Evo-Depth: A Lightweight Depth-Enhanced Vision-Language-Action Model

Tao Lin[1,*] **Yuxin Du**[1,*] **Jiting Liu**[1,*] **Nuobei Zhu**[1] **Yunhe Li**[1]
Yuqian Fu[2] **Yinxinyu Chen**[1] **Hongyi Cai**[1] **Zewei Ye**[1] **Bing Cheng**[1,4]
Kai Ye[1] **Yiran Mao**[1] **Yilei Zhong**[1] **Mingkang Dong**[1] **Junchi Yan**[1]
Gen Li[3,†] **Bo Zhao**[1,4,†]

[1]School of AI, Shanghai Jiao Tong University
[2]KAUST [3]NTU [4]SJTU-Quic Robot Joint Lab

taolin200108@gmail.com, bo.zhao@sjtu.edu.cn
https://github.com/MINT-SJTU/Evo-Depth

[*]Equal contribution [†]Corresponding authors

Abstract

Vision-Language-Action (VLA) models have emerged as a promising paradigm for robotic manipulation by unifying perception, language grounding, and action generation. However, they often struggle in scenarios requiring precise spatial understanding, as current VLA models primarily rely on 2D visual representations that lack depth information and detailed spatial relationships. While recent approaches incorporate explicit 3D inputs such as depth maps or point clouds to address this issue, they often increase system complexity, require additional sensors, and remain vulnerable to sensing noise and reconstruction errors. Another line of work explores implicit 3D-aware spatial modeling directly from RGB observations without extra sensors, but it often relies on large geometry foundation models, resulting in higher training and deployment costs. To address these challenges, we propose **Evo-Depth**, a lightweight depth-enhanced VLA framework that enhances spatially grounded manipulation without relying on additional sensing hardware or compromising deployment efficiency. Evo-Depth employs a lightweight *Implicit Depth Encoding Module* (IDEM) to extract compact depth features from multi-view RGB images. These features are incorporated into vision-language representations through a *Spatial Enhancement Module* (SEM) via depth-aware modulation, enabling efficient spatial-semantic enhancement. A *Progressive Alignment Training* strategy is further introduced to align the resulting depth-enhanced representations with downstream action learning. Extensive experiments in simulation and real-world settings demonstrate the effectiveness of Evo-Depth. With only **0.9B** parameters, Evo-Depth achieves superior performance across four simulation benchmarks. In real-world experiments, Evo-Depth attains the highest average success rate while also exhibiting the smallest model size, lowest GPU memory usage, and highest inference frequency among compared methods. These results demonstrate that lightweight implicit depth enhancement is an effective and practical solution for spatially grounded robotic manipulation.

1 Introduction

Recent advances in embodied intelligence have increasingly emphasized the importance of unifying perception, language, and control for general-purpose robotic manipulation, giving rise to Vision-

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

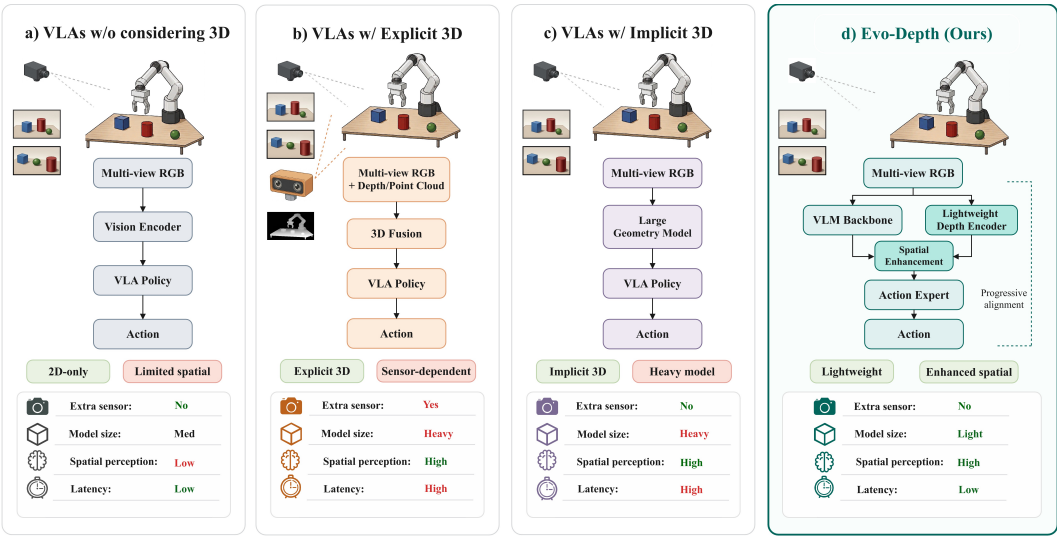


Figure 1: **Comparison of Evo-Depth with three representative VLA paradigms.** (a) 2D-only VLAs are efficient but lack spatial perception. (b) Explicit 3D VLAs offer strong spatial understanding at the cost of extra sensors and high latency. (c) Implicit 3D VLAs remove sensor dependency but rely on heavy geometry models. (d) Evo-Depth improves spatial perception with low latency by using the lightweight IDEM module to extract depth-aware features and the SEM module for efficient spatial enhancement.

Language-Action (VLA) models [2, 22, 31, 43, 51, 56]. By grounding natural language instructions in visual observations and directly generating actions, VLA models provide a promising framework for scalable robotic learning. Despite this progress, their performance still degrades noticeably in manipulation scenarios that require precise spatial understanding.

A major limitation of current VLA models [15, 17, 25] is that they rely primarily on 2D visual representations, which provide limited depth information and spatial relationships for precise manipulation. As a result, their performance often drops on tasks requiring accurate localization, fine-grained placement, and spatially consistent manipulation. To address this issue, recent studies have introduced explicit 3D information, such as depth maps or point clouds, into robotic policy learning [19, 38, 62]. However, reliance on additional sensing hardware or explicit geometry-processing pipelines often increases system complexity, making these methods sensitive to sensing noise and reconstruction errors, which in turn limits their practicality and scalability in real-world deployment. Another line of work explores implicit 3D-aware spatial modeling from RGB observations alone [1, 20, 24, 37, 55], avoiding explicit 3D sensing and providing a more scalable alternative. Nevertheless, these methods often introduce large-scale geometry foundation models, such as VGGT [46]. This increases model size and raises training and deployment costs, making them less suitable for real-time robotic deployment. At the same time, many existing fusion designs remain relatively heavy, leaving lightweight yet effective spatial-semantic fusion underexplored. In addition, how to effectively align such spatial enhancement with downstream action learning during training also remains an open challenge.

To address these challenges, we propose **Evo-Depth**, a lightweight depth-enhanced VLA framework for improving spatially grounded manipulation while preserving deployment efficiency, as illustrated in Fig. 1. Evo-Depth adopts a lightweight implicit depth enhancement strategy to strengthen spatial perception without relying on explicit 3D inputs or additional sensing hardware. It first extracts spatial information from multi-view RGB observations through a lightweight *Implicit Depth Encoding Module* (IDEM), which produces compact latent depth features that capture relative spatial relationships and object layout without constructing heavy geometry-rich scene representations. However, such information alone is insufficient to support downstream decision making, as it needs to be effectively incorporated into vision-language representations. To this end, we introduce a *Spatial Enhancement Module* (SEM), which uses implicit depth features as lightweight modulation signals to enhance visual features, rather than relying on heavy feature fusion, thereby preserving the original vision-language representations while injecting complementary spatial cues. In addition, to better support end-to-end

optimization for depth-enhanced policy learning, we propose a *Progressive Alignment Training* strategy that gradually aligns spatial representations with downstream action learning. Together, these designs enable Evo-Depth to improve spatial understanding and manipulation performance without relying on explicit 3D inputs or additional sensing hardware, while maintaining a lightweight and deployable VLA framework for effective cross-modal fusion and end-to-end policy learning.

Extensive experiments in both simulation and real-world settings demonstrate that Evo-Depth achieves superior performance while maintaining low memory usage and high inference frequency. Notably, with only 0.9B parameters, Evo-Depth achieves 84.4% on Meta-World, 41.1% on VLA-Arena, 95.4% on LIBERO, and 69.6% on LIBERO-Plus on simulation benchmarks. In real-world experiments, Evo-Depth achieves the highest average success rate of 90%, with the smallest model size, lowest GPU memory usage (3.2 GB), and highest inference frequency (12.3 Hz) among all compared methods. Moreover, under four disturbance settings in real-world generalization experiments, Evo-Depth consistently outperforms the baseline model, demonstrating improved robustness to unseen visual and spatial variations.

