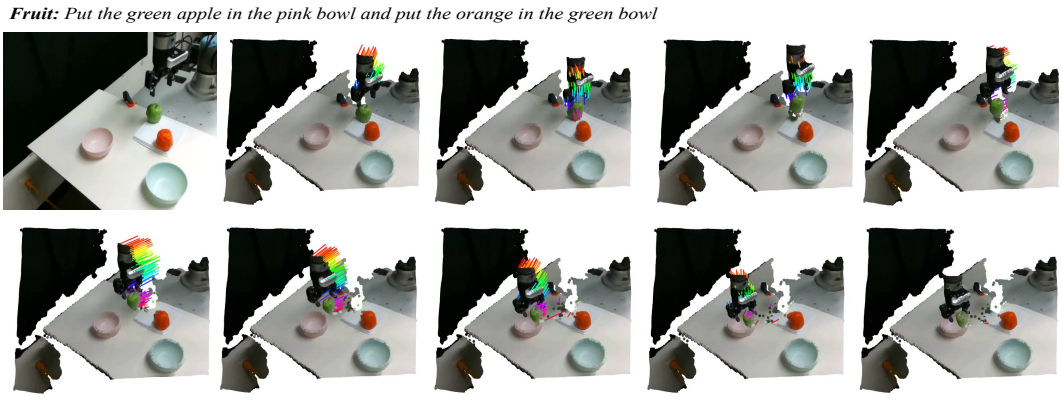


Figure 9: **VDPM 3D point trajectory visualization** on the real-world Fruit task.

LIBERO 10: *Put the black bowl in the bottom drawer of the cabinet and close it*

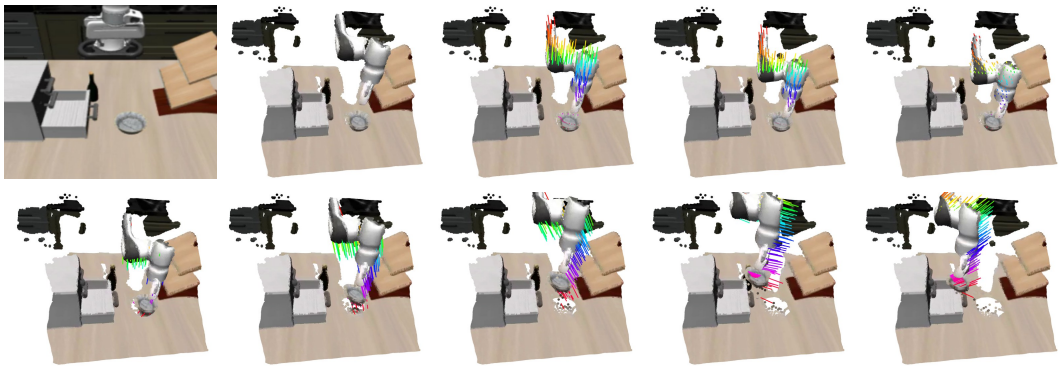


Figure 10: **VDPM 3D point trajectory visualization** on the LIBERO-10 task.

Selected Point Visualization. We provide a qualitative analysis of the proposed magnitude-based point selection by visualizing the selected K points used for training-time flow supervision. As shown in Fig. 11, the selected points consistently appear around regions undergoing physical motion, such as the TCP, the falling object (the juice in the first frame), and the object interacting with the TCP (the basket in the sixth frame). In contrast, points on static backgrounds and reconstruction-induced noisy regions are largely filtered out. These suggest that motion magnitude serves as a simple yet effective criterion for extracting motion-salient 3D points from reconstruction-derived trajectories, which often coincide with action-relevant regions. Importantly, this selection does not require semantic masks, object annotations, or contact labels, making it a scalable mechanism for constructing 3D dynamics supervision from RGB-only demonstrations.

