RotVLA: Rotational Latent Action for Vision-Language-Action Model

Qiwei Li[1,2] **Xicheng Gong**[1] **Xinghang Li**[2] **Peiyan Li**[2,3]
Quanyun Zhou[2] **Hangjun Ye**[2] **Jiahuan Zhou**[1*] **Yadong Mu**[1*]

[1]Wangxuan Institute of Computer Technology, Peking University
[2]Xiaomi Robotics [3]CASIA

https://qiweili00.github.io/rotvla-page/

Abstract

Latent Action Models (LAMs) have emerged as an effective paradigm for handling heterogeneous datasets during Vision-Language-Action (VLA) model pretraining, offering a unified action space across embodiments. However, existing LAMs often rely on discrete quantization encode and decode pipelines, which can lead to trivial frame reconstruction behavior, limited representational capacity, and a lack of physically meaningful structure. We introduce **RotVLA**, a VLA framework built on a continuous rotational latent action representation. Latent actions are modeled as elements of $\mathrm{SO}(n)$, providing continuity, compositionality, and structured geometry aligned with real-world action dynamics. A triplet frame learning framework further enforces meaningful temporal dynamics while avoiding degeneration. RotVLA consists of a VLM backbone and a flow-matching action head, pretrained on large-scale cross-embodiment robotic datasets and human videos with latent-action supervision. For downstream robot control, the flow-matching head is extended into a unified action expert that jointly denoises latent and robot actions. Here, latent actions serve as a latent planner, providing high-level guidance that conditions action generation. With only 1.7B parameters and 1700+ hours of pretraining data, RotVLA achieves 98.2% on LIBERO and 89.6% / 88.5% on RoboTwin2.0 under clean and randomized settings, respectively. It also demonstrates strong real-world performance on manipulation tasks, consistently outperforming existing VLA models.

1 Introduction

Recently, Vision-Language-Action (VLA) [1, 2] models that are pretrained from Vision-Language Models (VLMs) [3–5] using large-scale robot-related data have demonstrated substantial potential. However, VLA pretraining faces a fundamental challenge: effectively handling heterogeneous data collected from diverse robot embodiments and unlabeled human videos. A promising direction to address these issues is Latent Action Models (LAMs) [6–8], which aim to learn a unified action space shared across datasets. Most existing LAMs adopt an encode and decode paradigm (Fig.2(a)), and typically apply quantization methods such as VQ-VAE [9] to the latent action space. This constrains the space and makes it compatible with the next-token prediction training paradigm of large VLMs. However, this paradigm suffers from three key limitations: (1) LAMs have the risk of degenerating into trivial solutions that simply encode the target frame [10]. (2) Discretizing the latent action space severely restricts its representational capacity and contradicts the inherently continuous nature of

*Corresponding Author

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

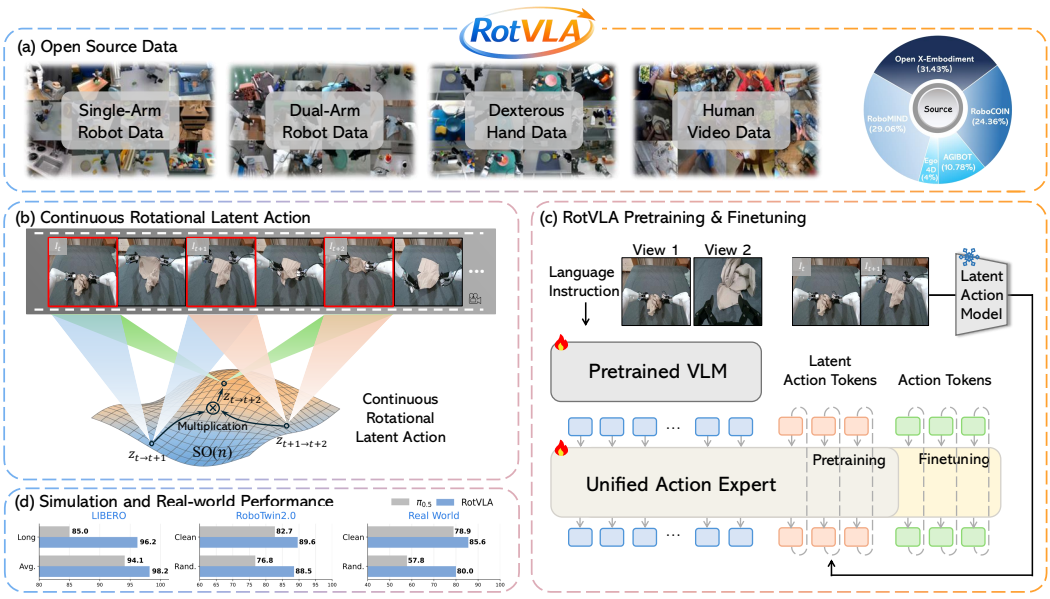


Figure 1: We introduce RotVLA, a Vision-Language-Action framework pretrained with a continuous rotational latent action on over 1700 hours of cross-embodiment robot and human video data. The latent actions are trained via triplet learning framework and further integrated into a unified flow-matching action expert, where they serve as high-level planners guiding robot control. Results demonstrate strong performance across simulation benchmarks and real-world manipulation tasks.

physical actions. (3) The latent action space lacks physical meaning, providing no notion of scale or meaningful action composition.

To address these issues, we propose **RotVLA**, a VLA framework built upon a continuous rotational latent action representation. As shown in Fig.1, instead of modeling latent actions as discrete tokens, we represent them as elements of the n-dimensional rotation group, $\mathrm{SO}(n)$. This formulation naturally aligns with the continuous nature of robot actions and provides substantially richer representational capacity compared to discrete tokens. Moreover, due to the closure property of $\mathrm{SO}(n)$, latent-action composition can be defined via matrix multiplication, mirroring real-world action composition.

Leveraging this representation, we formulate latent action learning as transitions across three frames (Fig. 2(b)). Given a frame triplet, we apply standard encoding and decoding to each pair of consecutive frames and adopt a soft quantization mechanism [11] that constrains the latent space while preserving continuity. We then compose the latent actions from the first and second transitions to infer the latent action from the first to the third frame, and supervise the prediction. This objective encourages the encoder to capture frame-to-frame dynamics rather than relying on direct reconstruction shortcuts.

After pretraining the latent action model, any pair of consecutive video frames can be mapped into a continuous $\mathrm{SO}(n)$ latent action. Building on this, we construct RotVLA, which integrates a pretrained VLM backbone with a flow-matching latent action head. RotVLA is pretrained to predict latent actions using large-scale human and cross-embodiment robot datasets.

To adapt RotVLA to downstream manipulation tasks, we extend the latent action head into a unified flow-matching action head that simultaneously denoises latent actions and robot actions. Through causal attention, robot-action generation is conditioned on latent actions, allowing the latent space to function as the planner that guides action prediction.

Our key contributions are summarized as follows:

(1) We propose a continuous rotational formulation of latent action based on $\mathrm{SO}(n)$, which captures the continuous and compositional properties of real-world actions. The latent action is learned from the dynamics across three consecutive frames and provides a unified action representation that generalizes across embodiments with zero-shot action generation capabilities.