Our contributions are summarized as follows:

1. We propose **Evo-Depth**, a lightweight and deployment-efficient depth-enhanced VLA framework that incorporates implicit depth information to improve spatially grounded manipulation without relying on explicit 3D inputs or additional sensing hardware.
2. We introduce a *Spatial Enhancement Module* to support lightweight spatial-semantic fusion, together with a *Progressive Alignment Training* strategy that progressively aligns depth-enhanced representations with downstream action learning.
3. Extensive experiments in both simulation and real-world settings demonstrate the effectiveness of Evo-Depth. With only 0.9B parameters, Evo-Depth achieves strong performance across simulation benchmarks and attains the highest average success rate in real-world experiments, while maintaining low GPU memory usage and high inference frequency, making it well suited for real-time robotic deployment.

2 Related Work

2.1 Vision-Language-Action Models

Vision-Language-Action (VLA) models [4, 5, 9, 26, 29, 30, 32, 40, 47, 48, 63, 65] provide a unified framework that grounds language instructions in visual observations and directly generates robot actions. Recent progress has shown that large-scale multimodal pretraining combined with end-to-end policy learning can substantially improve manipulation performance [3, 50, 54]. Representative generalist VLA models include OpenVLA [17], an open-source model pretrained on large-scale real-world robot demonstrations [33]; GR00T [2], which combines a vision-language module with a diffusion transformer for real-time action generation; and $\pi^*_{0.6}$ [15], which further improves VLA performance through real-world reinforcement learning. To improve deployment efficiency, recent work has also explored lightweight VLA models. TinyVLA [49] improves inference speed and data efficiency with a compact architecture, SmolVLA [41] combines a lightweight vision-language backbone with a flow-matching action expert, and Evo-1 [25] further reduces computation while preserving semantic alignment. Despite these advances, most VLA models still rely on 2D visual observations, which limits their effectiveness in tasks requiring precise spatial understanding and highlights the need for 3D-aware or depth-informed representations.

2.2 Vision-Language-Action Models with 3D Perception

To overcome the spatial limitations of 2D-based policies, recent studies have incorporated explicit 3D information into robotic policy learning [1, 8, 10, 21, 42, 44, 53]. Representative examples include 3D-VLA [62], which unifies 3D perception, reasoning, and action in a generative model; SpatialVLA [38], which introduces 3D spatial context through Ego3D Position Encoding and structured actions; and PointVLA [19], which injects point cloud features into pretrained VLA models. These methods improve spatial understanding through richer geometric representations, but their reliance on explicit 3D inputs often requires additional sensing hardware and complex processing pipelines, increasing system complexity and sensitivity to sensing noise and reconstruction errors. Recent studies have also explored implicit 3D-aware spatial modeling from RGB observations alone [20, 24, 35, 39].

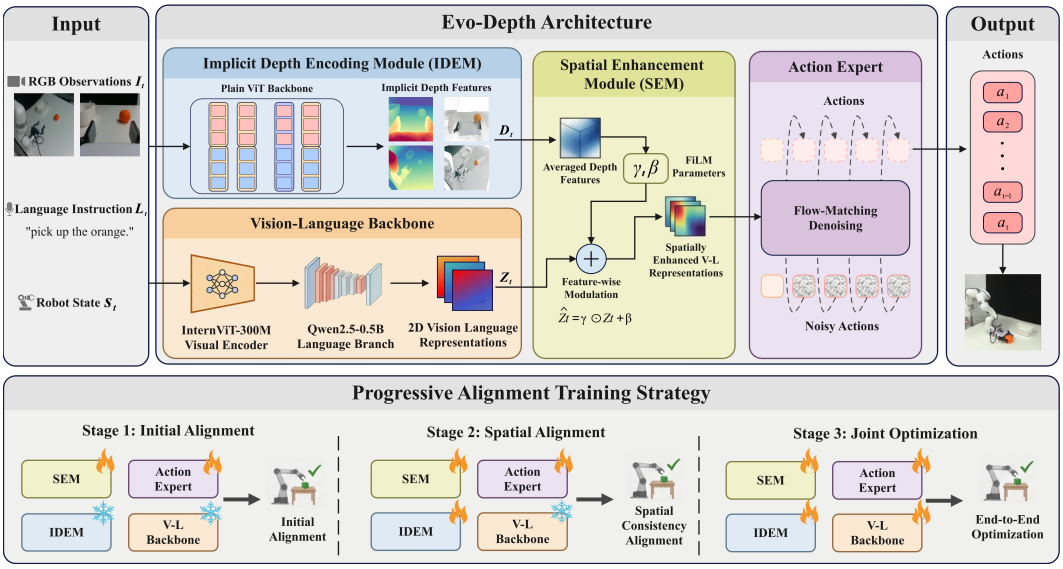


Figure 2: **Overview of Evo-Depth.** Given multi-view RGB observations, language instructions, and robot states, Evo-Depth first extracts implicit depth cues with the IDEM, then enhances vision-language representations through the SEM, and finally predicts actions with a flow-matching action expert. The bottom panel shows the progressive alignment training strategy, which improves cross-module coordination and facilitates effective fusion between implicit depth features and vision-language representations through staged optimization.

Although these methods avoid explicit 3D sensing pipelines, they are generally built upon large-scale geometry foundation models [46], resulting in larger model size and higher training and deployment costs. Together, these limitations motivate a lightweight implicit depth approach that enhances VLA models with depth-aware spatial information from RGB observations while preserving efficiency.

3 Method

3.1 Overview of Evo-Depth Architecture

Evo-Depth is a lightweight spatially enhanced Vision-Language-Action model that improves spatially grounded manipulation by incorporating implicit depth cues into policy learning, without requiring additional 3D sensors or explicit 3D inputs. As shown in Fig. 2, Evo-Depth consists of four components: (1) *Implicit Depth Encoding Module* (IDEM), (2) *Vision-Language Backbone*, (3) *Spatial Enhancement Module* (SEM), and (4) *Action Expert*.

Given multi-view RGB observations $\{I_t^i\}_{i=1}^N$, a language instruction L_t, and robot state s_t, Evo-Depth predicts the target action trajectory as

$$a_t = f_{\text{Evo-Depth}}\left(\{I_t^i\}_{i=1}^N, L_t, s_t; \theta\right), \tag{1}$$

where θ denotes all learnable parameters. Specifically, IDEM takes $\{I_t^i\}_{i=1}^N$ as input and produces implicit depth features D_t. In parallel, the vision-language backbone encodes $\{I_t^i\}_{i=1}^N$ and L_t into 2D vision-language representations Z_t. SEM then takes D_t and Z_t as input and outputs spatially enhanced representations $\hat{Z}_t$. Finally, the action expert predicts the target action trajectory a_t conditioned on $\hat{Z}_t$ and s_t.

3.2 Model Design

Implicit Depth Encoding Module. The Implicit Depth Encoding Module (IDEM) aims to extract implicit depth features directly from multi-view RGB observations, without requiring additional 3D perception sensors or explicit 3D inputs. Formally, given multi-view RGB observations $\{I_t^i\}_{i=1}^N$,

IDEM produces implicit depth features

$$D_t = f_{\text{IDEM}}\left(\{I_t^i\}_{i=1}^N\right), \tag{2}$$

where D_t denotes the latent depth-aware representation used for subsequent spatial enhancement. Concretely, IDEM is built upon a single plain Vision Transformer backbone with only 0.13B parameters. Its weights are initialized from a multi-view depth encoder [23] pretrained for joint any-view depth estimation. The pretrained encoder is learned with a minimal yet sufficient training objective for any-view depth estimation under large-scale, high-quality supervision, providing a strong initialization without relying on heavier architectural specialization. To capture spatial relationships across different observations, IDEM is designed to progressively enable information exchange across views. Let $H^{(l)}$ denote the token representations at layer l. The attention pattern is defined as

$$H^{(l+1)} = \begin{cases} \text{Attn}_{\text{within}}\left(H^{(l)}\right), & l < l_0, \\ \text{Attn}_{\text{cross/within}}\left(H^{(l)}\right), & l \geq l_0, \end{cases} \tag{3}$$

where the first l_0 layers apply self-attention independently within each view to preserve local image structure and establish stable view-specific representations, while the remaining layers alternate between within-view and cross-view attention to progressively enable information exchange across observations. Through this design, the encoder aggregates spatial cues from different views into a unified latent feature space. The latent transformer embeddings are taken as the output of IDEM and serve as the implicit depth features for subsequent processing. These features encode object layout, relative positional relationships, and cross-view spatial consistency, and are later used by the Spatial Enhancement Module to modulate the 2D vision-language representations.

