PointACT: Vision-Language-Action Models with Multi-Scale Point-Action Interaction

Shizhe Chen, Paul Pacaud, Cordelia Schmid
Inria, École normale supérieure, CNRS, PSL Research University
https://cshizhe.github.io/projects/pointact.html

***Abstract*—Vision–Language–Action (VLA) models have shown strong potential for general-purpose robotic manipulation by leveraging large pretrained vision-language backbones. However, most existing VLAs rely primarily on 2D visual representations, which limits their ability to reason about fine-grained geometry and spatial grounding - capabilities that are essential for precise and robust manipulation in 3D environments. In this paper, we propose PointACT, a dual-system 3D-aware VLA policy that integrates hierarchical 3D point cloud representations directly into the action decoding process. PointACT employs a multi-scale point-action interaction mechanism with efficient bottleneck window self-attention, enabling evolving action tokens to densely attend to both local geometric detail and global scene structure. We evaluate PointACT on the LIBERO and RLBench benchmarks and systematically compare it against monolithic and dual-system VLA baselines, including variants augmented with point cloud inputs. PointACT achieves consistent improvements across both benchmarks, increasing success rates by 10% on the challenging RLBench-10Tasks suite over state-of-the-art pretrained VLAs, with even larger gains when the vision–language backbone is frozen and the action expert is trained from scratch. Extensive ablation studies demonstrate that tightly coupling hierarchical 3D geometry with pretrained 2D semantic representations is critical for robust and spatially grounded robot control. Our results also highlight the promise of pretrained 3D representations for 3D-aware VLA policies.**

I. Introduction

Developing general-purpose robots that can map natural language instructions to diverse physical actions remains a central challenge in robotics. Recently, Vision–Language–Action models (VLAs) [32, 6, 5, 13, 53] have emerged as a promising paradigm by building robotic control policies on top of large pretrained vision–language models (VLMs) [1, 3]. Hence, VLAs inherit a strong semantic understanding of the world from VLMs, enabling broader generalization across objects, scenes, and instructions than traditional task-specific policies.

However, a fundamental mismatch remains: the physical world is inherently three-dimensional, while most state-of-the-art VLAs rely on 2D image representations. The absence of explicit geometric input weakens spatial understanding and limits performance in precise and robust manipulation. As a result, current VLAs often depend on large-scale data to achieve viewpoint invariance and still struggle with tasks that require accurate spatial reasoning [63]. Recent VLA approaches [54, 47, 33, 66] attempt to recover 3D structure from single-view or sparse-view RGB images, either using external depth models [4] or jointly trained predictors, but such reconstruction remains inherently ambiguous and unreliable for high-accuracy control.

With the growing availability of 3D sensors such as RGB-D cameras, explicit geometric observations are increasingly accessible, yet how to effectively integrate them into VLAs remains an open challenge. Some 3D-aware VLAs incorporate depth or 3D positional cues as auxiliary signals layered onto 2D feature representations [38, 72], where action generation is still primarily driven by image tokens and geometric structure is weakly coupled to the final control output. Other approaches introduce 3D inputs into action experts [34, 60, 65], but typically rely on coarse-grained point cloud features, limiting fine-grained geometric interaction. Moreover, prior studies report limited gains from transferring pretrained 3D representations to policy learning [34, 68], partly due to the limited availability of in-domain pretrained point cloud encoders.

In this work, we propose **PointACT**, a 3D-enhanced VLA framework that combines a pretrained VLM backbone and a 3D action expert to integrate point clouds into action generation. Geometric observations are encoded using a strong point-cloud encoder based on pretrained Point Transformer v3 [64], which provides useful geometric features despite being trained on out-of-domain data. We systematically investigate different strategies for incorporating 3D information into VLAs, including injecting point tokens into the VLM backbone in monolithic VLAs and integrating geometry with action experts in dual-system VLA architectures, as illustrated in Figure 1(c). To foster action prediction grounded in 3D, we further propose a point-action expert that enables multi-scale interaction between point tokens and action tokens, allowing geometric cues to continuously condition the evolving action tokens while capturing both global structure and local detail. To ensure efficiency, action tokens serve as bottleneck queries that attend to point tokens through windowed attention, avoiding the computational cost of dense point token interactions.

We evaluate PointACT on the LIBERO and RLBench simulators across diverse tasks and action types, including keypoint pose prediction and delta action movements. Ablation studies show that injecting point tokens into the VLM backbone is less effective, as it can interfere with pretrained representations. In contrast, integrating point clouds within the action expert, particularly through fine-grained point–action interaction, produces consistent and substantial gains. We further find that initializing the point-cloud expert with a pretrained point encoder provides useful geometric priors and improves performance, despite domain gaps between large-

scale 3D pretraining data and tabletop manipulation scenes.

Our contributions in this work are threefold:

- We systematically study strategies to improve 3D spatial understanding of VLAs with point clouds, comparing backbone injection vs. action-expert integration.
- We propose PointACT, a dual-system VLA with a pretrained VLM backbone and a dedicated point-action expert that enables fine-grained point–action interaction through an efficient bottleneck windowed self-attention over multi-scale point cloud features.
- PointACT achieves consistent gains on LIBERO and RLBench benchmarks across action types, demonstrating the benefit of fine-grained geometry-driven action prediction.

II. Related Work

A. Learning-based Visuomotor Policies

Learning-based visuomotor policies have advanced rapidly with strong neural network architectures [71, 29, 7] and training objectives [15, 49]. While 2D visual policies benefit from powerful pretrained representations [55, 59], they often struggle with precise spatial reasoning due to the lack of explicit geometry grounding [10]. Depth-based methods partially address this limitation for grasping and planning [45, 2], but depth remains a 2.5D representation. To enable richer spatial reasoning, recent works explore explicit 3D representations such as voxels [28, 57] and point clouds [43, 11, 17, 68, 14] for robotic manipulation. SUGAR [12] further strengthens point-based policies through 3D representation pretraining with simulated data. Hybrid approaches fuse the semantic strength of 2D features with the geometric accuracy of 3D reasoning. HiveFormer [22] and RVT [20, 21] predict multiview 2D action heatmaps and lift them to 3D coordinates using depth, while Act3D [18] and 3D Diffuser Actor [30] lift pretrained 2D features into 3D volumes and learn fully 3D action policies. However, these visuomotor policies are typically trained and evaluated on narrow task suites, limiting generalization.