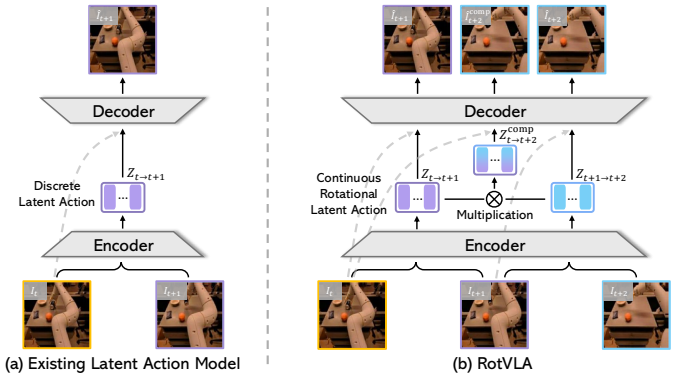


Figure 2: Illustration of existing LAMs (a) and RotVLA (b). Existing LAMs follow an encode–decode paradigm and model latent actions as discrete tokens. In contrast, RotVLA represents latent actions as continuous elements of $\mathrm{SO}(n)$. Additionally, we introduce a triplet learning framework that not only learns latent actions between consecutive frames but also composes these actions to predict future frames, reinforcing the capture of meaningful temporal dynamics.

(2) We introduce RotVLA, a VLA pretrained on large-scale human and cross-embodiment robot datasets using a flow-matching latent action expert. For fine-tuning on downstream tasks, we design a unified action expert that treats latent action as a planner to guide robot action prediction.

(3) With only 1.7B parameters, RotVLA achieves 98.2% on LIBERO [12] and 89.6% / 88.5% on RoboTwin2.0 [13] benchmark. Real-world experiments on tasks such as pick-and-place and cup stacking demonstrate improvements over existing VLA methods.

2 Related Work

2.1 Vision-Language-Action Model

With the rapid progress of Vision-Language Models (VLMs) [3–5], multimodal systems have demonstrated strong capabilities in perception and semantic understanding across both images and text. VLMs provide a unified embedding space that aligns visual observations with linguistic concepts, enabling powerful generalization, instruction following, and cross-modal reasoning. Building upon this foundation, Vision-Language-Action (VLA) models [1, 2, 14] extend VLMs into the action domain by integrating control within the multimodal framework.

Typically, VLA systems adopt a two-stage paradigm in which a pretrained VLM is first adapted to robot-specific multimodal data, followed by fine-tuning on target environments or tasks. Some prior works [7, 15–17] follow a next-token prediction formulation, where continuous actions are discretized into bins or encoded as action tokens, allowing pretraining and fine-tuning using standard language modeling techniques. To better preserve the continuous nature of actions, other approaches [18, 19] directly model actions as continuous values and train the model using regression objectives, achieving strong performance during fine-tuning. Beyond token-based and regression-based formulations, a growing body of research [8, 20–26] explores generative modeling techniques for action prediction. These methods employ diffusion-based or flow-matching policies that treat action generation as a conditional generative process, producing smooth and coherent action sequences.

2.2 Latent Action Model

Most VLA methods rely on large-scale, robot-specific annotated datasets for pretraining. However, learning from cross-embodiment datasets remains challenging due to substantial variations in visual observations, action spaces, and embodiment configurations [26]. Furthermore, these approaches rely on ground-truth action annotations, which limits their ability to leverage large-scale Internet video data. To address these issues, a line of work focuses on Latent Action Models, which aim to learn a unified action space across heterogeneous datasets by encoding the transition between sequential observations. Mainstream approaches [6, 27–31] typically adopt an Inverse Dynamics Model (IDM)

to infer latent actions from consecutive video frames, coupled with a Forward Dynamics Model (FDM) that reconstructs future observations conditioned on the inferred latent action. Building on this formulation, [32, 33] incorporate annotated actions to guide latent-action learning, while [22, 34] treat latent actions as surrogate labels for unlabeled data. Other works enhance these models by introducing additional modalities such as depth or optical flow [34–37].

Despite their promise, these pipelines have the risk of degenerating into trivial solutions that simply encode and reconstruct the future frame [10]. Existing methods attempt to mitigate this issue by limiting the latent action space by quantization, thereby encouraging the model to capture the underlying dynamics between frames. However, such discretization contradicts the inherently continuous and compositional nature of actions. To address this, we introduce a continuous latent action representation based on elements of $\mathrm{SO}(n)$, together with a triplet learning framework that explicitly enforces the latent action to model the underlying dynamics between frames.

3 RotVLA

3.1 Stage I: Continuous Rotational Latent Action

Preliminary. A typical LAM consists of an inverse dynamics encoder $\mathcal{E}$ and a forward dynamics decoder $\mathcal{D}$. Given two consecutive video frames with k interval, denoted as I_t and I_{t+1}, the encoder extracts a latent action representation composed of N tokens of dimension d, i.e. $z_{t\to t+1} \in \mathbb{R}^{N\times d}$. In conventional LAMs, this latent action is obtained through vector quantization methods VQ-VAE [9]:

$$z_{t\to t+1} = VQ\big(\mathcal{E}(I_t, I_{t+1})\big). \tag{1}$$

The decoder then predicts the next frame conditioned on the latent action and the current frame:

$$\hat{I}_{t+1} = \mathcal{D}(I_t, z_{t\to t+1}). \tag{2}$$

The LAM is supervised by minimizing the reconstruction error:

$$\mathcal{L}_{\mathrm{LAM}} = ||\hat{I}_{t+1} - I_{t+1}||_2. \tag{3}$$

Continuous latent action. As discussed earlier, existing LAMs suffer from discontinuities introduced by hard quantization, which limits their ability to model inherently continuous physical actions. To address this issue, we replace VQ-VAE [9] with SoftVQ [11], which aggregates multiple codewords through a soft categorical distribution. This mechanism preserves continuity in the latent space while retaining codebook structure. Details about SoftVQ are provided in the Appendix.

Rotational latent action. Building upon the continuous formulation, we further constrain the latent action to lie in the rotation group $\mathrm{SO}(n)$, yielding an $\mathrm{SO}(n)$-valued latent representation. Specifically, we represent each latent action as an $n \times n$ rotation matrix. We empirically find that directly enforcing the latent action to match the form of a rotation matrix overly constrains its representational capacity. Therefore, the model first predicts an unconstrained matrix, $M = U\Sigma V^\top$. We then obtain the closest rotation matrix in Frobenius norm through SVD-based orthogonal projection, defined as $\mathrm{Proj}(\cdot)$:

$$\mathrm{Proj}(M) = U\mathrm{diag}\big(1, 1, ..., \det(UV^\top)\big)V^\top. \tag{4}$$

This projection ensures that the latent action satisfies orthogonality and unit determinant constraints while maintaining expressive flexibility during learning. For notational simplicity, we define the latent action extraction operator:

$$\mathcal{F}(I_t, I_{t+1}) = \mathrm{Proj}\Big(VQ_{\mathrm{soft}}\big(\mathcal{E}(I_t, I_{t+1})\big)\Big), \tag{5}$$

where VQ_{soft} represents the SoftVQ quantization. Thus, the LAM learning with continuous rotational latent action with frame I_t, I_{t+1} in Eq.1 can be rewritten as:$z_{t\to t+1} = \mathcal{F}(I_t, I_{t+1})$.