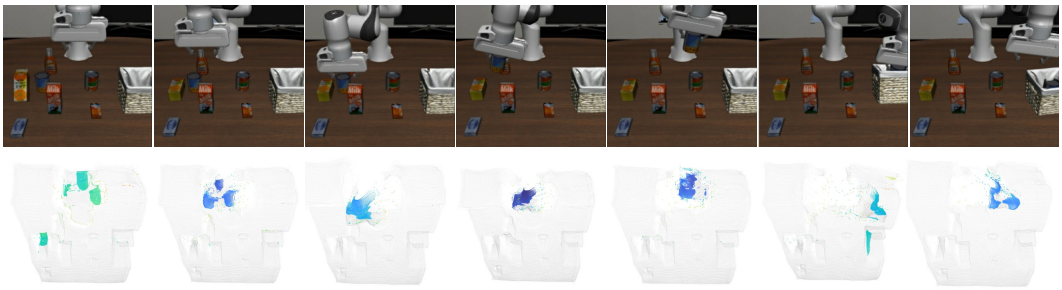
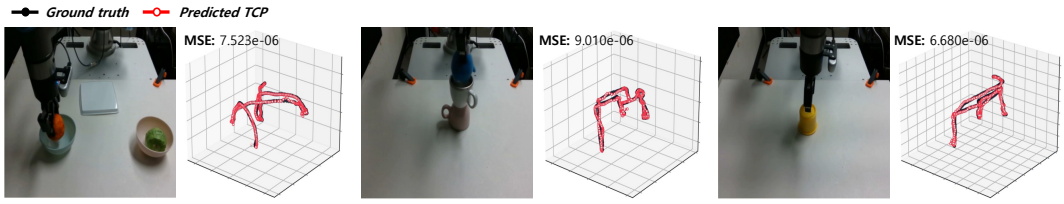


Figure 11: **Qualitative visualization** of magnitude-based point selection.

TCP Trajectory Visualization. We also visualize the predicted TCP trajectories to assess whether FlowVLA can convert scene-level point motion into control-relevant dynamics. Specifically, we compare the TCP trajectories predicted by the TCP-level aggregation module with the ground-truth TCP trajectories from robot execution. As shown in Fig. 12, the predicted trajectories closely follow the ground-truth trajectories, exhibiting similar motion directions. This qualitative agreement supports our scene-to-control distillation design: the model learns to aggregate dense 3D point displacements into an end-effector-level motion representation. Therefore, the auxiliary TCP-level objective provides a direct bridge between reconstruction-derived scene dynamics and the robot control trajectory.

TCP-Level Aggregation Weight Visualization. We further visualize the learned aggregation weights of the TCP-level branch to examine how FlowVLA transfers scene-level 3D motion cues into control-relevant dynamics. As shown in Fig. 13, magnitude-based selection identifies K motion-salient 3D points from reconstruction-derived trajectories, and we visualize the top-512 points with the largest TCP-level aggregation weights. The TCP-level aggregation module further refines these scene-level motion cues by assigning larger weights to points around the robot TCP and object–robot interaction region. This indicates that the TCP-level branch learns to distill motion-salient 3D point



(a) Fruit *(b) Cup & Plush* *(c) Stack Cups*

Figure 12: **Qualitative visualization** of predicted TCP trajectory.

displacements into end-effector-relevant flow, rather than treating all selected points uniformly. This visualization further supports our scene-to-control distillation design by showing how the TCP-level branch extracts control-relevant motion cues from reconstruction-derived scene dynamics.

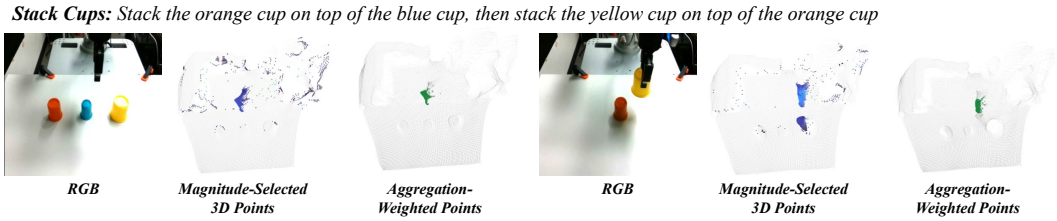


Figure 13: **Qualitative visualization** of TCP-level aggregation weights.