B. Vision-Language-Action Models (VLAs)

The success of LLMs [1] and VLMs [3] has inspired their use in robotic manipulation. Early works have explored zero-shot application of these foundation models for high-level planning [25], providing motion constraints [24], or code generation [39], but the performance remained limited without training on robot-specific datasets.

This motivates end-to-end VLAs which extend VLMs by adding an action modality and fine-tuning on robot datasets [48, 8]. RT-2 [76] and OpenVLA [32] use autoregressive next token prediction with discretized action tokens; Octo [19] and EO1 [53] introduce continuous action heads with diffusion training objectives [15]; OpenVLA-OFT [31] further studies VLA design and achieves strong performance and efficiency via simple regression. To enhance reasoning, several works incorporate Chain-of-Thought (CoT) [62]. ECoT [67] performs multi-step reasoning about plans, subtasks, motions, and visual grounding before predicting actions, while CoT-VLA [70] generates future image frames as visual goals. ThinkAct [23] further optimizes the reasoning process via reinforcement learning. However, relying on a single monolithic VLM for reasoning and control has raised concerns about inefficiency and catastrophic forgetting. Therefore, recent methods [6, 26, 5, 58, 73, 13, 44] such as π_0 [6] and GR00T [5] adopt a dual-system framework that decouples high-level perception and planning from low-level control. In these systems, the VLM is typically kept frozen, and its output representations are fed into a lightweight action expert for continuous action generation. Despite strong semantic capabilities, existing VLAs remain largely 2D-centric and are limited in 3D understanding and spatial generalization [75, 16, 52]. In this work, we address this by equipping VLAs with a 3D action expert for more accurate 3D action prediction.

C. 3D Spatial Reasoning in VLAs

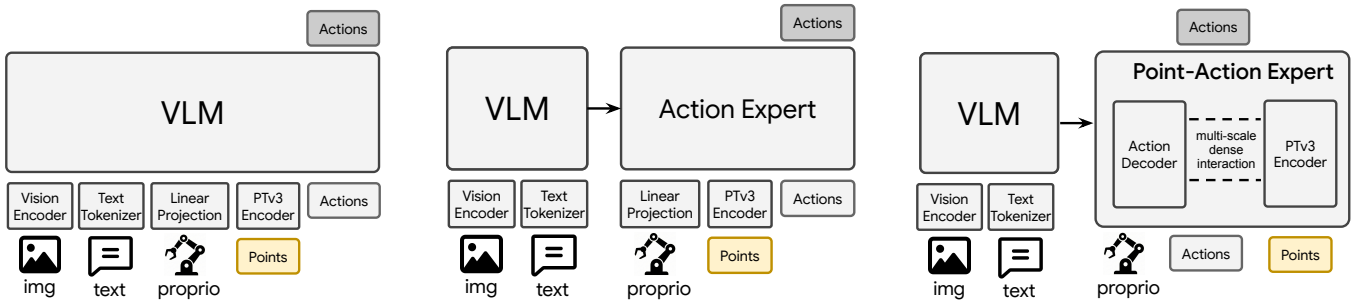
Recent VLAs increasingly incorporate 3D cues to improve spatial reasoning and generalization. One line of work augments RGB observations with predicted depth, such as fusing image and depth features within a monolithic VLA [54, 47]. To alleviate the reliance on external depth models, MolmoAct [33] generates depth tokens inside the VLM at the cost of token-generation overhead. DepthVLA [66] more tightly couples depth with control by integrating a depth transformer into the action expert and finetunes the depth model.

Another line of work directly incorporates sensor-derived geometry such as RGB-D or point clouds. Some works [72, 38] fuse point clouds and image representations as inputs to the VLM, which often requires training on large-scale 3D datasets. To better leverage pretrained 2D VLMs, BridgeVLA [35] relies on multi-view images to predict 2D action heatmaps and lifts them to 3D using camera poses and depth. While effective for long-horizon keypoint pose prediction, this design is specialized for positional control and is difficult to extend to other action formats like delta movements or velocity control. Other point cloud augmented VLAs [34, 60] train point cloud encoders to inject the learned 3D embeddings into action experts. However, these methods primarily rely on coarse-grained global geometry features, and observe no benefit from pretraining the point cloud encoder [34]. Our method instead integrates point clouds into the action expert at multiple scales, enabling dense interactions between 3D point features and action tokens, and demonstrates strong performance across multiple benchmarks and action types.

III. Method

A. Problem Formulation

We study vision-and-language guided policy learning for robotic manipulation. At each timestep t, the agent receives a multi-modal observation $O_t = \{I_t, P_t, s_t, L\}$. Here, $I_t \in \mathbb{R}^{N_I \times H_I \times W_I \times 3}$ represents multi-view RGB images, L is a natural language instruction, and s_t is the robot's proprioceptive state. To provide explicit geometric grounding, we include a colored 3D point cloud $P_t \in \mathbb{R}^{N_P \times 6}$. We do not assume that P_t is pre-aligned with I_t, although such alignment



(a) Monolithic 3D-aware VLA (b) Dual-system 3D-aware VLA (c) The proposed PointACT VLA

Fig. 1: **Comparison of 3D integration strategies in VLAs.** (a) Monolithic 3D-aware VLA: 3D point features are fed directly into the pretrained VLM backbone, which largely increases the computation burden and may disrupt pretrained representations. (b) Dual-system 3D-aware VLA: 3D information is introduced into a separate action expert, but typically through coarse-grained global features with limited interaction between geometry and actions. (c) PointACT (ours): a dedicated point–action expert enables fine-grained, multi-scale interaction between point cloud features and action tokens during decoding, preserving the pretrained VLM while allowing geometry to directly shape action generation.

can be recovered given calibrated camera poses or multi-view geometric reconstruction.

The objective is to learn a policy π_θ that predicts a sequence of H future actions $A_t = (a_t, \ldots, a_{t+H-1})$ to accomplish the task specified by L:

$$A_t = \pi_\theta(I_t, P_t, s_t, L), \tag{1}$$

where each a_t may represent end-effector pose, or joint-space commands depending on the control interface.

B. Preliminary: 3D-aware VLAs

Our goal is to enhance standard 2D-centric VLA policies with explicit 3D geometric reasoning. To contextualize our proposed model, we present two baseline strategies for injecting 3D information into VLA models: a *monolithic* integration strategy and a *dual-system* modular strategy (Figure 1 (a),(b)). We first describe the point cloud encoder shared by both strategies and then present them in detail. We conclude by pointing out the limitations of both strategies and how we improve over them.