Vision-Language Backbone. The Vision-Language Backbone encodes RGB observations together with the language instruction to produce 2D vision-language representations. Formally, given multi-view RGB observations $\{I_t^i\}_{i=1}^N$ and the language instruction L_t, the backbone produces

$$Z_t = f_{\text{VLB}}\left(\{I_t^i\}_{i=1}^N, L_t\right). \tag{4}$$

We instantiate this backbone with a pretrained InternVL3-1B model [66]. It is pretrained under a native multimodal paradigm, in which text corpora and diverse multimodal data are jointly optimized within a single pretraining stage, enabling strong cross-modal alignment between visual observations and task instructions. Its visual encoder adopts InternViT-300M [7], while the language branch leverages Qwen2.5-0.5B [13]. We retain only the first 14 layers of the language branch, as intermediate layers tend to exhibit stronger cross-modal alignment between visual and linguistic features, making them more suitable for downstream visuomotor control [41]. The hidden representations produced by this backbone are taken as the 2D vision-language representations Z_t of the current observation for subsequent spatial enhancement.

Spatial Enhancement Module. The Spatial Enhancement Module (SEM) is designed to inject the implicit depth features extracted by the IDEM into the 2D vision-language representations, so that the resulting representations are enriched with spatial information for downstream action prediction. Given the 2D vision-language representations Z_t and the implicit depth features D_t, the SEM first transforms the implicit depth features into a representation aligned with the hidden space of the 2D vision-language representations for feature-wise modulation. The transformed depth features are then averaged across tokens to obtain a compact global descriptor g_t. Based on this descriptor, a lightweight modulation head $f_{\text{mod}}(\cdot)$ predicts a pair of feature-wise modulation parameters, γ and β:

$$\gamma, \beta = f_{\text{mod}}(g_t), \tag{5}$$

where γ and β are used as channel-wise scaling and shifting factors, respectively. These parameters are then applied to the 2D vision-language representations in a FiLM-style manner:

$$\hat{Z}_t = \gamma \odot Z_t + \beta, \tag{6}$$

where $\odot$ denotes element-wise multiplication. The resulting outputs form the spatially enhanced vision-language representations, which are subsequently passed to the Action Expert for action generation.

Action Expert. The Action Expert is instantiated as a conditional denoising policy based on the flow-matching paradigm [27], and is implemented with a Diffusion Transformer [34] to generate the future action sequence from the spatially enhanced vision-language representations and the current

robot state. Specifically, under the joint conditioning of the spatially enhanced vision-language representations $\hat{Z}_t$ and the robot state s_t, the module learns a time-dependent vector field that progressively transforms noisy actions toward the target action sequence. Given the ground-truth action sequence A_t and a randomly sampled noise vector ϵ, the noisy action is constructed by linear interpolation:

$$A_t^\tau = \tau A_t + (1-\tau)\epsilon, \tag{7}$$

where $\tau \in [0,1]$ denotes the interpolation coefficient. The training objective then follows the flow-matching formulation:

$$\mathcal{L}^\tau(\theta) = \mathbb{E}_{p(A_t|\hat{Z}_t,s_t),\, q(A_t^\tau|A_t)} \left[\left\| v_\theta(A_t^\tau, \hat{Z}_t, s_t) - u(A_t^\tau \mid A_t) \right\|^2 \right], \tag{8}$$

where $u(A_t^\tau \mid A_t)$ denotes the target flow direction toward the ground-truth action sequence. During inference, the policy starts from a random action initialization and iteratively refines it through multiple denoising steps, ultimately producing the final executable robot actions.

3.3 Progressive Alignment Training

To better align implicit depth-enhanced spatial representations with downstream action learning, Evo-Depth adopts a progressive alignment training strategy. This strategy is motivated by two considerations: first, to gradually align newly initialized modules with pretrained components before full end-to-end optimization; second, to encourage more effective integration between implicit depth features and 2D vision-language representations.

Specifically, training is divided into three stages. In Stage 1 (Initial Alignment), we optimize the SEM and Action Expert while freezing the IDEM and the Vision-Language Backbone, establishing an initial alignment between the newly introduced modules and the pretrained representation space. In Stage 2 (Spatial Alignment), we unfreeze the IDEM while keeping the Vision-Language Backbone frozen, enabling implicit depth representations to align more effectively with 2D vision-language features and improving spatial consistency across modules. In Stage 3 (Joint Optimization), all modules are jointly optimized end-to-end, allowing the entire model to achieve full feature coordination and task-specific adaptation. By progressively establishing alignment across modules, this strategy improves representation compatibility and facilitates deeper fusion between depth-aware and vision-language features, ultimately benefiting the overall action modeling capability. Its effectiveness is further examined through an ablation study in Section 7.2, and qualitative visualization in Appendix B.3 further suggests that progressive alignment training encourages more task-focused IDEM spatial attention.

4 Simulation Experiments

Setup. We evaluate Evo-Depth on four representative simulation benchmarks, including Meta-World [58], VLA-Arena [60], LIBERO [28], and LIBERO-Plus [11], to assess manipulation performance under varying task difficulty, long-horizon execution, and distribution shifts. Detailed benchmark settings and evaluation protocols are provided in Appendix A.1.

Results. As shown in Table 1, Evo-Depth achieves state-of-the-art overall performance on Meta-World, obtaining the highest average success rate of 84.4 and the best results on the *medium*, *hard*, and *very hard* splits. On VLA-Arena, Evo-Depth delivers strong overall performance, with particularly good results in the *distractor* and *long horizon* settings. On LIBERO, Evo-Depth also performs strongly, achieving a high average success rate of 95.4 with balanced performance across different task suites. As shown in Table 2, Evo-Depth further demonstrates strong robustness on LIBERO-Plus, achieving impressive overall average and performing well across diverse perturbation dimensions. Overall, these results demonstrate that the implicit depth perception strategy effectively improves VLA performance across diverse simulation benchmarks, especially in challenging scenarios that require robust spatial perception, long-horizon execution, and generalisation under distribution shifts.

5 Real-World Experiments

Setup. We evaluate Evo-Depth on a physical robot across three real-world manipulation tasks of increasing difficulty, as illustrated in Fig. 3: *Orange Placement*, *Tennis Deposit*, and *Cup Stacking*.

Table 1: **Simulation Results.** Comparison across Meta-World, VLA-Arena, and LIBERO. "Params" denotes model parameters, and "Robo- Pretrain" indicates robotic-data pretraining. **Bold** and underline denote the best and second-best values within each benchmark; for "Params", they denote the smallest and second-smallest values. In VLA-Arena, "Extrap." and "Long" denote Extrapolation and Long Horizon, respectively.

Benchmark	Models	Params	Robo-Pretrain	Success Rate (%)				
Meta-World				**Easy**	**Medium**	**Hard**	**Very Hard**	**Avg.**
	TinyVLA [49]	1.3B	No	77.6	21.5	11.4	15.8	31.6
	π_0 [3]	3.5B	Yes	71.8	48.2	41.7	30.0	47.9
	SmolVLA [41]	2.3B	No	87.1	51.8	70.0	64.0	68.2
	RoboTron Mani [52]	4B	No	85.5	67.7	76.7	81.0	77.7
	Evo-1 [25]	**0.8B**	No	**89.2**	76.8	77.2	79.2	80.6
	Evo-Depth (Ours)	0.9B	No	83.1	**84.7**	**87.3**	**82.4**	**84.4**
VLA-Arena				**Safety**	**Distractor**	**Extrap.**	**Long**	**Total**
	SmolVLA [41]	2.3B	No	16.5	21.0	10.4	24.7	16.4
	π_0-FAST [36]	3.5B	Yes	34.1	39.0	4.2	20.7	25.6
	UniVLA [4]	7B	Yes	45.5	41.3	31.1	22.0	38.7
	OpenVLA [17]	7B	Yes	41.9	43.0	**37.8**	26.7	39.6
	OpenVLA-OFT [18]	7B	Yes	43.2	52.3	30.4	26.7	39.9
	Evo-Depth (Ours)	**0.9B**	No	**47.3**	**52.7**	25.8	**46.5**	**41.1**
LIBERO				**Spatial**	**Object**	**Goal**	**Long**	**Avg.**
	TraceVLA [64]	7B	Yes	84.6	85.2	75.1	54.1	74.8
	SpatialVLA [38]	4B	Yes	88.2	89.9	78.6	55.5	78.1
	4D-VLA [61]	4B	Yes	88.9	95.2	90.9	79.1	88.6
	SmolVLA [41]	2.3B	No	93.0	94.0	91.0	77.0	88.8
	GR00T-N1 [2]	2B	Yes	94.4	97.6	93.0	90.6	93.9
	DM0 [57]	2.4B	Yes	**98.2**	98.8	**96.6**	82.6	94.1
	π_0 [3]	3.5B	Yes	96.8	98.8	95.8	85.2	94.2
	$\pi_{0.5}$ [16]	3.5B	Yes	98.0	97.8	95.6	85.8	94.3
	VLA-0 [12]	3B	No	97.0	97.8	96.2	87.6	94.7
	BitVLA [45]	3B	Yes	97.4	**99.6**	94.4	87.7	94.8
	Evo-1 [25]	**0.8B**	No	92.7	97.7	96.3	**92.3**	94.8
	DepthVLA [59]	4.1B	No	96.4	98.0	95.8	89.2	94.9
	NORA-1.5 [14]	3B	Yes	98.0	96.0	95.4	90.5	95.0
	UniVLA [4]	7B	Yes	96.5	96.8	95.6	92.0	95.2
	Evo-Depth (Ours)	0.9B	No	95.6	99.2	95.6	91.3	**95.4**

Table 2: **LIBERO-Plus Results.** Results in LIBERO-Plus benchmark. "Bkgd." denotes Background. **Bold** indicates the best result in each column, and underlined indicates the second-best result.