C.3 Failure Case Visualization

Fig. 14, Fig. 15, and Fig. 16 visualize representative failure cases in the real-world experiments. With limited demonstration data, the baseline frequently stops execution prematurely or places objects at incorrect locations under variations in the initial object configuration. These failures become more severe in 3D-aware tasks and **viewpoint variation**, often leading to collisions or emergency stops. In the **height variation** setting, the baseline also struggles to recognize shifted grasp positions, resulting in missed grasps or collisions with the table surface. FlowVLA substantially mitigates these failure modes. It shows no premature stopping behavior and performs precise manipulation in most 3D-aware scenarios, except for extreme cases where the target object lies near the camera boundary. FlowVLA also achieves more accurate grasping under **viewpoint** and **height variations**, significantly reducing collisions and insertion failures. These qualitative results suggest that training-only 3D flow supervision improves the 3D spatial awareness of VLA policies, even with limited demonstration.

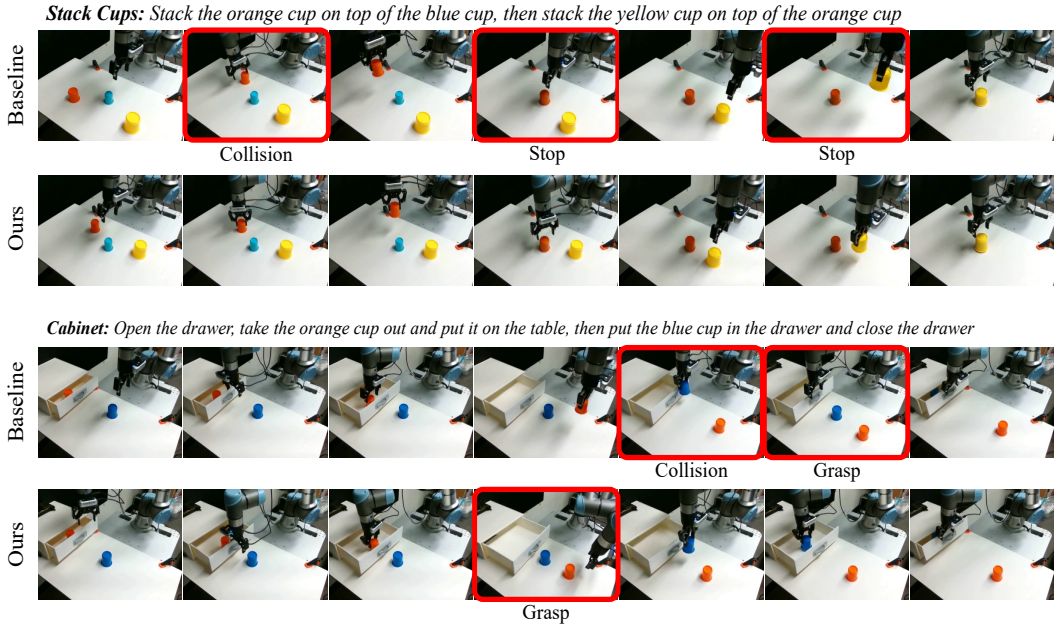


Figure 14: **Failure case visualization** on *Stack Cups* and *Cabinet*.