Point Cloud Encoder. To encode geometric observations P_t, we employ Point Transformer v3 (PTv3) [64]. PTv3 operates directly on unordered point sets and uses a serialization-based grouping strategy to construct local neighborhoods. Self-attention is performed within each group, enabling efficient computation while preserving local geometric structure. The network follows a hierarchical design in which point features are progressively downsampled across stages, producing multi-scale geometric representations that capture both local detail and global layout. We initialize the encoder from a PTv3 model pretrained via large-scale self-supervised learning on building-level 3D scenes [69]. Although this pretraining domain differs from manipulation settings, it provides strong low-level geometric priors.

Monolithic 3D-aware VLA. This approach treats geometric data as an additional input modality to a unified VLM backbone (Figure 1a). We build upon the EO1 [53] VLA architecture, which utilizes Qwen2.5-VL [3] as the foundation. The multi-view images are processed by a pretrained vision encoder, while text tokens and proprioceptive state s_t are mapped into the VLM's latent space via an embedding layer and a linear projection, respectively. We further project point embeddings from the final layer of the PTv3 encoder into the same latent space, and concatenate all the modality tokens as inputs to the VLM. This enables the method to perform joint conditioning on image, language, and geometry within a single transformer. The model is trained using a flow-matching objective [40], where the VLM receives noised action tokens and learns to predict the velocity field for iterative denoising.

Dual-system 3D-aware VLA. This approach decouples semantic reasoning from motor control (Figure 1b), following the GR00T [5] VLA architecture. However, instead of using the GR00T pretrained weights, we use the same Qwen2.5-VL as the VLM backbone for fair comparison with the monolithic VLA variant. The VLM backbone remains frozen to preserve its general-purpose representations. A separate *action expert* (e.g., diffusion transformer [50]) is trained from scratch to predict A_t conditioned on the VLM features, robot's proprioceptive state s_t, and point embeddings from the last layer of the PTv3 encoder.

In both paradigms, point features are fused only at a high level, limiting fine-grained geometric influence during action prediction. We address this limitation with a multi-scale point-action interaction mechanism.

C. PointACT: Multi-scale Point-Action Interaction

We introduce **PointACT**, a geometry-aware action expert that injects structured point cloud information into action decoding process through efficient multi-scale point-action interaction. Figure 2 illustrates our model architecture.

Let $Z_p^l \in \mathbb{R}^{N_p^l \times D^l}$ denote the point embeddings from the l-th layer of the PTv3 encoder, where N_p^l is large in early layers and decreases with depth due to hierarchical pooling. Let $Z_a^l \in \mathbb{R}^{(H+1)\times D^l}$ denote the action token embeddings and the robot

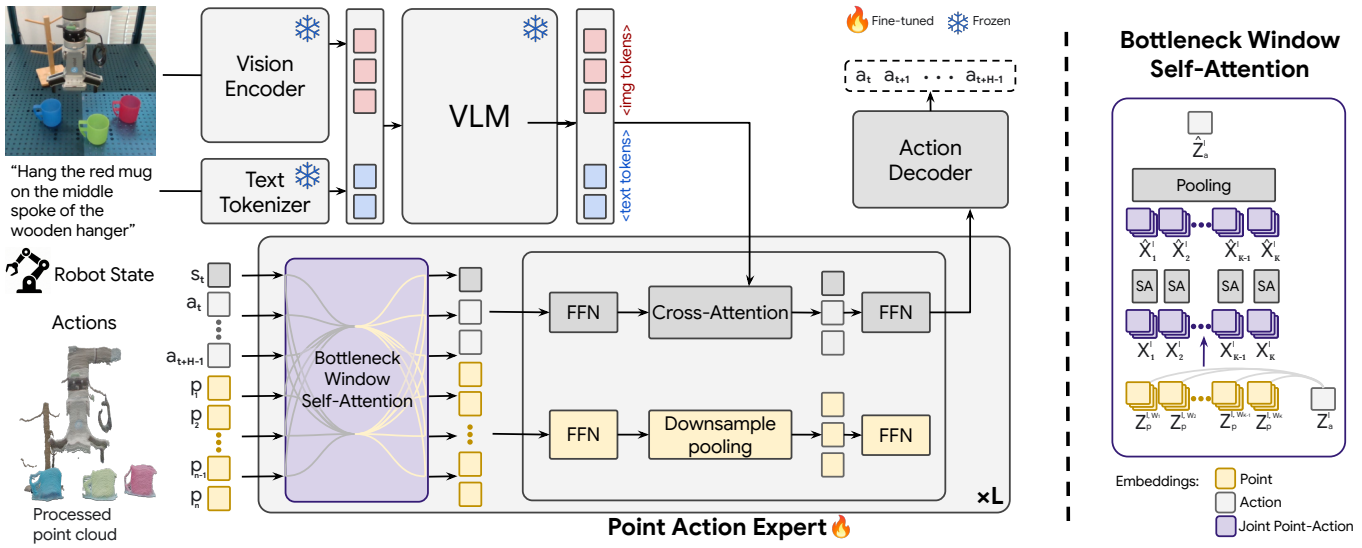


Fig. 2: **(Left): PointACT Dual-Model Architecture. (Right): Bottleneck Window Self-Attention mechanism.** PointACT is a VLA model that equips a frozen pretrained VLM backbone with a point-cloud action expert for geometry-aware control. Language, images, robot state, and 3D point clouds are encoded into tokens, with point clouds producing multi-scale geometric features via a Point Transformer. These point tokens interact with action tokens through multi-scale point–action attention in the action expert, allowing both global structure and local geometry to guide action generation.

proprioceptive state at the corresponding decoding layer, and Z_{vlm} the output embeddings from the VLM backbone.

As N_p^l can be large, we propose a *bottleneck window-based self-attention* mechanism inspired by [46]. Specifically, we partition the point cloud into K disjoint spatial windows $(W_1, \cdots, W_K)$ using the same serialization strategy as PTv3, with $\lceil N_p^l / K \rceil$ points per window. For each window k, we broadcast the action tokens to the point cloud tokens:

$$X_k^l = [Z_p^{l,W_k}; Z_a^l]. \tag{2}$$

We then apply a shared self-attention block across all windows to update the joint point-action representation:

$$\hat{X}_k^l = \text{Self-Attn}(X_k^l). \tag{3}$$

The action tokens serve as a latent bottleneck that aggregates local geometric context, enabling interaction with many local regions while keeping attention complexity linear in the number of windows. To produce the updated global action representation, we extract the action features from each window and perform an average pool across all K groups:

$$\hat{Z}_a^l = \frac{1}{K} \sum_{k=1}^{K} \hat{X}_{k,a}^l \tag{4}$$

To distinguish action and point modalities, we employ distinct feed-forward networks (FFN) for action and point tokens after the self-attention.