Model	Camera	Robot	Language	Light	Bkgd.	Noise	Layout	Avg.
WorldVLA [6]	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
NORA [14]	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA [4]	1.8	46.2	69.6	69.0	81.0	21.2	31.9	43.9
π_0-Fast [36]	13.8	6.0	58.8	85.0	**81.4**	**79.0**	68.9	53.6
Evo-Depth (Ours)	**47.2**	**49.2**	**78.9**	**88.1**	76.4	77.6	**69.6**	**69.6**

These tasks impose progressively stronger requirements on spatial perception and fine-grained manipulation. Notably, the third-view camera is partially occluded during critical manipulation stages, making the setup particularly challenging for accurate grasping and placement. For each task, we conduct 20 trials, and the success rate is computed as the ratio of successful trials to the total number of trials. Detailed task definitions and real-world evaluation protocols are provided in Appendix B.1.

Results. Evo-Depth achieves the strongest overall performance across all three real-world tasks. As shown in Table 3, the performance gap among the three methods becomes more evident on tasks with higher spatial complexity. In particular, tasks involving target motion and precise object-object interaction are more sensitive to the model's spatial perception quality. This trend suggests that real-world manipulation performance is strongly related to how well a model can capture object layout and relative positions from visual observations. The results further indicate that introducing implicit depth cues into VLA learning can provide more informative spatial representations, which is beneficial for more accurate action generation in challenging real-world settings. We also conduct a detailed failure case analysis for the three real-world tasks, which is provided in the Appendix B.2.

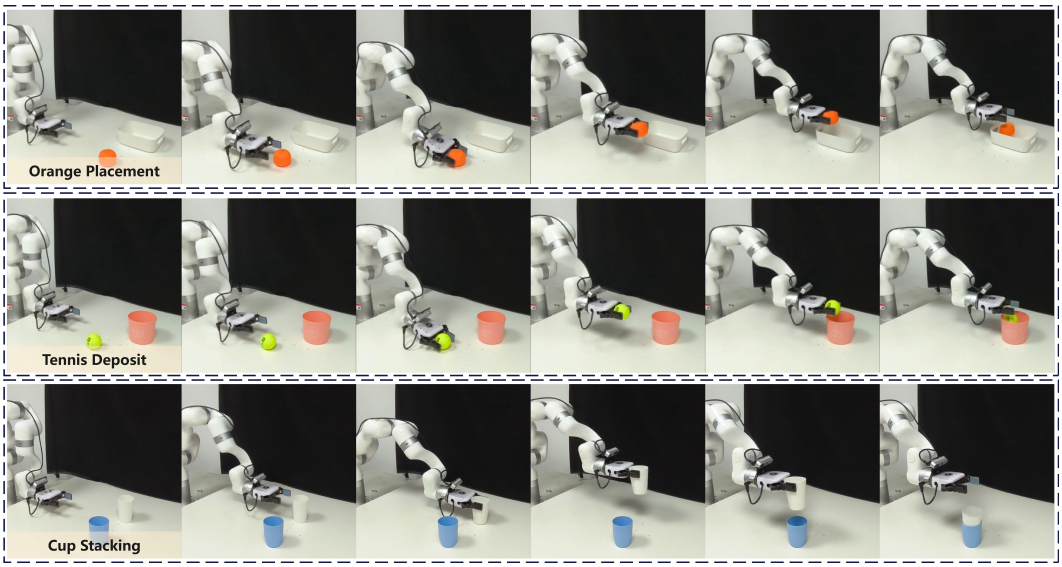


Figure 3: **Real-World Setup.** Real-world experiment process for three tasks, *Orange Placement*, *Tennis Deposit*, and *Cup Stacking*, with increasing manipulation difficulty.

Table 3: **Real-Robot Comparison.** Inference efficiency and task-wise success rate comparison of different VLA models in real-robot experiments. **Bold** and underline denote the best and second-best values, respectively, except for "Params" and "GPU Mem.", where they denote the smallest and second-smallest values. All task-wise success rates and average success are reported in percent.

Model	Params (B)	GPU Mem. (GB)	Infer. Freq. (Hz)	Orange Placement	Tennis Deposit	Cup Stacking	Avg. Success (%)
OpenVLA-OFT [18]	7.0	15.1	7.8	50.0	40.0	25.0	38.3
π_0 [3]	3.5	17.9	10.2	75.0	70.0	55.0	66.7
Evo-Depth (Ours)	**0.9**	**3.2**	**12.3**	**95.0**	**90.0**	**85.0**	**90.0**

Inference Efficiency Analysis. We further compare baseline VLA models in terms of parameter size, GPU memory usage, inference frequency, and average success rate in real-robot experiments. As shown in Table 3, Evo-Depth achieves the best overall trade-off between efficiency and performance. Despite using only 0.9B parameters and 3.2 GB GPU memory, Evo-Depth runs at 12.3 Hz and attains an average success rate of 90.0%, significantly outperforming OpenVLA-OFT and π_0 in real-world manipulation tasks.

6 Generalization Experiments

Setup. To evaluate the real-world generalization ability of Evo-Depth, we further test it under four types of distribution shifts and environmental disturbances using the same three tasks as in the real-world experiments: *Orange Placement*, *Tennis Deposit*, and *Cup Stacking*. As illustrated in Fig. 4, these settings cover appearance changes, distractor interference, and spatial perturbations in both horizontal and vertical dimensions. For each task under each disturbance setting, we conduct 20 trials, and the success rate is computed as the ratio of successful trials to the total number of trials. Detailed disturbance settings are provided in Appendix C.1.

Results. As shown in Fig. 5a, Evo-Depth consistently achieves higher success rates than the baseline under all four disturbance settings across the three real-world tasks. These results demonstrate that Evo-Depth generalizes more effectively to unseen visual and spatial variations. In particular, its advantage becomes more significant in challenging settings that require accurate spatial reasoning, such as position perturbations and precise object-object interaction. This suggests that implicit depth-aware representations enhance spatial perception and improve the robustness of VLA policies under real-world distribution shifts.

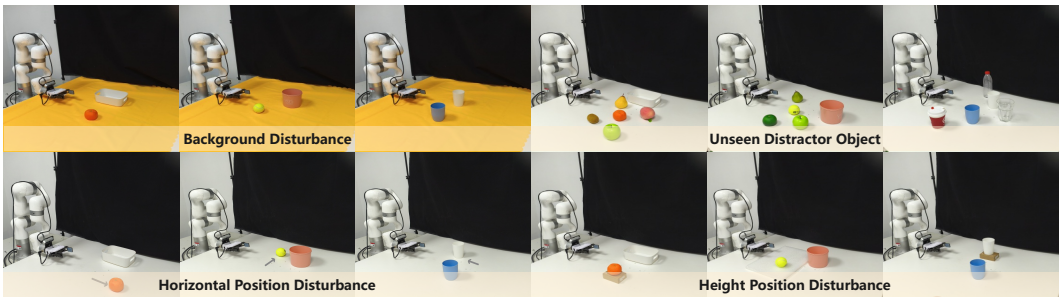


Figure 4: **Generalization Setup.** Generalization experiments under four disturbance settings: background disturbance, unseen distractors, horizontal and height position disturbance.

7 Ablation Studies

7.1 Implicit Depth Perception Analysis

Ablation Setup. To evaluate the contribution of implicit depth perception, we remove the implicit depth perception module from Evo-Depth while keeping the remaining architecture and training pipeline unchanged. We conduct this ablation on LIBERO-Plus, and detailed evaluation settings are provided in Appendix D.1.