Triplet learning framework. To prevent degeneration into trivial frame reconstruction solutions, we introduce a triplet learning framework for latent action modeling. Given three consecutive frames sampled with interval k, denoted as I_t, I_{t+1} and I_{t+2}, the framework extends the standard two-frame LAM objective by explicitly enforcing temporal compositionality across two transitions. We first extract the corresponding latent actions using the encoder:

$$z_{t\to t+1} = \mathcal{F}(I_t, I_{t+1}), \quad z_{t+1\to t+2} = \mathcal{F}(I_{t+1}, I_{t+2}). \tag{6}$$

Following standard LAM training, single-step predictions are supervised via reconstruction losses:

$$\hat{I}_{t+1} = \mathcal{D}(I_t, z_{t\to t+1}), \quad \hat{I}_{t+2} = \mathcal{D}(I_{t+1}, z_{t+1\to t+2}), \tag{7}$$

$$\mathcal{L}_{\text{single}} = \|\hat{I}_{t+1} - I_{t+1}\|_2^2 + \|\hat{I}_{t+2} - I_{t+2}\|_2^2. \tag{8}$$

The latent action space admits a gauge ambiguity: without additional constraints, the identity transformation is not guaranteed to coincide with the canonical identity element of SO(n). We therefore explicitly anchor the identity latent action using stable frame pairs. Concretely, we compute the batch-wise mean latent action from identical frame pairs and project it onto $\mathrm{SO}(n)$ to obtain an identity element $z_{\mathcal{I}}$:

$$z_{\mathcal{I}} = \mathrm{Proj}\Big(\mathbb{E}\big(\mathcal{F}(I_t, I_t)\big)\Big). \tag{9}$$

Since each action lies on $\mathrm{SO}(n)$, action composition naturally corresponds to matrix multiplication. The composed two-step latent action is defined as:

$$z^{\text{comp}}_{t\to t+2} = z_{t+1\to t+2} \cdot z_{\mathcal{I}}^{-1} \cdot z_{t\to t+1}. \tag{10}$$

We then generate a two-step prediction by applying the composed action to the first frame:

$$\hat{I}^{\text{comp}}_{t+2} = \mathcal{D}(I_t, z^{\text{comp}}_{t\to t+2}). \tag{11}$$

A reconstruction loss penalizes the difference between this prediction and the ground-truth frame I_{t+2} as $\mathcal{L}_{\text{comp}} = \|\hat{I}^{\text{comp}}_{t+2} - I_{t+2}\|_2^2$. The overall training loss:

$$\mathcal{L}_{\text{triplet}} = \mathcal{L}_{\text{single}} + \mathcal{L}_{\text{comp}} + \mathcal{L}_{\text{soft}}, \tag{12}$$

where $\mathcal{L}_{\text{soft}}$ is the Kullback-Leibler loss in SoftVQ [11] for codebook learning.

The objective $\mathcal{L}_{\text{triplet}}$ not only promotes accurate single step frame reconstruction through $\mathcal{L}_{\text{single}}$, but also enforces structural consistency under action composition via $\mathcal{L}_{\text{comp}}$. By enforcing that the composition of $z_{t\to t+1}$ and $z_{t+1\to t+2}$ reproduce I_{t+2}, the model is explicitly penalized if the latent actions degenerate to trivial solutions that only reconstruct the immediate next frame. In this way, the triplet learning framework promotes the learning of meaningful dynamic representations that capture temporal transitions.

3.2 Stage II: RotVLA Pretraining

After learning the continuous rotational latent action space, the latent actions are used as supervision for training the RotVLA. Following prior VLA frameworks [22, 26], we construct RotVLA by combining a pretrained VLM backbone with a diffusion transformer (DiT) based action expert. We adopt InternVL3.5-1B [38] as the VLM model. The overall architecture of RotVLA is illustrated in Fig.1. Given a pair of consecutive frames (I_t, I_{t+1}) along with language instruction, the VLM processes this multimodal input and produces the corresponding representation of vision and language tokens, h. Meanwhile, the corresponding latent action tokens $z_{t\to t+1} \in \mathbb{R}^{n\times n}$ are obtained via Eq.6. These latent actions are implemented as an n-dimensional action chunk with horizon n. RotVLA is then pretrained using a flow-matching objective [21, 26]. Specifically, the action expert predicts a velocity field, $v_\theta(z_\tau, \tau, h)$, conditioned on the VLM features h, where $z_\tau = \tau z_{t\to t+1} + (1-\tau) z_0$, $z_0 \sim \mathcal{N}(0, \mathrm{I})$, $\tau \in [0, 1]$ is the time index. Flow-matching minimizes the squared error between predicted and true velocities:

$$\mathcal{L}^{\text{FM}}_{\text{LA}} = \mathbb{E}_{\tau, z_{t\to t+1}, z_0}\left[\|v_\theta(z_\tau, \tau, h) - (z_{t\to t+1} - z_0)\|_2^2\right]. \tag{13}$$

3.3 Stage III: RotVLA Finetuning

After the pretraining stage, RotVLA has been empowered with embodiment knowledge, by learning a multimodal model that maps actions from diverse robotic embodiments into a unified latent-action space. To adapt the pretrained model to real robotic manipulation, we introduce a unified flow-matching action expert that jointly predicts: (1) latent actions $z \in \mathrm{SO}(n)$, serving as a latent planner; (2) d-dimensional robot action chunk with horizon N, $a \in \mathbb{R}^{N\times d}$, serving as robot controller. Following the procedure described in Sec.3.2, we extract latent actions and define a joint action

Table 1: Evaluation on simulation benchmarks (LIBERO and RoboTwin2.0). * denotes models pretrained with latent actions, while † denotes models pretrained with both latent and robot actions.

Methods	Size	LIBERO					RoboTwin2.0	
		Spatial	Object	Goal	Long	Avg.	Clean	Rand.
GO-1† [8]	2B	96.2	97.8	96.0	89.2	94.8	37.8	36.2
CoMo* [31]	1B	80.3	95.0	85.0	63.0	80.8	-	-
OpenVLA [15]	7B	84.7	88.4	79.2	53.7	76.5	-	-
OpenVLA-OFT [19]	7B	97.6	98.4	97.9	94.5	97.1	-	-
π_0 [43]	3B	96.8	98.8	95.8	85.2	94.2	-	-
$\pi_{0.5}$ [21]	3B	**98.8**	98.2	98.0	92.4	96.9	82.7	76.8
GR00T-N1.6† [22]	3B	97.7	98.5	97.5	94.4	97.0	-	-
UniVLA* [7]	9B	95.4	98.8	93.6	94.0	95.5	-	-
villa-X† [32]	3B	97.5	97.0	91.5	74.5	90.1	-	-
X-VLA [26]	0.9B	98.2	98.6	97.6	**97.8**	98.1	72.8	72.8
StarVLA [44]	4B	97.8	98.6	96.2	93.8	96.6	88.2	88.3
Motus† [34]	8B	-	-	-	**97.8**	-	88.7	87.0
LingBot-VLA [45]	4B	-	-	-	-	-	88.6	86.7
RotVLA	1.7B	98.2	**99.6**	**98.4**	96.4	**98.2**	**89.6**	**88.5**

variable, $x = (a, z_{t\to t+1})$. We then apply a unified flow-matching objective to jointly supervise latent and robot actions:

$$\mathcal{L}^{\mathrm{FM}}_{\mathrm{LA-RA}} = \mathbb{E}_{\tau,x,x_0}\left[\|v_\theta(x_\tau,\tau,h) - (x - x_0)\|_2^2\right], \tag{14}$$

where $x_\tau = \tau x + (1-\tau)x_0, x_0 \sim \mathcal{N}(0,\mathrm{I}), \tau \in [0,1]$ is the time index. This formulation enables simultaneous denoising of latent planning variables and embodiment-specific control actions within a unified diffusion framework.