Following this geometric fusion, the action tokens undergo cross-attention with the VLM embeddings Z_{vlm} to incorporate high-level visual and linguistic instructions:

$$\bar{Z}_a^l = \text{Cross-Attn}(\hat{Z}_a^l, Z_{vlm}). \tag{5}$$

By repeating this process across the hierarchical stages of the PTv3 encoder, the action tokens are conditioned on a spectrum of geometric information, from fine-grained local details to coarse-grained global scene structures.

D. Action Prediction and Training

PointACT is trained using behavior cloning on demonstration trajectories. Depending on the action space, we utilize either a regression-based or a classification-based action head.

Regression Head. For continuous control and short-horizon action chunking, we employ a multi-layer perceptron (MLP) to map the output action token embeddings to a sequence of control parameters. Following recent findings that regression can be more training-efficient than diffusion-based methods [31, 49], we optimize the model using an L_2 loss over the predicted action chunk per training example:

$$\mathcal{L}_{\text{reg}} = \frac{1}{H} \sum_{i=t}^{t+H-1} \|\mathbf{a}_i - \mathbf{a}_i^*\|_2^2 \tag{6}$$

where a_i^* is the ground-truth action.

Classification Head. For predicting keypoint actions (e.g., long-term end-effector poses) [61, 22], we adopt a point-anchored classification strategy to better handle the multi-modal distribution of robot trajectories. We discretize the target workspace into bins relative to the observed point cloud, as the keypoint often denotes or is close to contact points of the object. Specifically, for a target position $v^* = (x, y, z)$, we compute its coordinates relative to the nearest point $p_j \in P_t$ and assign it to a discrete spatial bin around the point. The rotations are also discretized into N_{rot} bins.

To strengthen geometric grounding, we concatenate the final point embeddings with the action tokens before the prediction

layer. The model is trained using a cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = - \sum_{k \in \{pos, rot, open\}} \sum_{b=1}^{B_k} y_{k,b} \log(\hat{y}_{k,b}) \tag{7}$$

where B_k denotes the number of bins for each action component. This approach allows the model to leverage the point cloud as a spatial anchor for localization.

IV. Experiments

A. Implementation Details

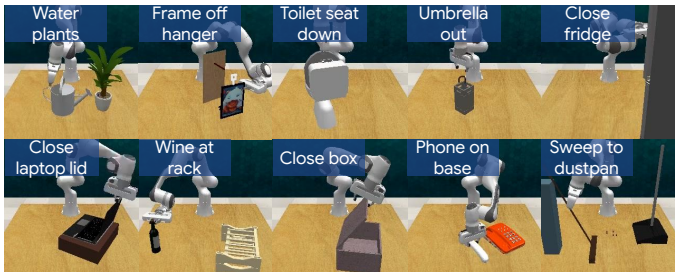
We utilize Qwen2.5-VL [3] as the VLM backbone. To initialize from the pretrained PTv3 model [69], we use the same model configuration as PTv3-Large. Specifically, there are 5 hierarchical stages with $(3, 3, 3, 12, 3)$ layers and feature dimensionality of $(64, 128, 256, 512, 768)$ per stage. The PointACT module, including the hierarchical interaction blocks and the action head, comprises approximately 300M trainable parameters. To maintain the knowledge of the VLM, we keep the VLM backbone frozen if not otherwise mentioned. We crop the input point cloud using a workspace bounding box defined relative to the robot base frame [57, 11], and perform voxelization with size of 1cm. We set the maximum number of points as 4096.

Across our experimental benchmarks, the model is trained with a total batch size of 128 distributed across 2 NVIDIA H100 GPUs. We optimize the model for 20K to 50K gradient steps using the AdamW optimizer with a learning rate of 5×10^{-5} and a cosine decay schedule. We apply standard point cloud augmentations, including random rotation around the gravity axis, alongside image-level augmentations for the VLM input during training.

B. Evaluation Benchmarks

We evaluate PointACT on two widely used simulation benchmarks LIBERO [41] and RLBench [27], to assess its performance in both short-horizon actions and long-horizon keypoint actions. Figure 3 presents some examples of the two benchmarks.

LIBERO. We utilize four task suites from the LIBERO benchmark: *LIBERO-Spatial*, *LIBERO-Object*, *LIBERO-Goal*, and *LIBERO-Long*. Each suite contains 10 tasks with 50 expert demonstrations per task. Following the protocol in [31], we filter unsuccessful demonstrations to ensure high-quality training data. For this benchmark, the action space consists of delta end-effector poses. Therefore, we employ the regression head with an action chunk size of $H = 16$. During inference, we execute the first 8 steps before replanning. For evaluation, we run 50 episodes per task following the standard protocol, resulting in 500 evaluation episodes per task suite. The supervised training and evaluation setup in the LIBERO benchmark is relatively easy, as test episodes differ from training data only by minor variations in object placement. We report the average task success rate for each suite, using a binary score (1 for success, 0 otherwise) with no partial credit.



(a) RLBench 10 tasks.

(b) LIBERO 4 task suites.

Fig. 3: **Illustration of the simulated benchmarks.** (a) Examples of tasks from the RLBench 10 tasks benchmark [42], covering a diverse set of manipulation skills such as object placement, articulated object interaction and tool use. (b) Representative tasks from the LIBERO benchmark [41], including spatial reasoning, object pick-and-place, goal-conditioned tasks, and long-horizon manipulation.

RLBench. We use the same 10 RLBench tabletop manipulation tasks as HybridVLA [42], including *Close box*, *Close laptop*, *Toilet seat down*, *Sweep to dustpan*, *Close fridge*, *Phone on base*, *Take umbrella out*, *Frame off hanger*, *Wine at rack*, and *Water plants*. We train a unified multi-task policy using 100 demonstrations per task, and use the keypoint action space following prior work [42, 57, 11, 22, 17, 52]. The classification head is used for keypoint prediction with action chunk size $H = 1$ since the action is already long-horizon. An RRT-based motion planner is used to compute collision-free trajectories to the predicted keypoint. Following the HybridVLA test setup, the testing episodes contain distinct object placements compared to the training episodes. We evaluate each task over 100 episodes to ensure statistical reliability, as prior work has shown that motion planner in RLBench can introduce large execution variability [22]. The success rate is reported for each task.