Results Analysis. As shown in Fig. 5b, Evo-Depth consistently outperforms the variant without the implicit depth perception module across all four evaluation dimensions. The gains are particularly clear on *camera*, *robot*, and *layout*, indicating that implicit depth cues are especially helpful for handling viewpoint changes, embodiment variation, and object arrangement. Overall, these results verify the effectiveness of implicit depth perception in improving spatial perception and manipulation robustness.

7.2 Multi-Stage Fusion Training

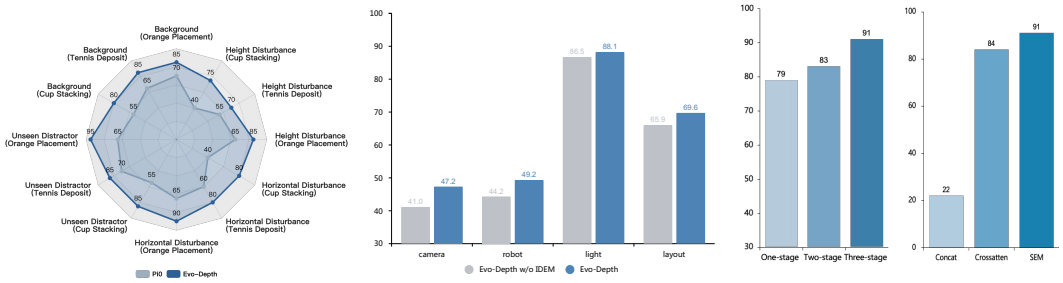
Ablation Setup. To evaluate the proposed Progressive Alignment Training strategy, we compare one-stage, two-stage, and three-stage training variants on the LIBERO Long benchmark. Detailed stage definitions are provided in Appendix D.1.

Results Analysis. The results consistently verify the effectiveness of the proposed Progressive Alignment Training strategy. As shown in Fig. 5c, the three-stage variant achieves the best performance, outperforming both the one-stage and two-stage variants. This result suggests that implicit depth cues are more effectively incorporated into the vision-language-action framework when cross-module alignment is introduced progressively rather than through abrupt joint optimization. In particular, while the two-stage variant already improves over one-stage training, the additional intermediate alignment stage in the three-stage variant further strengthens multimodal integration and leads to the best overall performance.

7.3 Fusion Strategy Analysis

Ablation Setup. To investigate the impact of different fusion strategies, we compare concatenation, cross-attention, and SEM-based modulation for integrating implicit depth features into the 2D vision-language representations on the LIBERO Long benchmark. Detailed variant definitions are provided in Appendix D.1.

Results Analysis. As shown in Fig. 5d, different fusion strategies lead to markedly different performance. The *Concat* variant performs substantially worse than the other two variants, suggesting that treating implicit depth features and 2D vision-language representations in a symmetric manner is not effective for downstream action prediction. In contrast, both *Crossatten* and *SEM* achieve much stronger results, indicating that implicit depth cues are more useful when they are incorporated as complementary spatial signals conditioned on the 2D vision-language representations. Among them, *SEM* achieves the best performance, suggesting that FiLM-style feature-wise modulation provides a



(a) (b) (c) (d)

Figure 5: **Generalization and Ablation Studies on Evo-Depth.** (a) Generalization performance of Pi0 and Evo-Depth across multiple perturbations. (b) Effect of the implicit depth encoding module on LIBERO-Plus under four evaluation dimensions. (c) Effect of different alignment training stages on LIBERO Long benchmark. (d) Effect of different fusion strategies on LIBERO Long benchmark.

more effective way to inject spatial information while preserving the dominant role of the original 2D vision-language representations.

8 Conclusion

We introduced Evo-Depth, a lightweight depth-enhanced VLA framework that improves spatially grounded robotic manipulation while preserving deployment efficiency without relying on additional sensing hardware. Evo-Depth integrates an Implicit Depth Encoding Module, a Spatial Enhancement Module, and Progressive Alignment Training to enhance depth-aware spatial understanding for downstream action learning. Experiments in simulation and real-world settings demonstrate its effectiveness, robustness, and deployment efficiency. These results suggest that lightweight implicit depth enhancement is an effective direction for practical VLA systems.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science and Technology Major Project (Grant No. 2025ZD1606203) and the National Natural Science Foundation of China (NSFC; Grant Nos. 62676236 and 62306046). The authors declare no competing interests.

References

[1] Ali Abouzeid, Malak Mansour, Qinbo Sun, Zezhou Sun, and Dezhen Song. Geoaware-vla: Implicit geometry aware vision-language-action model. *arXiv preprint arXiv:2509.14117*, 2025.

[2] 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.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π0: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.

[4] 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.

[5] Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026.

[6] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

[7] 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.

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

[9] Yuxin Du, Tao Lin, Zile Zhong, Runting Li, Xiyao Chen, Jiting Liu, Chenglin Liu, Ying-Cong Chen, Yuqian Fu, and Bo Zhao. Focusable monocular depth estimation, 2026. URL `https://arxiv.org/abs/2605.11756`.

[10] Xianzhe Fan, Shengliang Deng, Xiaoyang Wu, Yuxiang Lu, Zhuoling Li, Mi Yan, Yujia Zhang, Zhizheng Zhang, He Wang, and Hengshuang Zhao. Any3d-vla: Enhancing vla robustness via diverse point clouds. *arXiv preprint arXiv:2602.00807*, 2026.

[11] 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.

[12] Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. Vla-0: Building state-of-the-art vlas with zero modification. *arXiv preprint arXiv:2510.13054*, 2025.

[13] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[14] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

[15] 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.

[16] 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. *arXiv preprint arXiv:2504.16054*, 2025.

[17] 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.

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

[19] 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.

[20] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

[21] Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. Qdepth-vla: quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025.

[22] Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. Langforce: Bayesian decomposition of vision language action models via latent action queries. *arXiv e-prints*, pages arXiv–2601, 2026.

[23] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[24] Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, Yuxin Du, Jiting Liu, Encheng Gu, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *arXiv preprint arXiv:2507.00416*, 2025.

[25] 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.

[26] Tao Lin, Yuxin Du, Yiran Mao, Zewei Ye, Yilei Zhong, Bing Cheng, Yiming Wang, Jiting Liu, Yang Tian, Junchi Yan, Feiran Wu, Zenan Meng, Hu Wei, Yuqian Fu, Gen Li, and Bo Zhao. La4vla: Learning to act without seeing via language-action pretraining, 2026. URL `https://arxiv.org/abs/2606.27295`.

[27] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[28] 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.

[29] Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Bridging the semantic-action gap in visual token pruning for efficient vla inference. *arXiv preprint arXiv:2511.16449*, 2025.

[30] 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.

[31] Yuankai Luo, Woping Chen, Tong Liang, and Zhenguo Li. Coral: Scalable multi-task robot learning via lora experts. *arXiv preprint arXiv:2603.09298*, 2026.

[32] Yuankai Luo, Woping Chen, Tong Liang, Baiqiao Wang, and Zhenguo Li. Simvla: A simple vla baseline for robotic manipulation. *arXiv preprint arXiv:2602.18224*, 2026.

[33] 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.

[34] 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.

[35] Haosong Peng, Hao Li, Yalun Dai, Yushi Lan, Yihang Luo, Tianyu Qi, Zhengshen Zhang, Yufeng Zhan, Junfei Zhang, Wenchao Xu, et al. Omnivggt: Omni-modality driven visual geometry grounded transformer. *arXiv preprint arXiv:2511.10560*, 2025.

[36] 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.

[37] Quanhao Qian, Guoyang Zhao, Gongjie Zhang, Jiuniu Wang, Ran Xu, Junlong Gao, and Deli Zhao. Gp3: A 3d geometry-aware policy with multi-view images for robotic manipulation. *arXiv preprint arXiv:2509.15733*, 2025.

[38] 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.

[39] Zhifeng Rao, Wenlong Chen, Lei Xie, Xia Hua, Dongfu Yin, Zhen Tian, and F Richard Yu. Augvla-3d: Depth-driven feature augmentation for vision-language-action models. *arXiv preprint arXiv:2602.10698*, 2026.

[40] Xiang Shi, Wenlong Huang, Menglin Zou, and Xinhai Sun. Saivla-0: Cerebrum–pons–cerebellum tripartite architecture for compute-aware vision-language-action. *arXiv preprint arXiv:2603.08124*, 2026.

[41] 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.

[42] Bin Tan, Changjiang Sun, Xiage Qin, Hanat Adai, Zelin Fu, Tianxiang Zhou, Han Zhang, Yinghao Xu, Xing Zhu, Yujun Shen, et al. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026.

[43] GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv preprint arXiv:2510.19430*, 2025.