Structured attention mechanism. To ensure proper information flow between perception, latent planning, and robot control, we introduce a structured attention mechanism within the action head. Specifically, latent action tokens attend only to vision-language tokens, without attending to real action tokens aligning with the pretraining stage. While the robot action tokens attend to both latent action tokens and vision–language tokens. This structured attention enforces a clear separation between planning and control, while maintaining effective cross-modal interaction. By preserving latent action during finetuning, RotVLA retains the action semantics learned from cross-embodiment data, while enabling efficient adaptation to new robotic platforms and downstream tasks.

3.4 Data

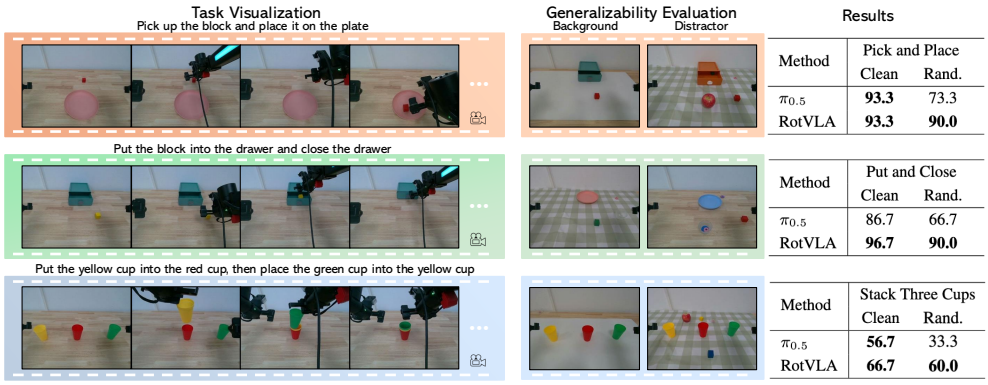
To construct the pretraining datasets for both LAM and RotVLA, we aggregate a large-scale dataset, including Open X-Embodiment [39], AGIBOT-beta [8], RoboMIND [40], RoboCOIN [41] and Ego4D [42], spanning diverse single-arm, bi-manual robots, dexterous hands and human video data over 1700 hours. Detailed dataset statistics are provided in the Appendix.

4 Experiment

4.1 Implementation Details

Model architecture. Following UniVLA [7], we implement the LAM encoder using a frozen DINOv2 [46] to extract frame-wise visual features. These features are further processed by a spatial–temporal transformer [47]. The decoder is implemented with standard transformer [48]. The latent action dimension n is set to 16. For RotVLA, the VLM backbone is initialized from InternVL3.5-1B [38], followed by an action expert implemented as a 24-layer Diffusion Transformer (DiT) [49]. The full model contains approximately 1.7B, including 304M in the vision encoder, 752M in the language model, 305M in the action head, and 290M in the latent action model.

Training details. We pretrain both LAM and RotVLA for 200k steps, with a batch size of 256. During downstream finetuning, RotVLA is trained with a batch size of 128. For robot action representation during finetuning, we adopt the absolute end-effector (abs EEF) pose formulation, consisting of xyz



Method	Pick and Place	
	Clean	Rand.
$\pi_{0.5}$	**93.3**	73.3
RotVLA	**93.3**	**90.0**

Method	Put and Close	
	Clean	Rand.
$\pi_{0.5}$	86.7	66.7
RotVLA	**96.7**	**90.0**

Method	Stack Three Cups	
	Clean	Rand.
$\pi_{0.5}$	**56.7**	33.3
RotVLA	**66.7**	**60.0**

Figure 3: Visualization of real-world tasks and generalization setting.

position and Rotate6D [50] representation for orientation, which aligns naturally with our rotational latent action formulation. The model is optimized using the AdamW optimizer with a learning rate of 1e-4 and weight decay of 0.01. During RotVLA pretraining and finetuning, we adopt the learning rate strategy proposed in X-VLA [26]: the VLM backbone is assigned a reduced learning rate to mitigate catastrophic forgetting, and the action expert is warmed up for the first 5k steps while keeping the VLM frozen. Pretraining is conducted on 8 NVIDIA H200 GPUs for 50 hours. Additional implementation details are provided in the Appendix.

4.2 Simulation Results

We evaluate RotVLA on two widely-used benchmarks: LIBERO [12] and RoboTwin2.0 [13].

LIBERO benchmark. The LIBERO benchmark [12] is a multi-task robotic manipulation suite designed to evaluate manipulation ability across diverse tasks and scene configurations. It includes four suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal and LIBERO-Long, each consists of 10 tasks. Following OpenVLA [15], we filter out unsuccessful trajectories from the training data. RotVLA is finetuned jointly on the four suites for 80k steps, yielding a single multi-task model capable of handling all tasks. As shown in Table.1, RotVLA achieves an average success rate of **98.2%**, surpassing other strong baselines such as UniVLA, GR00T-N1.6, and X-VLA.

RoboTwin2.0 benchmark. RoboTwin2.0 [13] is a large-scale simulation benchmark for evaluating bimanual robotic manipulation. The benchmark instantiates a suite of 50 dual-arm collaboration tasks. Following the evaluation protocol of [34, 45], we train a single model across all 50 tasks. Each task includes 50 clean demonstrations and 500 randomized demonstrations. During evaluation, we execute 100 rollouts per task under different random seeds. As reported in Table.1, RotVLA achieves a success rate of **89.6%** and **88.5%** on the clean and randomized setting, outperforming prior VLA baselines. Detailed results are provided in Appendix.

The strong performance of RotVLA can be attributed to the VLA pretraining strategy guided by the continuous rotational latent action representation, which captures more representative smooth and compositional dynamics. The triplet learning framework further mitigates collapse of LAM by enforcing compositional consistency, thereby encouraging the learning of meaningful temporal representations. In addition, the latent actions function as high-level planners that guide action generation, leading to more stable long-horizon manipulation and improved multi-task generalization.

4.3 Real World Experiment

We evaluate RotVLA on a dual-arm ARX R5 robotic platform across two single-arm tasks: (1) *pick up the block and place it on the plate*, and (2) *put the block into the drawer and close the drawer*, as well as one dual-arm task: (3) *stack three cups*. We collect 100 demonstration trajectories for each task and train methods in a multi-task setting. We compare our method against $\pi_{0.5}$ [21]. As shown in Fig.3, RotVLA consistently outperforms $\pi_{0.5}$ [21] across all three tasks under standard evaluation settings. In the two single-arm tasks, RotVLA achieves **over 90%** success rate. In the dual-arm cup stacking task, which requires bimanual motion and long-horizon planning, RotVLA achieves a substantially lower failure rate than the baseline. Under domain-shift conditions with

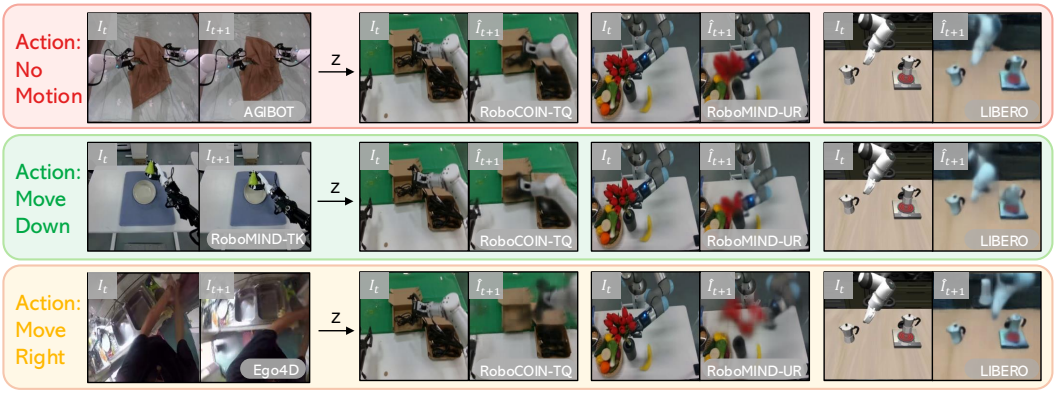


Figure 4: Illustration that the latent action extracted by one dataset can generalize to other seen and unseen datasets. RoboMIND-TK is short for Tien Kung robot in RoboMIND [40]. RoboCOIN-TQ is short for Tian Qin robot in RoboCOIN [41].