For both LIBERO and RLBench benchmarks, we use a single front-view RGB image with 256×256 resolution. The point cloud P_t is obtained from depth camera using known camera intrinsics and extrinsics.

C. Comparison with State of the Art

LIBERO. Table I compares PointACT with prior VLA methods on the LIBERO benchmark. Direct comparison across VLAs is challenging because methods often use different training setups, data mixtures, and model configurations that are not always fully specified. For a fair comparison, we reproduce EO1 [53] with our training pipeline. As a monolithic model, EO1 requires finetuning the VLM backbone (we keep the vision encoder frozen), resulting in more trainable parameters than PointACT. Even under this favorable setting,

TABLE I: **Success rate (%) on the LIBERO benchmark.** "Pretrained" denotes whether the model is fully pretrained, and "train tasks" denotes the number of LIBERO tasks the model is fine-tuned on; some methods leverage all LIBERO tasks including LIBERO-90. We put a question mark if the setting is not clear from the original paper.

	Model size	Wrist camera	Pre-trained	Frozen VLM	Train tasks	Spatial	Object	Goal	Long	Avg.
2D VLAs										
TraceVLA [74]	7B	?	✓	×	40	84.6	85.2	75.1	54.1	74.8
ThinkAct [23]	7B	?	✓	×	?	88.3	91.4	87.1	70.9	84.4
Octo [19]	0.1B	×	✓	×	10	78.9	85.7	84.6	51.1	75.1
OpenVLA [32]	7B	×	✓	×	10	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [31]	7B	✓	✓	×	10	97.6	98.4	97.9	94.5	97.1
FLOWER [56]	1B	✓	✓	×	10	97.1	96.7	95.6	93.5	95.7
SmolVLA [58]	2B	✓	✓	✓	40	93	94	91	77	88.8
MolmoAct [33]	7B	✓	✓	×	10	87	95.4	87.6	77.2	86.8
GR00T-N1 [5]	3B	✓	✓	?	100	94.4	97.6	93	90.6	93.9
GR00T-N1.5 [5]	3B	✓	✓	✓	10	92	92	86	76	86.5
GR00T-N1.6 [5]	3B	✓	✓	✓	10	97.7	98.5	97.5	94.4	97.0
π_0 [6]	3B	✓	✓	×	10	96.8	98.8	95.8	85.2	94.2
π_0 + FAST [51]	3B	✓	✓	×	10	96.4	96.8	88.6	60.2	85.5
$\pi_{0.5}$ [26]	3B	✓	✓	×	10	98.8	98.2	98.0	92.4	96.9
X-VLA [73]	0.9B	✓	✓	×	?	98.2	98.6	97.8	97.6	98.1
EO1 [53]	3B	✓	✓	×	100	99.7	99.8	99.2	94.8	**98.4**
EO1 [53] (reproduced)	3B	×	✓	×	10	91.8	98.2	96.6	85.6	93.1
3D-aware VLAs										
SpatialVLA [54]	4B	×	✓	×	10	88.2	89.9	78.6	55.5	78.1
PointAct (Ours)	3B	×	×	✓	10	97.4	99.6	96.2	90.6	96.0

TABLE II: **Success rate (%) on RLBench 10 tasks.** We reproduce EO1 [53] and ACT3D [18], and evaluate them alongside PointACT using 100 episodes per task. The results for the other methods are directly taken from [42], where each method is evaluated over 20 episodes.

	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean
ManipLLM (7B) [37]	50	80	40	20	80	35	10	25	15	20	38
OpenVLA (7B) [32]	65	40	75	60	80	20	35	15	10	10	41
π_0 [6] (2.6B)	90	60	100	30	90	25	35	75	5	45	55
CogACT (7B) [36]	80	85	90	65	90	50	60	35	25	25	60
HybridVLA (7B) [42]	85	95	100	90	100	50	50	70	50	50	74
EO1 (3B) [53]	97	99	100	95	83	46	76	40	61	35	73.2
ACT3D [18]	94	62	99	80	87	53	97	31	31	11	64.5
PointACT (3B)	91	99	96	59	81	99	99	69	90	40	**82.3**

PointACT outperforms the reproduced EO1 by a large margin on Spatial and Long task suites, while achieving comparable performance on the Goal and Object suites. These results highlight the benefit of explicit 3D geometric information for spatially complex and temporally extended tasks. Additional controlled comparisons are provided in our ablation studies.

Our method outperforms most existing methods, though it performs 2% worse than X-VLA [73] and EO1 [53]. However, results for these methods are not directly comparable, due to differences in experimental settings; specifically, those methods leverage pretraining on large-scale robot datasets and employ multi-view camera setups.

RLBench. On RLBench, we report baseline results from the HybridVLA paper and reproduce EO-1 and ACT3D [18] under our evaluation setup to ensure a fair comparison. The compared methods, except for ACT3D, are image-based VLAs that do not use explicit 3D geometric inputs, whereas ACT3D is a state-of-the-art policy conditioned on both image and point cloud inputs. Under the same evaluation protocol, PointACT achieves substantial performance gains over prior 2D- and 3D-based baselines across tasks.

Figure 4 illustrates representative failure modes across RLBench tasks, highlighting current limitations in spatial perception and long-horizon interaction.

- *Perceptual occlusion*: In the CloseFridge task, failure predominantly stems from partial views. This suggests multi-view integration or active perception is required to resolve occlusions.
- *Limited failure recovery*: Across several tasks, we observe a lack of reactive recovery once a trajectory deviates from the nominal path. If an initial prediction error leads to an unintended collision or a missed grasp, the model often fails to execute a corrective maneuver, instead continuing the pre-planned sequence. For example, in the Frame off Hanger task, the initial predicted pose is typically accurate; however, slight contact during the reach phase often shifts the object. The model does not recover from this failure but continues to perform the next step.

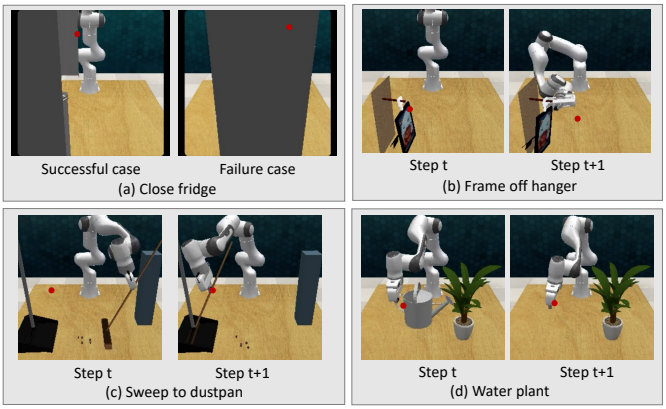


Fig. 4: Failure cases in RLBench benchmark. The red dot denotes the position of the predicted next keypoint action projected in the front view image.