[44] An Dinh Vuong, Minh Nhat Vu, and Ian Reid. Improving robotic manipulation with efficient geometry-aware vision encoder. *arXiv preprint arXiv:2509.15880*, 2025.

[45] Hongyu Wang, Chuyan Xiong, Ruiping Wang, and Xilin Chen. Bitvla: 1-bit vision-language-action models for robotics manipulation. *arXiv preprint arXiv:2506.07530*, 2025.

[46] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[47] Runze Wang, Yuqian Fu, Yu Li, Tao Lin, Tianwen Qian, Mohamed Elhoseiny, Bo Zhao, Yanwei Fu, Yu-Gang Jiang, and Xiangyang Xue. Afford-vla: Action-aligned visual planning via internalized affordance. *arXiv preprint arXiv:2605.24203*, 2026.

[48] 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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pages 18638–18646, 2026.

[49] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

[50] 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.

[51] Bin Xie, Erjin Zhou, Fan Jia, Hao Shi, Haoqiang Fan, Haowei Zhang, Hebei Li, Jianjian Sun, Jie Bin, Junwen Huang, et al. Dexbotic: Open-source vision-language-action toolbox. *arXiv preprint arXiv:2510.23511*, 2025.

[52] Feng Yan, Fanfan Liu, Liming Zheng, Yufeng Zhong, Yiyang Huang, Zechao Guan, Chengjian Feng, and Lin Ma. Robotron-mani: All-in-one multimodal large model for robotic manipulation. *arXiv preprint arXiv:2412.07215*, 2024.

[53] Sizhe Yang, Linning Xu, Hao Li, Juncheng Mu, Jia Zeng, Dahua Lin, and Jiangmiao Pang. Robo3r: Enhancing robotic manipulation with accurate feed-forward 3d reconstruction. *arXiv preprint arXiv:2602.10101*, 2026.

[54] Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

[55] Bin Yu, Shijie Lian, Xiaopeng Lin, Zhaolong Shen, Yuliang Wei, Haishan Liu, Changti Wu, Hang Yuan, Bailing Wang, Cong Huang, et al. 3d-mix for vla: A plug-and-play module for integrating vggt-based 3d information into vision-language-action models. *arXiv preprint arXiv:2603.24393*, 2026.

[56] Bin Yu, Shijie Lian, Xiaopeng Lin, Yuliang Wei, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Xinming Wang, Bailing Wang, Cong Huang, et al. Twinbrainvla: Unleashing the potential of generalist vlms for embodied tasks via asymmetric mixture-of-transformers. *arXiv preprint arXiv:2601.14133*, 2026.

[57] En Yu, Haoran Lv, Jianjian Sun, Kangheng Lin, Ruitao Zhang, Yukang Shi, Yuyang Chen, Ze Chen, Ziheng Zhang, Fan Jia, et al. Dm0: An embodied-native vision-language-action model towards physical ai. *arXiv preprint arXiv:2602.14974*, 2026.

[58] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[59] 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.

[60] Borong Zhang, Jiahao Li, Jiachen Shen, Yishuai Cai, Yuhao Zhang, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. Vla-arena: An open-source framework for benchmarking vision-language-action models. *arXiv preprint arXiv:2512.22539*, 2025.

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

[62] 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. *arXiv preprint arXiv:2403.09631*, 2024.

[63] 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.

[64] 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.

[65] Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. *arXiv preprint arXiv:2601.11404*, 2026.

[66] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Simulation Experiment Details

A.1 Simulation Benchmark Setup

We evaluate Evo-Depth on four complementary simulation benchmarks to assess its effectiveness from different perspectives, including base manipulation competence, VLA-oriented generalization, cross-task transfer, and robustness under perturbations. For all benchmarks, we report success rates and compare Evo-Depth with representative VLA baselines under aligned evaluation settings. All baseline results are derived from the original paper or official implementation. Representative task trajectories from Meta-World and LIBERO are shown in Figs. 6 and 7, respectively, to illustrate the diversity of manipulation scenes considered in these benchmarks.

Meta-World. Meta-World is a widely used multi-task robotic manipulation benchmark that contains 50 distinct manipulation environments. Its task set is intentionally diverse yet structured, covering a broad range of fundamental manipulation skills with shared underlying object-interaction patterns. We include Meta-World because it provides a broad evaluation of whether Evo-Depth improves core manipulation ability beyond narrow single-task settings. In our evaluation, we group the benchmark into four difficulty splits (*easy*, *medium*, *hard*, and *very hard*) to analyze how performance changes as tasks demand more accurate localization, spatial coordination, and fine-grained object placement. This benchmark is particularly relevant to our study because it tests whether lightweight implicit depth cues remain beneficial as manipulation difficulty increases.

VLA-Arena. VLA-Arena is a benchmark designed specifically for evaluating Vision-Language-Action models. Its original design quantifies VLA difficulty along three orthogonal axes, namely task structure, language command, and visual observation, and organizes 170 tasks into four task-structure dimensions: *safety*, *distractor*, *extrapolation*, and *long horizon*. In particular, the *distractor* and *long horizon* dimensions are closely related to the ability to maintain stable spatial grounding over cluttered observations and multi-stage action sequences, while *extrapolation* probes generalization beyond seen conditions. This benchmark is particularly relevant to our study because it directly tests whether the proposed implicit-depth enhancement improves robustness and spatial reasoning in challenging VLA settings without sacrificing the lightweight nature of the policy.

LIBERO. LIBERO is used in our evaluation through four categories, *spatial*, *object*, *goal*, and *long*, which respectively cover manipulation settings with different emphases on spatial reasoning, object understanding, goal-conditioned execution, and long-horizon task composition. We include LIBERO because it provides a balanced evaluation across multiple types of language-conditioned manipulation settings, rather than emphasizing only a single source of difficulty. This benchmark is particularly relevant to our study because it allows us to examine model behavior under different task demands using a unified evaluation setup. In particular, the *spatial* and *long* categories are useful for analyzing settings that place relatively higher demands on spatial grounding and sustained action coordination over time.

LIBERO-Plus. LIBERO-Plus extends LIBERO with a dedicated robustness benchmark for VLA models. It systematically introduces perturbations along seven dimensions, including *objects layout*, *camera viewpoints*, *robot initial states*, *language instructions*, *light conditions*, *background textures*, and *sensor noise*, in order to expose failure modes that may be hidden by standard benchmark scores. We include LIBERO-Plus because one of the main motivations of Evo-Depth is to strengthen spatial perception from RGB alone in a lightweight manner, and such an improvement should be reflected not only in in-distribution success rates but also in robustness to visual and spatial perturbations. In our evaluation, we report results on *camera*, *robot*, *language*, *light*, *background*, *noise*, and *layout*. This benchmark is particularly relevant to our study because perturbations in viewpoint, embodiment, and scene arrangement directly test whether the implicit-depth features extracted by IDEM provide more stable spatial grounding than a purely 2D visual representation.

B Real-World Experiment Details

B.1 Real-World Experiment Setup

Task Definitions. We evaluate Evo-Depth on three real-world manipulation tasks with increasing difficulty. *Orange Placement* requires grasping an orange and placing it into a white box. *Tennis Deposit* involves picking up a tennis ball and depositing it into a small orange bucket, which is more

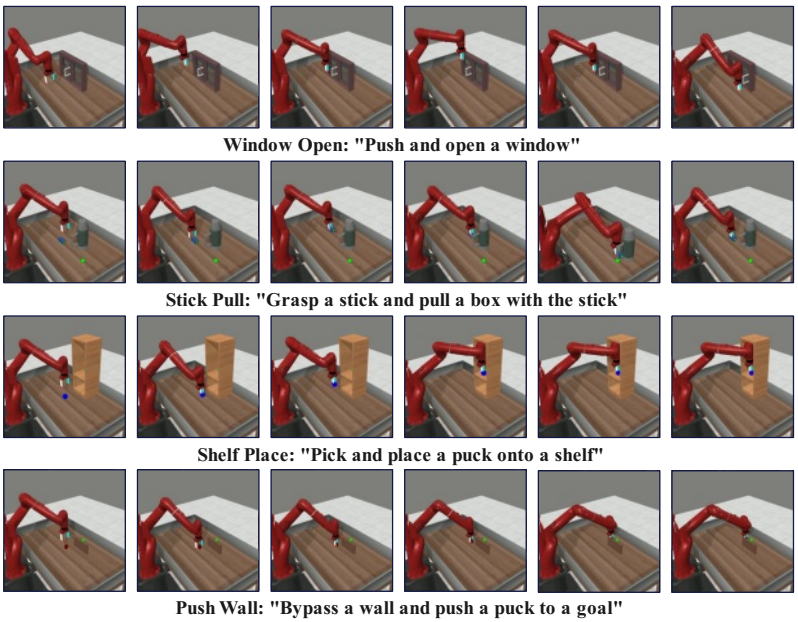


Figure 6: **Representative task trajectories from Meta-World.** We show example multi-step manipulation sequences from several Meta-World tasks to illustrate the benchmark's diversity in object interaction and spatial control.

challenging because the ball is easily disturbed and may roll away during contact. *Cup Stacking* requires stacking a white cup onto a blue cup and demands the most precise spatial understanding and fine-grained control due to the small geometric tolerance during insertion. Together, these tasks impose progressively stronger requirements on spatial perception and manipulation.