LAM Training	$\widehat{\text{MSE}}$	$\widehat{\text{MSE}}'$	$\Delta \uparrow$
Recon-Only [7]	0.0029	0.0066	0.0037
Triplet (Ours)	0.0030	0.0078	**0.0048**

Table 2: Comparison of next-frame reconstruction and imagined-frame error. $\widehat{\text{MSE}}$ = MSE $(\hat{I}_{t+1}, I_{t+1})$, $\widehat{\text{MSE}}'$ = MSE $(\hat{I}'_{t+2}, I_{t+1})$.

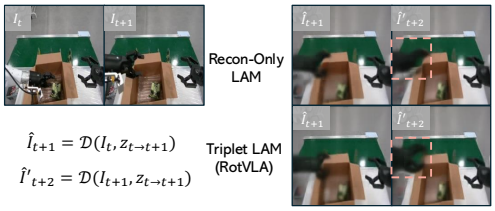


Figure 5: Visualization results of $\hat{I}_{t+1}$ and $\hat{I}'_{t+2}$.

varying backgrounds and the introduction of distractor objects, the performance gap becomes more pronounced. This robustness can be attributed to the continuous rotational latent action based pretraining and the latent planner finetuning, which captures high-level motion semantics rather than overfitting to background appearance. In terms of efficiency, on an NVIDIA H20 server, RotVLA achieves an inference latency of 79ms per step, compared to 61ms for $\pi_{0.5}$ [21], demonstrating competitive real-time performance.

4.4 Analysis of Continuous Rotational Latent Action

To demonstrate the effectiveness of the proposed LAM, we conduct cross-domain visualization experiments in Fig.4. Specifically, we extract latent actions from one domain and apply them to different domains. For example, in the third row, a latent action extracted from the Ego4D human dataset, representing a right motion, generalizes well to the robotic platforms such as Tian Qin and UR5, and even to unseen datasets LIBERO [12], which were not used during training. Similarly, latent actions corresponding to motions such as moving right or remaining stationary exhibit strong cross-domain transferability. These results suggest that the learned latent actions capture high-level motion semantics that are independent of specific embodiments or visual appearances.

Beyond visualization results, we further analyze the proposed latent action from multiple perspectives.

(1) Training stability. Did the triplet learning framework mitigate the risk of degrading? To answer this question, we compare the proposed triplet learning framework with a conventional reconstruction-only baseline. Given two frames I_t, I_{t+1} with interval k, we first extract the latent action, $z_{t\to t+1} = \mathcal{F}(I_t, I_{t+1})$ according to Eq.6. We then generate the reconstructed frame $\hat{I}_{t+1} = \mathcal{D}(I_t, z_{t\to t+1})$ and further apply the same latent action to I_{t+1} to obtain an imagined frame $\hat{I}'_{t+2} = \mathcal{D}(I_{t+1}, z_{t\to t+1})$. We compute the distances between I_{t+1} and both $\hat{I}_{t+1}$ and $\hat{I}'_{t+2}$. Ideally, $\hat{I}_{t+1}$ should closely match the ground-truth I_{t+1}, while $\hat{I}'_{t+2}$ should reflect the application of the learned motion to I_{t+1} and therefore differ from I_{t+1}. As shown in Table.2, both methods achieve low reconstruction error for I_{t+1}. However, the reconstruction-only baseline produces significantly smaller differences between I_{t+1} and $\hat{I}'_{t+2}$, indicating that the latent actions collapse to encoding

Table 3: Results of different latent action methods on LARY benchmark [51].

Model	Regression MSE↓					Classification Accuracy↑		
	CALVIN	VLABench	RoboCOIN	AgiBot	Avg.	Robot	Human	Avg.
LAPA [6]	0.96	0.95	0.96	1.00	0.97	14.61	23.64	19.13
UniVLA [7]	0.82	0.74	0.94	0.97	0.87	19.08	18.56	18.82
villa-X [32]	0.86	0.72	0.94	0.97	0.87	17.80	29.90	23.85
LAPA-DINOv3 [51]	0.50	0.25	0.82	0.84	0.63	64.19	27.04	45.62
RotVLA	**0.30**	**0.08**	**0.43**	**0.44**	**0.31**	**67.62**	**74.33**	**70.98**

frame appearance rather than true motion dynamics. Qualitative results in Fig.5 further illustrate this effect. The LAM trained with the triplet framework successfully captures the dynamic of lifting the left arm and can extrapolate the next step from I_{t+1}. In contrast, the reconstruction-only model generates a frame nearly identical to $\hat{I}_{t+1}$, suggesting that its latent action encodes appearance cues instead of motion information.

(2) Representation expressiveness. Is the proposed latent action more expressive than the previous methods? To answer this question, we compare different latent action methods on the LARY benchmark [51], which evaluates latent action representations via linear probing. Results in Table 3 show that RotVLA substantially outperforms existing methods, achieving both lower regression error for low-level robotic control and higher classification accuracy for high-level semantic actions. These results demonstrate that the proposed latent action representation captures both fine-grained control dynamics and high-level semantic structure more effectively than prior approaches.

4.5 Ablation Study

In Table.4, we conduct ablation studies to analyze key design choices in RotVLA, including the dimension of the latent action n, the triplet learning framework, the latent planner during finetuning.

Latent planner. We ablate the latent action supervision during finetuning. When latent action tokens are not preserved as a separate planning signal, performance drops, particularly on long-horizon tasks such as LIBERO-Long [12].

Dimension of the latent action. We vary the n across 8, 16 and 32. As shown in the last line, RotVLA with $n = 16$ achieves the best performance. Smaller dimensions limit dynamic expressiveness, while larger dimensions increase optimization difficulty and increased computational cost.

Table 4: Ablation results on different components.

Methods	Spatial	Object	Goal	Long	Avg.
w/o. Pretrain	94.8	97.8	94.2	89.8	94.2
w/o. Planner	96.0	**99.6**	97.2	93.2	96.5
$n = 8$	97.2	99.4	98.2	94.4	97.3
$n = 32$	**98.2**	**99.6**	97.4	94.0	97.3
Discrete	95.4	99.4	96.6	89.6	95.3
Cont.	96.6	99.4	97.2	93.6	96.7
Cont. $\mathrm{SO}(n)$	97.0	99.2	97.6	94.2	97.0
RotVLA	**98.2**	**99.6**	**98.4**	**96.4**	**98.2**

Latent Action Design. We ablate different latent action designs. The discrete latent action paradigm [7] brings only limited improvement over the w/o. Pretrain baseline due to the restricted representation capacity. Replacing it with continuous latent actions improves performance by 1.4%. Adopting the $\mathrm{SO}(n)$ latent action alone yields marginal gains, suggesting that the structural constraint itself is insufficient. However, when the compositional supervision of $\hat{I}_{t+2}^{\mathrm{comp}}$ is further introduced on top of the $\mathrm{SO}(n)$ latent action, the performance improves by an additional 1.2%. This result indicates that the triplet training framework encourages the LAM to capture motion dynamics rather than simply memorizing frame appearance.