TABLE III: **Comparison of 3D integration strategies in VLAs.** We evaluate monolithic and dual-system VLAs with and without point cloud integration on LIBERO-Spatial and RLBench-10Tasks. For dual-system models, action experts are trained from scratch. GR00T(arch) uses the GR00T [5] action expert architecture without pretrained weights.

		Trainable Params	LIBERO -Spatial	RLBench -10Tasks
Monolithic VLA	EO1	3B	91.8	73.2
	EO1 + Point	3B	94.0	18.6
Dual-system VLA	GR00T(arch)	1B	87.0	50.8
	GR00T(arch) + Point	1B	92.0	69.7
	PointACT	300M	**97.4**	**82.3**

This suggests that while 3D geometric features enhance initial precision, the model currently lacks the closed-loop temporal reasoning necessary to recover from high-level "deadlock" states or physical perturbations.

- *Tool-mediated interaction*: In the Sweep to Dustpan and Water Plant tasks, we observe inaccuracy in indirect object manipulation. While our PointACT accurately predicts primary contact points (e.g., grasping the broom), it struggles with the secondary spatial constraints of the tool itself. Specifically, inaccurate height estimation leads to insufficient contact between the broom and the dustpan, while collisions happen between the watering bottle and the plant. These cases indicate a need for better reasoning regarding tool-to-target spatial relationships.

D. Ablation Studies

3D injection architectures. We compare two strategies for integrating geometric information into the VLA framework as described in Section III-B. The baselines include: (1)EO1 [53], a standard 2D VLA; (2) EO1 + Point, the monolithic 3D-aware architecture; (3) GR00T(arch), a dual-system VLA following GR00T [5] architecture (using the same frozen VLM backbone as EO1 and without GR00T pretraining); (4) GR00T(arch) + Point, which adds coarse-grained geometric tokens to the action expert; and (5) our proposed PointACT model with multi-scale point-action interaction.

TABLE IV: **Comparison of different strategies for incorporating multi-scale point features into dual-system VLAs.** We report the average success rate on RLBench-10Tasks. K denotes the number of point tokens sampled from each scale.

	GR00T (arch)	GR00T(arch) + Point		PointAct (Ours)
		Final layer	Multi-scale K=64	Multi-scale K=128	
RLBench SR	50.8	69.7	65.2	65.6	**82.3**

Results are presented in Table III. For monolithic VLAs, injecting point cloud features does not achieve consistent improvements across benchmarks. It significantly decreases the performance on more challenging RLBench, where the training and testing episodes exhibit larger placement difference. This suggests that directly augmenting pretrained VLMs with 3D tokens does not effectively translate geometric information into improved action generation and may interfere with learned representations in VLM, and thus may require more pre-training with 3D integrated. Adding point clouds to action experts improves performance more stably as shown in the dual-system VLAs. Compared to GR00T(arch)+Point which only utilizes coarse-grained point features from the last point encoder layer, the proposed PointACT achieves better results with fewer trainable parameters, demonstrating the effectiveness of fine-grained point–action interaction.

Multi-scale point features. We further investigate the use of multi-scale point features in the more stable dual-system VLA setting. As a straightforward baseline, we project features from different scales using separate MLPs, concatenate them into a single token sequence, and inject them into the action expert via cross-attention. To keep the computational cost manageable despite the large number of point tokens, we sample a fixed number of tokens K from each of the five scales in the PTv3 model. As shown in Table IV, this naive multi-scale concatenation strategy does not improve performance for dual-system VLAs. This result suggests that simply aggregating point features across scales is insufficient, and further demonstrates the necessity of our fine-grained multi-scale point-action interaction mechanism.

Contribution of the 2D image modality. To isolate the importance of 2D visual context versus 3D geometric reasoning, we evaluate PointACT with and without image features. In the "without image" variant, the model relies solely on language instructions from the VLM and the 3D point cloud. While 3D geometry provides strong spatial understanding, our results in Table V indicate that 2D visual features still provide critical semantic cues that complement the geometric representation.

Model size of PointACT. We investigate the performance impact of the action expert's capacity by scaling the PointACT module. Following the PTv3 architectural configurations, we evaluate Small (∼59M parameters), Base (∼167M parameters), and Large (∼314M parameters) variants. As shown in Figure 5, larger models slightly improve the performance.

TABLE V: Performance with and without conditioning on image embeddings from the frozen VLM.

Image conditioning	LIBERO-Spatial	RLBench-10Tasks
×	94.2	79.8
✓	**97.4**	**82.3**

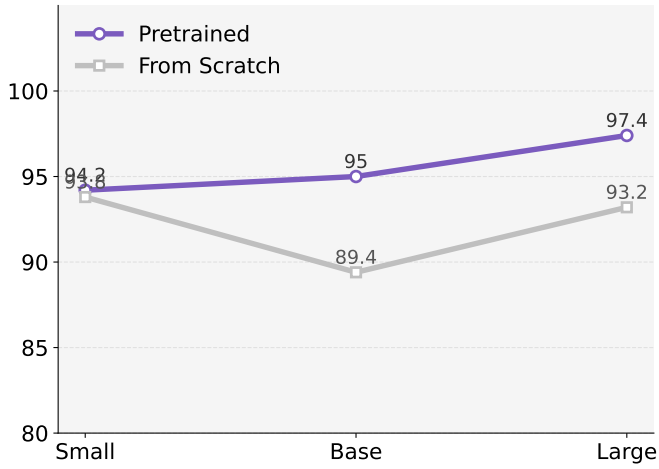


Fig. 5: Performance on LIBERO-Spatial across different action expert sizes, with and without pretrained PTv3 weights [69].

Notably, the smallest PointACT action expert outperforms the baseline without a point cloud action expert shown in Table III.

PTv3 pretraining. We further measure the benefits from pre-trained PTv3D models in Figure 5. There is a large domain gap between the pretraining dataset and the manipulation dataset, where the pretraining dataset consists of large-scale building-level scenes with many points that are coarsely sampled, while manipulation focuses on near areas with more densely sampled points. Pretraining does not help much for small models, but provides greater benefits for larger models, potentially due to the increased optimization difficulty associated with higher-capacity architectures. We will investigate pretraining on manipulation related environments [12] in future work.