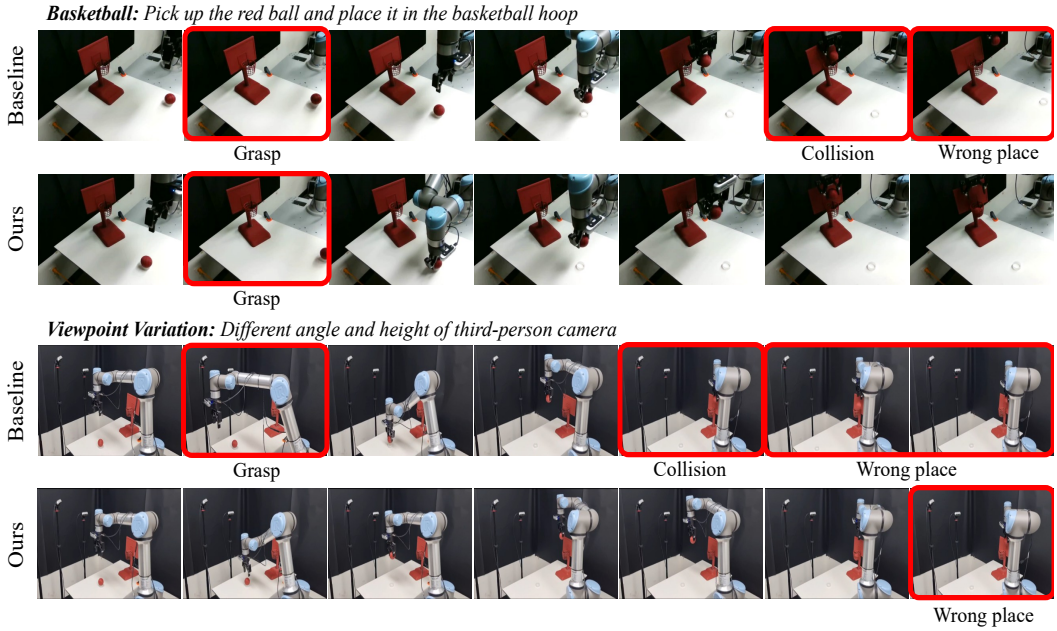


Figure 15: **Failure case visualization** on *Basketball* with **viewpoint variation**.

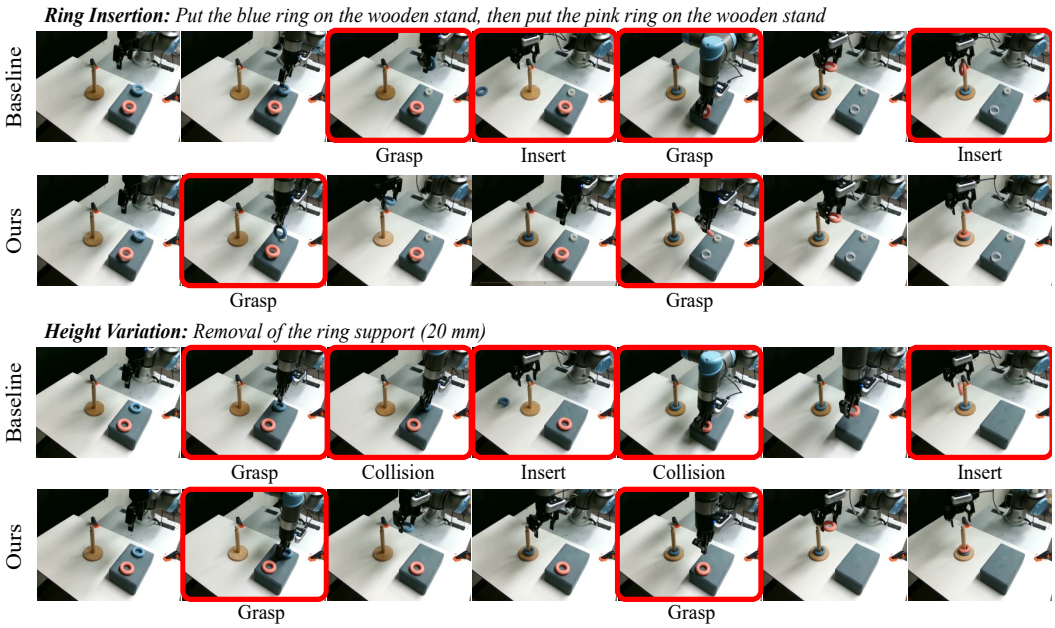


Figure 16: **Failure case visualization** on *Ring Insertion* with **height variation**.