5 Conclusion

We introduced RotVLA, a Vision-Language-Action framework built upon a continuous rotational latent action representation in $\mathrm{SO}(n)$. By replacing discrete latent tokens with structured rotations and introducing a triplet learning objective that enforces temporal compositionality, RotVLA learns meaningful and transferable motion dynamics while mitigating representation collapse. Built on a pretrained VLM with a flow-matching action expert, RotVLA treats latent actions as high-level planners that guide embodiment-specific control. Extensive experiments in simulation and real-world manipulation demonstrate strong generalization and competitive performance with 1.7B parameters. Our results suggest that incorporating geometric structure and continuity into latent action modeling is a promising direction toward scalable and generalizable robot foundation models.

Acknowledgments and Disclosure of Funding

This paper makes use of Open X-Embodiment, AGIBOT-beta, RoboMIND, RoboCOIN, Ego4D, LIBERO and RoboTwin. The authors confirm that the use of the aforementioned datasets are solely for academic research purposes and not for any commercial activities.

This work is supported by the National Key R&D Program of China (2024YFA1410000), Peking University Education Foundation (No. 050621_001) and the National NaturalScience Foundation of China (62376011).

References

[1] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

[2] Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025.

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[4] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[6] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[7] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[8] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[9] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[10] Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

[11] Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28358–28370, 2025.

[12] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[13] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[14] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024.

[15] 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. *arXiv preprint arXiv:2406.09246*, 2024.

[16] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025.

[17] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[18] Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *arXiv preprint arXiv:2509.09372*, 2025.

[19] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[20] 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*, 2025.

[21] 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, 2025. *URL https://arxiv. org/abs/2504.16054*, 1 (2):3.

[22] 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.

[23] Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. *arXiv preprint arXiv:2511.04555*, 2025.

[24] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[25] Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation. *arXiv preprint arXiv:2507.23523*, 2025.

[26] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[27] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[28] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024.

[29] Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *arXiv preprint arXiv:2412.04445*, 8, 2024.

[30] Zuolei Li, Xingyu Gao, Xiaofan Wang, and Jianlong Fu. Latbot: Distilling universal latent actions for vision-language-action models. *arXiv preprint arXiv:2511.23034*, 2025.

[31] Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025.

[32] Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

[33] Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos. *arXiv preprint arXiv:2601.04061*, 2026.

[34] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

[35] Zhejia Cai, Yandan Yang, Xinyuan Chang, Shiyi Liang, Ronghan Chen, Feng Xiong, Mu Xu, and Ruqi Huang. Seeing space and motion: Enhancing latent actions with spatial and dynamic awareness for vla. *arXiv preprint arXiv:2509.26251*, 2025.

[36] Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. Laof: Robust latent action learning with optical flow constraints. *arXiv preprint arXiv:2511.16407*, 2025.

[37] Manish Kumar Govind, Dominick Reilly, Pu Wang, and Srijan Das. Unilact: Depth-aware rgb latent action learning for vision-language-action models. *arXiv preprint arXiv:2602.20231*, 2026.

[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[39] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya 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.

[40] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

[41] Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

[42] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[43] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[44] StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

[45] Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

[46] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[47] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

[48] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[49] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[50] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[51] Dujun Nie, Fengjiao Chen, Qi Lv, Jun Kuang, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. Lary: A latent action representation yielding benchmark for generalizable vision-to-action alignment. *arXiv preprint arXiv:2604.11689*, 2026.

[52] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[53] Realman Robotics. URL `https://realmanrobotics.com/`.

[54] Agilex Robotics. URL `https://global.agilex.ai/products/cobot-magic`.

[55] Galaxea. URL `https://galaxea-ai.com/products/R1-Lite`.

[56] Agibot. URL `https://www.agibot.com/products/G1`.

[57] Airbot. URL `https://airbots.online/mmk2`.

[58] Unitree Robotics. URL `https://www.unitree.com/g1/`.

[59] TQ-Artisan. URL `https://www.tqartisan.com/productDetails?type=A2`.

[60] UR5e Robotics. URL `https://www.universal-robots.com/products/ur5e/`.

[61] Franka Emika Panda Robotics. URL `https://franka.de/`.

[62] Tien Kung Robotics. URL `https://x-humanoid.com/`.

[63] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 44(10-11):1863–1891, 2025.

[64] 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. *arXiv preprint arXiv:2212.06817*, 2022.

[65] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[66] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[67] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *The International Journal of Robotics Research*, 44(4):592–606, 2025.

[68] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[69] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023.

[70] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. In *Conference on Robot Learning*, pages 2113–2133. PMLR, 2023.

[71] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. *arXiv preprint arXiv:2210.11435*, 2022.

[72] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *The International Journal of Robotics Research*, 44(10-11):1727–1742, 2025.

[73] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, et al. Train offline, test online: A real robot learning benchmark. *arXiv preprint arXiv:2306.00942*, 2023.

[74] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. *arXiv preprint arXiv:2210.01911*, 2022.

[75] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[76] Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[77] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. `https://sites.google.com/view/berkeley-ur5/home`.

[78] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. Clvr jaco play dataset, 2023. URL `https://github.com/clvrai/clvr_jaco_play_dataset`.

[79] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Conference on Robot Learning*, pages 1199–1210. PMLR, 2023.

[80] Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingxiao Huo, Wei Zhan, Masayoshi Tomizuka, and Mingyu Ding. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot, 2023.

[81] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multistage cable routing through hierarchical imitation learning. *IEEE Transactions on Robotics*, 40:1476–1491, 2024.

[82] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[83] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2): 4126–4133, 2022.

[84] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *arXiv preprint arXiv:2308.10901*, 2023.

[85] Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Jörn Vogel. Shared control templates for assistive robotics. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 1956–1962. IEEE, 2020.

[86] Kris Wu Ge Yan and Xiaolong Wang. ucsd kitchens Dataset. August 2023.

A Limitation and Future Work

Parameter scalability. RotVLA achieves state-of-the-art performance with 1.7B parameters, yet the overall model scale remains relatively moderate. In addition, the LAM model that provides supervision for the VLA model contains only 290M parameters, making it considerably smaller. Scaling both the LAM and VLA models may further improve performance.

Learning from human videos. Leveraging human videos has become a promising direction for addressing the scarcity of robotic data, and LAM provides an effective mechanism for generating supervision from large-scale unlabeled human videos. In RotVLA, human videos are used only as supplementary training data. Extending the framework to incorporate more human videos, or even pretraining solely on human videos, is an important direction for future work.