Hardware Setup. The real-world experiments are conducted on a fixed-base UFACTORY xArm6 platform. For visual observation, we use two Intel RealSense D435i RGB cameras, including one external camera mounted on a stand and one wrist-mounted camera attached near the end effector, as shown in Fig. 8. This setup provides complementary views of the workspace and the manipulation area.

Camera Configuration. The real-world setup uses two RGB views for observation, including one external scene camera and one wrist-mounted camera. During critical manipulation stages, the external scene camera is partially occluded, resulting in incomplete visual observations when accurate grasping and placement are most needed. This setting increases the difficulty of real-world manipulation and provides a challenging testbed for evaluating the spatial perception and reasoning capability of VLA models.

Evaluation Protocol. For each task, we execute the policy under the same observation and control setting and report task success rates. All compared methods are evaluated under aligned real-world conditions to ensure fair comparison. Success is determined by whether the robot completes the corresponding manipulation goal for each task.

Efficiency Measurement. We additionally compare parameter size, GPU memory usage, and inference frequency in the real-robot setting. GPU memory usage and inference frequency are measured on an NVIDIA RTX 4090D platform. These metrics are used together with task success rate to evaluate the deployment efficiency of different VLA models in real-world manipulation.

B.2 Failure Case Analysis

We further analyze the typical failure cases in the three real-world tasks. In *Orange Placement*, failures mainly occur during grasping under partial camera occlusion, where the gripper may collide with the orange and trigger collision protection. In *Tennis Deposit*, failures are primarily caused by the tennis ball being easily disturbed and rolling away when the gripper fails to enclose it securely.

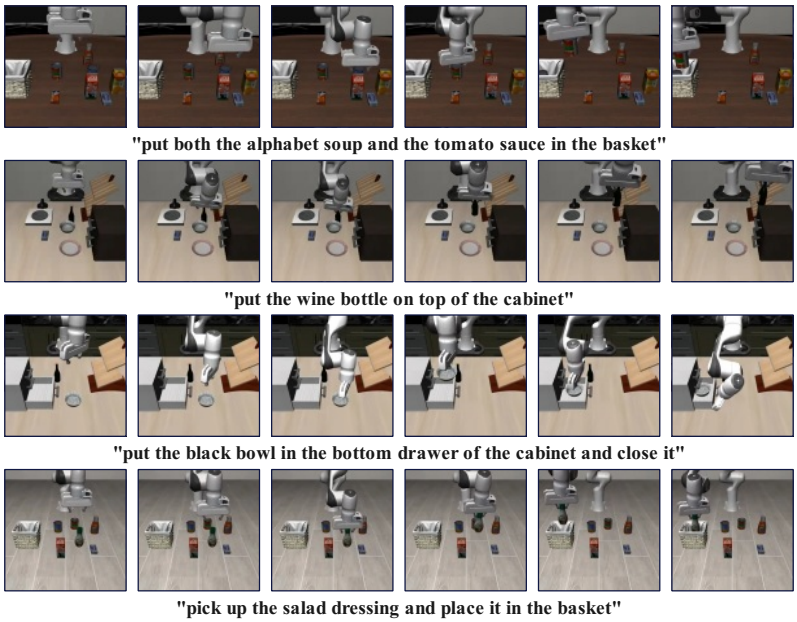


Figure 7: **Representative task trajectories from LIBERO.** We show example task sequences from LIBERO to illustrate the benchmark's language-conditioned manipulation settings and varied object arrangements.

In *Cup Stacking*, failures mainly result from the small tolerance between the two cups, where slight misalignment can lead to collision and tipping during insertion. These cases suggest that, although Evo-Depth improves spatial perception and manipulation capability, highly precise contact-rich interactions under occlusion, target motion, and tight geometric constraints remain challenging in real-world settings.

B.3 Visualization of IDEM Spatial Attention

To provide qualitative evidence on how the proposed training strategy affects spatial perception, we visualize the spatial attention patterns of IDEM on representative real-world manipulation frames. As shown in Fig. 9, the initial IDEM already captures coarse object-level structure, while the IDEM in Evo-Depth exhibits more concentrated responses around manipulation-relevant regions after progressive alignment training. In particular, the attention becomes more focused on the gripper-object interaction area and the target placement region. These visualizations provide qualitative support that Evo-Depth encourages implicit depth features to better align with downstream manipulation needs.

C Generalization Experiment Details

C.1 Generalization Experiment Setup

We evaluate real-world generalization under four disturbance settings using the same three tasks as in the main real-world experiments: *Orange Placement*, *Tennis Deposit*, and *Cup Stacking*. These disturbances are designed to cover appearance changes, clutter interference, and spatial perturbations in both horizontal and vertical directions. For each disturbance setting, the task objective and evaluation criterion remain the same as in the standard real-world setup, while only the corresponding environmental factor is changed. Fig. 4 illustrates representative examples of the four disturbance settings across the three tasks.

Background Disturbance. To introduce an appearance change without altering the task itself, we place a yellow tablecloth on the work surface while keeping the task objects and goal configuration unchanged. This setting changes the visual appearance of the scene and tests whether the policy relies too strongly on the original scene color and background cues.



Figure 8: **Real-world hardware setup.** The real-world platform uses a fixed-base UFACTORY xArm6 robot with two Intel RealSense D435i RGB cameras, including one external camera and one wrist-mounted camera.

Unseen Distractor Object. We add several task-irrelevant objects into the workspace, making the scene more cluttered and increasing visual interference around the target objects. This setting tests whether the policy can maintain stable attention to the task-relevant objects and preserve correct action generation when additional objects are present in the scene.

Horizontal Position Disturbance. We perturb the horizontal position of the target object or related task object within the workspace, so that the relative left-right or front-back spatial relationship differs from the standard setup. This setting tests whether the policy can adapt to horizontal spatial changes and still produce accurate grasping and placement behaviors when the object arrangement is shifted.

Height Position Disturbance. We vary the height of the target object or related task object, resulting in a different vertical spatial relationship from the standard setup. This setting is intended to test whether the policy can handle changes in object height during grasping, alignment, and placement, which require more precise spatial judgment in the vertical direction.

D Ablation Study Details

D.1 Ablation Setup

Implicit Depth Perception Ablation. To evaluate the contribution of implicit depth perception, we remove the implicit depth perception module from Evo-Depth while keeping the remaining architecture and training pipeline unchanged. We conduct this ablation on LIBERO-Plus and report results on four representative dimensions: *camera*, *robot*, *light*, and *layout*. These dimensions are chosen because they are closely related to spatial perception and provide a suitable test suite for assessing how implicit depth cues affect robustness under viewpoint, embodiment, illumination, and scene arrangement variations.

Progressive Alignment Training Ablation. To evaluate the proposed Progressive Alignment Training strategy, we compare one-stage, two-stage, and three-stage training variants on the LIBERO

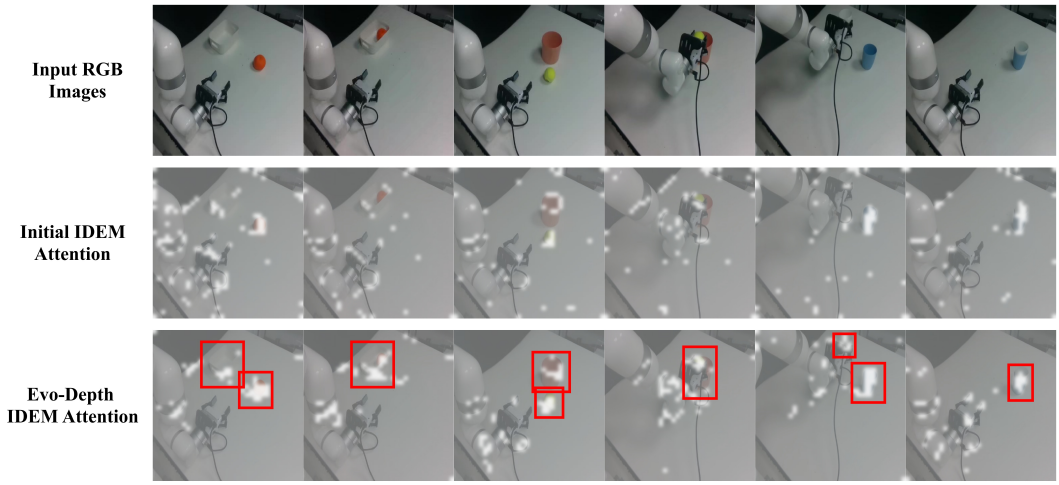


Figure 9: **Visualization of IDEM spatial attention.** The first row shows input RGB observations from representative stages of real-world manipulation. The second row shows attention maps from the initial IDEM initialized from the depth encoder. The third row shows attention maps from the IDEM in Evo-Depth after progressive alignment training. Compared with the initial IDEM, the trained IDEM exhibits more focused responses on the gripper-object interaction region and the target placement area, suggesting improved alignment between implicit depth features and task-relevant spatial cues.