V. Real World Experiments

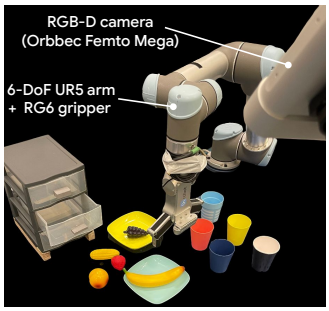
We evaluate our method on two distinct robotic platforms to assess performance under realistic conditions, including the SO-100 with microstep action control and the UR5 with keypoint-based actions.

We compare our approach against two state-of-the-art VLAs: π_0 [6] and GR00T-N1.5 [5]. We additionally compare with a state-of-the-art point cloud based policy 3DLotus [17], which is designed to predict keypoint actions, on the UR5 platform. Multi-task policies are trained to perform tasks following textual instructions.

A. SO-100 Robot Platform

Experimental Setup. The SO-100 is a 6-DoF 3D-printed arm equipped with a parallel gripper. The workspace consists of a 50cm × 50cm tabletop area monitored by a fixed front-view Intel RealSense D435 camera, see Figure 6a. The observation

(a) 6-DoF SO100 arm with a front-view RGB-D camera.



(b) 6-DoF UR5 arm with a front-view RGB-D camera.

Fig. 6: **SO100 and UR5 robot setups.**

TABLE VI: **Performance on the SO-100 robot platform**. We report success rates over 10 trials, with partial scores shown in parentheses.

Task	π_0 [6]	GR00T-N1.5 [5]	PointACT (Ours)
Put Banana In Plate	**10/10** (10)	8/10 (8)	**10/10** (10)
Put Sock In Drawer	2/10 (5)	5/10 (6.5)	**9/10** (9)
Open Microwave	7/10 (7)	5/10 (5)	**8/10** (8)

space includes an RGB image, a point cloud obtained from depth maps and known camera parameters, and 6D joint positions. The action space is the 6D absolute joint positions.

Tasks and Protocol. We evaluate performance across three manipulation tasks described in the following. We report success rate, but also a partial score defined for each tasks.

- *Put Banana In Plate:* Pick up a banana and place it on a blue plate. (Partial credit: 0.5 for grasp, 0.5 for placement)
- *Put Sock In Drawer:* Pick up a sock amidst distractors, place it inside of a middle drawer, and finally close the drawer. (Partial credit: 0.5 for grasp, 0.25 for partial closure of the drawer; 0.5 for full closure)
- *Open Microwave:* Open the microwave door to 90°. (Partial credit: 0.5 for partially open, 1 for fully open)

We collect 50 episodes per task for training using the leader-follower arms. Each episode involves randomized initial object placements. Evaluation consists of 10 independent test episodes per task. For fair comparison, all methods are tested using the same initial object positions.

Baselines. We fine-tune baseline models on our collected SO-100 dataset following the standard configurations provided by the LeRobot library [9]. For π_0, we use the official pretrained model weights and train the entire model parameters including the vision encoder with an action chunk size of 50. For GR00T-N1.5: We freeze the VLM backbone and fine-tune only the action expert with an action chunk size of 16. Unlike the baselines, our model lacks massive robotic pre-training. Therefore, we initialize our PointACT from a LIBERO checkpoint and fine-tune the action expert (VLM is kept frozen) with an action chunk of 16.

Results. Table VI shows the result with the SO-100 robot. Despite the lack of extensive pre-training, our model achieves comparable or better performance than π_0 and GR00T.

TABLE VII: **Performance on the UR5 robot platform.** We report success rates over 10 trials, with partial scores shown in parentheses.

Task	π_0	GR00T-N1.5	3DLotus	PointACT
Stack Yellow Cup Onto Pink Cup	0/10 (2)	0/10 (1.5)	7/10 (7)	**7/10** (7.5)
Close Drawer	**9/10** (9)	**9/10** (9)	2/10 (2)	7/10 (8)
Put Grapes and Banana in Plates	0/10 (2.25)	0/10 (1.75)	0/10 (3)	**4/10** (7)

Compared to π_0, our approach performs similarly on Put Banana and Open Microwave tasks, but outperforms π_0 by a margin (2 versus 9 successful actions) on the more complex Put Sock task. While our approach correctly places the socks inside the drawer and manages to close it, π_0 often fails to place the sock fully inside the drawer, leading to collisions that prevent the drawer from closing. We hypothesize that this behavior is due to missing 3D information.

Compared to GR00T, our approach performs consistently better. We observe that GR00T often produces jittery actions, struggles with precise grasps, and can fail even on simple tasks such as Put Banana when objects appear in diverse poses. In contrast, PointACT generates noticeably smoother trajectories with reduced jitter and more accurate actions, benefiting from explicit 3D perception.

B. UR5 Robot Platform

Experimental Setup. We use a 6-DoF UR5 robotic arm with a parallel RG6 gripper. The robot operates over a $1\,\mathrm{m} \times 1\,\mathrm{m}$ tabletop workspace observed by a fixed Orbbec Femto Mega RGB-D camera, see Figure 6b. Observations include an RGB image, a point cloud reconstructed from depth, and an 8D end-effector state (position, orientation, and gripper openness). Actions are specified as 8D absolute end-effector poses of the next keypoint, as collecting keypoint actions is easier with a joystick controller compared to continuous microstep actions.
Tasks and Protocol. We evaluate performance across three manipulation tasks, and report success rate and a partial score. The tasks and partial success scores are defined as follows:

- *Stack Yellow Cup Onto Pink Cup:* Pick up a yellow cup amidst distractors, and stack it onto a pink cup. (Partial credit: 0.5 for grasp; 0.5 for placement)
- *Close Drawer:* Move an open drawer (either top or middle-level drawer) to its closed position. (Partial credit: 0.5 for partially close; 1 for fully close)
- *Put Grapes and Banana in Plates:* Pick up grapes amidst distractors, place it into a yellow plate, then pick up a banana amidst distractors, place it into a pink plate. (Partial credit: 0.25 per correct grasp or placement)

We collect 20 episodes per task for training with a joystick controller, and evaluate on 10 test episodes per task, with randomized initial object placements in each episode.
Baselines. All models are fine-tuned on our collected UR5 dataset with action chunk size of 1 for keypoint action prediction. Following standard practice, the models are first pretrained on RLBench and subsequently fine-tuned on real-robot tasks using the pretrained checkpoints.