B Details of Continuous Latent Action

We replace the VQ-VAE with SoftVQ [11] to preserve continuity in the latent space while retaining codebook structure. SoftVQ impose a soft categorical distribution on the posterior as:

$$posterior : q_\phi(\mathrm{z}|\mathrm{x}) = \mathrm{Softmax}(-\frac{\|z - \mathcal{C}\|_2}{\tau}) \tag{15}$$

$$latent : \mathrm{z} = q_\phi(\mathrm{z}|\mathrm{x})\mathcal{C} \tag{16}$$

where $\mathcal{C}$ represents the codebook with K codewords, τ is the temperature parameter. When assuming the prior is a uniform distribution over the codebook:

$$\begin{aligned}
\mathcal{L}_{\mathrm{kl}} &= \big(q_\phi(\mathrm{z}) \| p(\mathrm{z})\big) && (17)\\
&= \int q_\phi(\mathrm{z}) \big(\log q_\phi(\mathrm{z}) - \log p(\mathrm{z})\big) \mathrm{dz} && (18)\\
&= H\big(q_\phi(\mathrm{z})\big) - H\big(q_\phi(\mathrm{z}), p(\mathrm{z})\big), && (19)
\end{aligned}$$

where $p(\mathrm{z})$ is the uniform piror that $p(\mathrm{z}) \sim \mathcal{U}(0, K)$. In practice, the $\mathcal{L}_{\mathrm{kl}}$ is calculated as:

$$\mathcal{L}_{\mathrm{kl}} = H\big(q_\phi(\mathrm{z})\big) - H\big(\mathbb{E}_{\mathrm{x}\sim p(\mathrm{x})}\, q_\phi(\mathrm{z}|\mathrm{x})\big), \tag{20}$$

which represents the $\mathcal{L}_{\mathrm{soft}}$ in Eq.13 in the main paper.

Table 5: Detailed hyperparameters of RotVLA

Hyperparameters	RotVLA		
	Stage I	**Stage II**	**Stage III**
Learning rate	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-4}
LR scheduler	Constant	Cosine	Cosine
Weight decay	0.01	0.001	0.001
Gradient clip	-	1.0	1.0
Optimizer	AdamW($\beta_1 = 0.9, \beta_2 = 0.999$)		
Warm-up steps	0	5k	5k
Training steps	200k	200k	80-120k
Batch size	256	256	128
Image Size	224	224	224
Data Augumentation	-	ColorJitter	
Trainable Parameters	0.20B	1.36B	1.36B
Total Parameters	0.29B	1.65B	1.65B

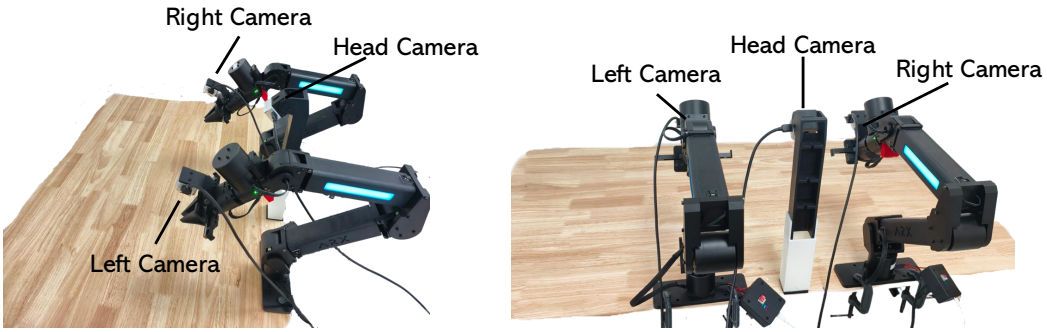


Figure 6: The robotic platform used in real-world experiments.

C Implementation Details

During stage I and II, we pretrain both LAM and RotVLA for 200k steps, with the batchsize of 256. During downstream finetuning, RotVLA is trained with a batch size of 128. The training steps for LIBERO, RoboTwin2.0 and real-world experiment is 80k, 120k and 40k respectively. The latent action model is frozen after stage I. Detailed hyperparameters are list in Table.5.

D Real-World Experiments

Real-world experiments are conducted on a dual-arm ARX R5 robotic platform equipped with three RealSense D405 cameras: one mounted on the head and two on the wrists. The detailed hardware setup is shown in Fig.6. We collect 30 minutes of teleoperation data per task at 30Hz. A single model is trained across all three tasks, conditioned on different language instructions.

We evaluate RotVLA on three tasks: (1) Pick up the block and place it on the plate. (2) Put the block into the drawer and close the drawer. (3) Put the yellow cup into the red cup, then place the green cup into the yellow cup. Qualitative results of RotVLA are provided in the `videos/` folder of the supplementary material. Each tasks is evaluated for 30 trials each with clean and randomized settings. For the randomized setting: 10 trials with background changes (two out-of-distribution backgrounds: plain white and patterned), 10 trials with distractor objects, and 10 trials with both background changes and distractors.

E More Visualization of Continuous Rotational Latent Action

To demonstrate the effectiveness of the proposed LAM, we visualize reconstruction results in Fig.7. The first line shows the sampled frame I_t, I_{t+1} and I_{t+2}, while the second line presents the corresponding generated frames $\hat{I}_{t+1}, \hat{I}_{t+2}$, and the composed prediction $\hat{I}^{\text{comp}}_{t+2}$, obtained by applying the composed latent action to I_t. The results indicate that the LAM effectively captures the dynamic evolution of the robot across frames. Importantly, the composed latent action successfully represents the two-step transition from I_t to I_{t+2}, demonstrating compositional consistency in the learned latent space.

F Detailed Results on RoboTwin2.0

In Table.6 we provide detailed results of RoboTwin2.0 benchmark [13] across 50 tasks. RotVLA achieves the success rate of **89.6% / 88.5%** on the clean and randomized settings.

G Data Statistic

We curate a training dataset totaling over 1700 hours from Open X-Embodiment [39], AGIBOT [8], RoboCOIN [41], RoboMIND [40], and Ego4D [42], consisting of dual-arm and single-arm robot

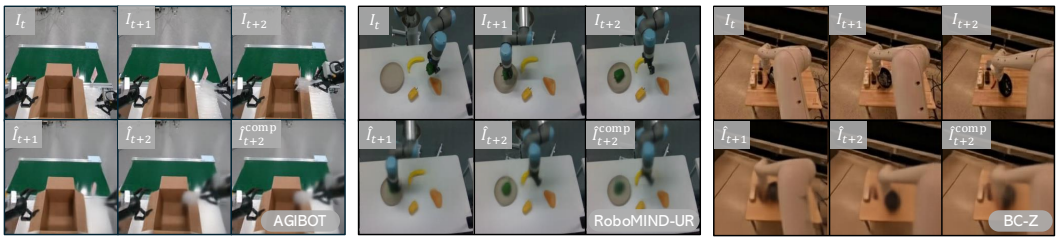


Figure 7: Illustration of reconstruction of $\hat{I}_{t+1}$, $\hat{I}_{t+2}$ and also the composed $\hat{I}^{\text{comp}}_{t+2}$ on AGIBOT [8], RoboMIND [40] and BC-Z [52] datasets.

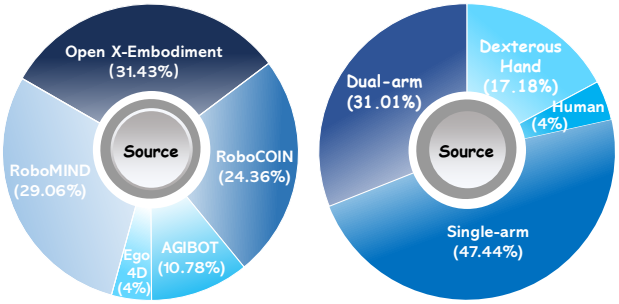


Figure 8: Statistics of the pretraining data used by RotVLA, grouped by dataset and embodiment type.

data, dexterous hands data as well as egocentric human videos. Detailed statistics of the different data sources are listed in Table.7. When grouped by dataset source (Fig.8(a)), the proportions are: Open X-Embodiment (31.43%), RoboMIND (29.06%), RoboCOIN (24.36%), AGIBOT (10.78%), and Ego4D (4.37%). When grouped by embodiment type (Fig.8(b)), the proportions are: dexterous hands (17.18%), human videos (4.37%), single-arm robots (47.44%), and dual-arm robots (31.01%).