Long benchmark. In the one-stage variant, all modules are jointly optimized from the beginning. In the two-stage variant, Stage 1 trains only SEM and the Action Expert while IDEM and the Vision-Language Backbone are frozen, and Stage 2 unfreezes the entire model for end-to-end optimization. In the three-stage variant, Stage 1 also trains only SEM and the Action Expert, Stage 2 additionally unfreezes IDEM while keeping the Vision-Language Backbone frozen, and Stage 3 unfreezes the entire model for end-to-end optimization. This ablation examines whether more gradual cross-module alignment improves long-horizon manipulation performance.

Fusion Strategy Ablation. To investigate the impact of different fusion strategies, we compare three ways of integrating the implicit depth features extracted by IDEM into the 2D vision-language representations. *Concat* directly concatenates the depth features with the 2D representations before the Action Expert. *Crossatten* uses the 2D representations as queries and the depth features as keys and values for cross-attention fusion. *SEM* first aligns the depth features to the hidden space of the 2D representations and then injects them through FiLM-style feature-wise modulation. This ablation examines how different fusion mechanisms affect the use of implicit depth cues for action prediction.

E Implementation Details and Reproducibility

Unless otherwise specified, Evo-Depth is trained with the three-stage progressive alignment strategy described in Section 3. In Stage 1, only the SEM and Action Expert are optimized, while the vision-language backbone and IDEM remain frozen. In Stage 2, IDEM is unfrozen and jointly optimized with the SEM and Action Expert, while the vision-language backbone remains frozen. In Stage 3, all modules are jointly optimized end-to-end. For sequential training, Stage 2 is initialized from the final checkpoint of Stage 1, and Stage 3 is initialized from the final checkpoint of Stage 2.

Table 4 lists a representative set of training hyperparameters for the Meta-World setting. The remaining simulation benchmarks follow the same overall training recipe unless otherwise specified, with benchmark-dependent dataset configurations and stage lengths. Across all stages, we use AdamW with a linear warmup followed by cosine learning-rate decay.

Table 4: **Representative Training Hyperparameters on Meta-World.**

Setting	Value
Vision-language backbone	InternVL3-1B
Action head	Flow matching
Input image size	448
Data augmentation	Enabled
Batch size	16
Optimizer	AdamW
Learning rate	1×10^{-5}
Learning-rate schedule	Linear warmup + cosine decay
Warmup steps	1,000
Weight decay	1×10^{-3}
Gradient clipping norm	1.0
Action horizon	50
Action dimension	24
State dimension	24
Action head layers	8
Dropout	0.2
Data-loading workers	4
Stage 1 steps	5,000
Stage 2 steps	10,000
Stage 3 steps	120,000
Stage 1 trainable modules	SEM + Action Expert
Stage 2 trainable modules	IDEM + SEM + Action Expert
Stage 3 trainable modules	VLM + IDEM + SEM + Action Expert

Table 5: **External assets and licenses.**

Asset	License / terms
InternVL3-1B	Apache-2.0
Depth Anything 3 (DA3-BASE)	Apache-2.0
Meta-World	MIT
VLA-Arena	Apache-2.0
LIBERO	MIT
LIBERO-Plus	MIT

F Compute Resources

Unless otherwise specified, training for both simulation and real-world settings was conducted on NVIDIA A100 GPUs. As a representative example, Meta-World training used 8 GPUs under the three-stage training pipeline described above. The remaining training runs follow the same overall hardware setting, with benchmark-dependent variation in dataset scale and training length.

For real-robot deployment, inference efficiency is measured on an NVIDIA RTX 4090D platform, as reported in the main paper.

G Assets and Licenses

We build on several publicly available models and benchmarks. We cite the corresponding original papers in the main text and use these assets under their stated licenses or benchmark terms. The main external assets used in this work are summarized in Table 5.

H Limitations

Our current real-world evaluation focuses on representative tabletop manipulation tasks. While these tasks cover object grasping, placement, and stacking scenarios with different spatial demands, future work can further broaden the evaluation to a wider range of manipulation tasks and scene configurations.

In addition, this study focuses on fixed-base manipulation rather than broader embodied settings such as mobile manipulation or whole-body interaction. We view this as a scope choice consistent with

the goal of studying lightweight depth-enhanced spatial perception for manipulation, and extending the framework to more general embodied platforms is an interesting direction for future work.

I Broader Impacts

This work may contribute to more efficient and practical robotic manipulation by improving spatially grounded policy learning under lightweight RGB-only settings. In the long term, such systems may benefit application scenarios such as laboratory automation, industrial assistance, and other structured manipulation environments where reliable and efficient robot operation is important.

At the same time, if such systems are deployed without sufficient safeguards, errors in perception or action prediction may lead to failed grasps or object damage. We therefore believe that real-world deployment should still be accompanied by appropriate safety constraints and human oversight.

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: The main claims in the abstract and introduction are consistent with the method description and experimental evidence reported in Sections 1, 3, 4, 5, 6, and 7. The paper claims improved spatially grounded manipulation with a lightweight depth-enhanced VLA, and these claims are matched to the reported simulation and real-world results.

   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: The paper discusses practical limitations through the real-world failure case analysis and a dedicated limitations discussion in the supplemental material. These discussions clarify the current task scope, remaining failure modes, and deployment constraints of the proposed method.

   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 present theoretical theorems, lemmas, or formal proofs; it is an empirical systems paper centered on model design and experimental evaluation.

   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: Section 3 and Appendices A-E describe the model architecture, progressive training strategy, and experimental protocols. The public project repository provides implementation and evaluation instructions, together with links to pretrained simulation checkpoints, to support verification of the reported simulation results.

   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: [Yes]

   Justification: Code, training and evaluation instructions, and pretrained simulation checkpoints are publicly available through `https://github.com/MINT-SJTU/Evo-Depth` and the linked model repositories. The repository documents evaluation on LIBERO, LIBERO-Plus, MetaWorld, and VLA-Arena.

   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 paper reports the main experimental setup in Sections 4–7 and the appendix, and the supplemental material provides the detailed training and evaluation settings, including hyperparameters, optimization setup, and benchmark-specific protocols required to understand the reported results.

   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: The paper reports success rates and repeated real-world trials, but it does not include error bars, confidence intervals, or formal statistical significance tests for the main experimental results. We therefore do not claim statistical significance beyond the reported evaluation outcomes.

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 real-robot inference platform is described in the paper, and the supplemental material provides the compute resources used for training and evaluation, including hardware type, memory-related information, and execution-time details needed to reproduce the experiments.

   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: The authors have reviewed the NeurIPS Code of Ethics and, to the best of our knowledge, the research described in this paper conforms to its requirements. The work involves standard model training and robot evaluation in controlled settings, without crowdsourcing or human-subject experiments.

   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: The supplemental material includes a broader-impact discussion covering both potential benefits and risks of improved robotic manipulation policies. This discussion addresses possible positive impacts in automation and assistance, as well as safety and misuse concerns in real-world deployment.

    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 does not introduce a high-risk generative model, a scraped Internet-scale dataset, or another asset category that would typically require special release safeguards under this checklist item.

    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: The original sources of the external models and benchmarks are cited in the paper. Appendix G and Table 5 summarize the licenses or terms of the main external assets used in this work.

    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: [Yes]

    Justification: The public project README documents installation, training, checkpoint use, and simulation evaluation alongside the released code. It also links to pretrained checkpoints for the simulation benchmarks.

    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: The paper does not involve crowdsourcing, human annotation studies, or experiments with human participants.

    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: The paper does not include crowdsourcing or human-subject research, so IRB approval or an equivalent review was not required for the reported experiments.

    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 methodological contribution of this work does not rely on LLMs as an important, original, or non-standard component under this checklist definition. The paper uses standard pretrained vision-language backbones as part of the model architecture and describes them in the method section.

    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.