Results. Table VII reports the results on the UR5 platform. The 3D-based policies outperform 2D-VLAs for all tasks except Close Drawer. In this task, the drawer's transparency leads to noisy and incomplete point clouds. As a result, 3DLotus, which relies purely on geometric input, suffers a significant performance drop. In contrast, our PointACT fuses RGB and point cloud cues, achieving substantially stronger results than 3DLotus. This result also highlights the importance of improving real-world point cloud quality. On the cup-stacking task, our method achieves performance comparable to 3DLotus while substantially outperforming the 2D VLA baselines, π_0 and GR00T-N1.5. This task requires precise localization of the cup rim for successful grasping, which is challenging for 2D-only VLAs with limited robot data. At the same time, the task relies less on semantic understanding, which narrows the performance gap between our method and the pure geometry-based 3DLotus baseline. For the put fruit in plate task, both geometric precision and semantic understanding are important. PointACT achieves more accurate pre-grasp alignment, which we attribute to its fine-grained point cloud reasoning combined with semantic guidance from the VLM. Most failures occur during grasping, where small pose errors of the parallel gripper lead to consistent misses, particularly for the grapes.

Experimental results on the SO-100 and UR5 platforms demonstrate the superiority of our approach on real robots under noisy point clouds and actuation uncertainty. These results highlight the importance of joint 2D-3D reasoning with multi-scale point–action interactions.

VI. Conclusion

In this work, we presented PointACT, a 3D-powered Vision-Language-Action framework that integrates point clouds directly into action generation through a dedicated point-cloud action expert. Unlike prior approaches that attach 3D cues to image features or inject geometry into pretrained backbones, PointACT introduces multi-scale point-action interaction within the action decoder, enabling geometric information to continuously condition action tokens while preserving pretrained VLM representations. Our systematic study of 3D integration strategies shows that action-expert-level fusion with fine-grained point–action attention is significantly more effective than backbone-level injection or coarse geometric features. Future work includes scaling point-based pretraining on large-scale robot datasets for manipulation-centric geometry, improving robustness under noisy point observations, and enhancing failure recovery abilities with 3D VLAs.

Acknowledgments

This work was granted access to HPC resources of IDRIS under the allocation AD011014846 made by GENCI. It was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "France 2030" program, reference ANR-23-IACL-0008 (PR[AI]RIE-PSAI projet), the ANR project 3D-GEM (ANR-25-CE23-7777-01), the ANR project VideoPredict (ANR-21-FAI1-0002-01) and the Paris Île-de-France Région in the frame of the DIM AI4IDF. Cordelia Schmid would like to acknowledge the support by the Körber European Science Prize.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Smail Ait Bouhsain, Rachid Alami, and Thierry Simeon. Simultaneous action and grasp feasibility prediction for task and motion planning through multi-task learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2042–2048. IEEE, 2023.

[3] 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.

[4] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[5] 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.

[6] 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.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *RSS*, 2023.

[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] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[10] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[11] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. In *7th Conference on Robot Learning (CoRL 2023)*, 2023.

[12] Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. SUGAR: Pre-training 3D visual representations for robotics. In *CVPR*, 2024.

[13] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025.

[14] Zerui Chen, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Vividex: Learning vision-based dexterous manipulation from human videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3336–3343. IEEE, 2025.

[15] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *IJRR*, 2025.

[16] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

[17] Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8996–9002. IEEE, 2025.

[18] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3D: 3D feature field transformers for multi-task robotic manipulation. In *CoRL*, 2023.

[19] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

[20] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. RVT: Robotic view transformer for 3D object manipulation. In *CoRL*, 2023.

[21] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. RVT2: Learning precise manipulation from few demonstrations. In *RSS*, 2024.

[22] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *CoRL*, 2023.

[23] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

[24] Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation mod-

els. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9488–9495. IEEE, 2024.

[25] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, pages 540–562. PMLR, 2023.

[26] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

[28] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *CVPR*, 2022.

[29] 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 *CoRL*, 2022.

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

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

[32] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713. PMLR, 2025.

[33] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[34] Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026.

[35] 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.

[36] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[37] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18061–18070, 2024.

[38] Xiaoqi Li, Liang Heng, Jiaming Liu, Yan Shen, Chenyang Gu, Zhuoyang Liu, Hao Chen, Nuowei Han, Renrui Zhang, Hao Tang, et al. 3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation. In *9th Annual Conference on Robot Learning*, 2025.

[39] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[40] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

[41] 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.

[42] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[43] Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. In *Conference on Robot Learning*, pages 527–538. PMLR, 2023.

[44] Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0.5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026.

[45] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[46] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *Advances in neural information processing systems*, 34:14200–14213, 2021.

[47] Nikolay Nikolov, Giuliano Albanese, Sombit Dey, Aleksandar Yanev, Luc Van Gool, Jan-Nico Zaech, and Danda Pani Paudel. Spear-1: Scaling beyond robot demonstrations via 3d understanding. *arXiv preprint*

arXiv:2511.17411, 2025.

[48] 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.

[49] Chaoyi Pan, Giri Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. *arXiv preprint arXiv:2512.01809*, 2025.

[50] 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.

[51] 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.

[52] Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. In *RSS 2024*.

[53] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. *arXiv preprint arXiv:2508.21112*, 2025.

[54] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[56] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. *arXiv preprint arXiv:2509.04996*, 2025.

[57] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *CoRL*, 2023.

[58] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[59] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[60] Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

[61] Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. Kite: Keypoint-conditioned policies for semantic manipulation. In *Conference on Robot Learning*, pages 1006–1021. PMLR, 2023.

[62] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[63] 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.

[64] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.

[65] Rujia Yang, Geng Chen, Chuan Wen, and Yang Gao. Fp3: A 3d foundation policy for robotic manipulation. *arXiv preprint arXiv:2503.08950*, 2025.

[66] Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

[67] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning*, pages 3157–3181. PMLR, 2025.

[68] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[69] Yujia Zhang, Xiaoyang Wu, Yixing Lao, Chengyao Wang, Zhuotao Tian, Naiyan Wang, and Hengshuang Zhao. Concerto: Joint 2d-3d self-supervised learning emerges spatial representations. *arXiv preprint arXiv:2510.23607*, 2025.

[70] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea 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.

[71] Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *RSS*, 2023.

[72] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: a 3d vision-language-action generative world model. In *Proceedings of the 41st International Conference on Machine Learning*, pages 61229–61245, 2024.

[73] 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.

[74] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[75] Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

[76] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.