H Data scalability

We analyze how RotVLA benefits from increasing the scale of pretraining data. When pretrained on only 40% of the robot data, RotVLA achieves a 3% improvement on LIBERO and over 7% on RoboTwin compared to training without pretraining. Expanding to the full robot datasets, including Open X-Embodiment [39], AGIBOT [8], RoboMIND [40], and RoboCOIN [41], further improves performance. Moreover, incorporating human video dataset, Ego4D [42], provides additional gains, particularly on RoboTwin2.0. These results demonstrate that scaling both the quantity and diversity of pretraining data consistently enhances performance, highlighting strong data scalability.

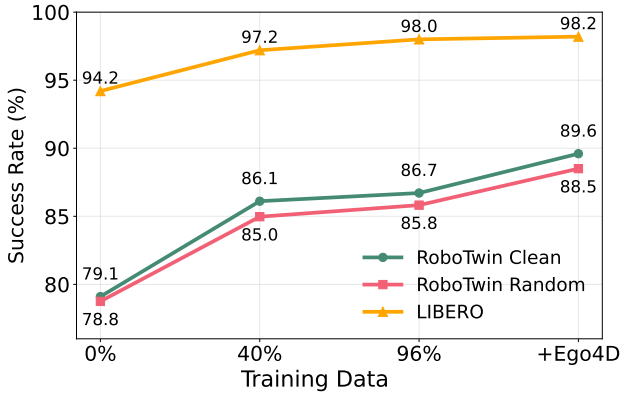


Figure 9: Impact of pretraining data scale on performance.

Table 6: Evaluation on RoboTwin 2.0 simulation benchmark.

Task	**Clean**	**Rand.**	**Task**	**Clean**	**Rand.**
Adjust Bottle	100%	100%	*Place Can Basket*	83%	76%
Beat Block Hammer	92%	85%	*Place Cans Plasticbox*	100%	100%
Blocks Ranking Rgb	98%	96%	*Place Container Plate*	99%	100%
Blocks Ranking Size	78%	77%	*Place Dual Shoes*	93%	96%
Click Alarmclock	47%	46%	*Place Empty Cup*	100%	99%
Click Bell	97%	92%	*Place Fan*	97%	97%
Dump Bin Bigbin	97%	98%	*Place Mouse Pad*	83%	87%
Grab Roller	100%	100%	*Place Object Basket*	91%	90%
Handover Block	95%	84%	*Place Object Scale*	92%	93%
Handover Mic	100%	99%	*Place Object Stand*	97%	100%
Hanging Mug	44%	52%	*Place Phone Stand*	94%	94%
Lift Pot	98%	100%	*Place Shoe*	99%	100%
Move Can Pot	95%	95%	*Press Stapler*	83%	81%
Move Pillbottle Pad	95%	99%	*Put Bottles Dustbin*	91%	88%
Move Playingcard Away	99%	99%	*Put Object Cabinet*	87%	82%
Move Stapler Pad	86%	84%	*Rotate Qrcode*	95%	95%
Open Laptop	98%	99%	*Scan Object*	88%	85%
Open Microwave	36%	12%	*Shake Bottle Horizontally*	99%	99%
Pick Diverse Bottles	88%	88%	*Shake Bottle*	100%	100%
Pick Dual Bottles	89%	88%	*Stack Blocks Three*	90%	86%
Place A2b Left	93%	95%	*Stack Blocks Two*	99%	99%
Place A2b Right	97%	96%	*Stack Bowls Three*	92%	88%
Place Bread Basket	96%	93%	*Stack Bowls Two*	97%	96%
Place Bread Skillet	89%	88%	*Stamp Seal*	81%	83%
Place Burger Fries	99%	98%	*Turn Switch*	46%	46%
Average				**89.6%**	**88.5%**

Table 7: The data mixture of RotVLA.

Source	Robot / Dataset	Norm	Type
Ego4D [42]	Human	4.37%	Human
RoboCOIN [41]	RMC-AIDA-L [53]	3.08%	Dual-Arm
	Agilex Split ALOHA [54]	1.94%	Dual-Arm
	Agilex Cobot Magic [54]	4.65%	Dual-Arm
	Galaxea R1 Lite [55]	3.75%	Dual-Arm
	AgiBot G1 [56]	2.13%	Dual-Arm
	Airbot MMK2 [57]	3.83%	Dexterous Hand
	Unitree G1edu-u3 [58]	2.55%	Dexterous Hand
	Tianqing A2 [59]	2.43%	Dual-Arm
RoboMIND [40]	UR5e [60]	8.22%	Single-Arm
	Franka Emika Panda [61]	7.79%	Single-Arm
	Tien Kung [62]	2.26%	Dual-Arm
	Tien Kung [62]	10.79%	Dexterous Hand
Open X-Embodiment [39]	BC-Z [52]	3.42%	Single-Arm
	Furniture Bench [63]	2.49%	Single-Arm
	Fractal [64]	5.92%	Single-Arm
	Kuka [65]	1.16%	Single-Arm
	DobbE [66]	0.71%	Single-Arm
	FMB Datasets [67]	3.60%	Single-Arm
	Bridge [68]	2.57%	Single-Arm
	UTAustin Mutex [69]	1.14%	Single-Arm
	Stanford Hydra Dataset [70]	2.27%	Single-Arm
	Austin Sailor Dataset [71]	1.12%	Single-Arm
	Austin Sirius Dataset [72]	0.89%	Single-Arm
	Toto [73]	0.93%	Single-Arm
	Taco Play [74]	1.35%	Single-Arm
	Roboturk [75]	1.07%	Single-Arm
	CMU Franka Pick-Insert [76]	0.46%	Single-Arm
	Berkeley Autolab UR5 [77]	0.56%	Single-Arm
	Jaco Play [78]	0.22%	Single-Arm
	Viola [79]	0.44%	Single-Arm
	Berkeley Fanuc Manipulation [80]	0.40%	Single-Arm
	Berkeley Cable Routing [81]	0.12%	Single-Arm
	NYU Franka Play [82]	0.33%	Single-Arm
	Austin Bud [83]	0.11%	Single-Arm
	CMU Stretch [84]	0.08%	Single-Arm
	DLR EDAN Shared Control [85]	0.03%	Single-Arm
	UCSD Kitchen [86]	0.03%	Single-Arm
AGIBOT [8]	AgiBot G1 [56]	10.78%	Dual-Arm
Total		**100.00%**	

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification:We discuss our contributions in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitation in the Appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The implementation details are provided in Sec. 4.1 and the Appendix.

   Guidelines:

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

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

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

   Answer: [No]

   Justification: The code will be made publicly available after the paper is accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental details are provided in Sec 4.1 and 4.2.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the high computational cost of training and evaluating, error bars are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The compute resources are provided in Sec. 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper complies with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We mention in the Conclusion Section. Our results suggest that incorporating geometric structure and continuity into latent action modeling is a promising direction toward scalable and generalizable robot foundation models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper is based on open-source datasets and models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code package or dataset.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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