GIVLA: Deep Geometry Internalization for A Lightweight VLA via Geometry Instruction and Gradient-Informed Training

Yunhe Li*
School of Automation and Intelligent Sensing
Shanghai Jiao Tong University
Shanghai, 200240, China
lyh_5321@sjtu.edu.cn

Qiming Liu*
School of Automation and Intelligent Sensing
Shanghai Jiao Tong University
Shanghai, 200240, China
qimingliu@sjtu.edu.cn

Haoyuan Wang*
School of Automation and Intelligent Sensing
Shanghai Jiao Tong University
Shanghai, 200240, China
wanghaoyuan@sjtu.edu.cn

Hesheng Wang†
School of Automation and Intelligent Sensing
Shanghai Jiao Tong University
Shanghai, 200240, China
wanghesheng@sjtu.edu.cn

Abstract

Vision-Language-Action (VLA) models often struggle with precise manipulation due to their lack of explicit spatial awareness. To bridge this gap, we propose **GIVLA**, a framework that internalizes geometric priors into the VLA backbone through a deep-coupled architecture. GIVLA implements a two-stage gradient-informed training paradigm to resolve task-level interference and ensure precise execution, a design grounded in the perception-action duality identified through our analysis of training dynamics. Extensive experiments on LIBERO and physical robot platforms demonstrate that GIVLA achieves superior accuracy and robustness with high parameter efficiency. Our project is available at https://github.com/Alexander-bugwriter/GIVLA.

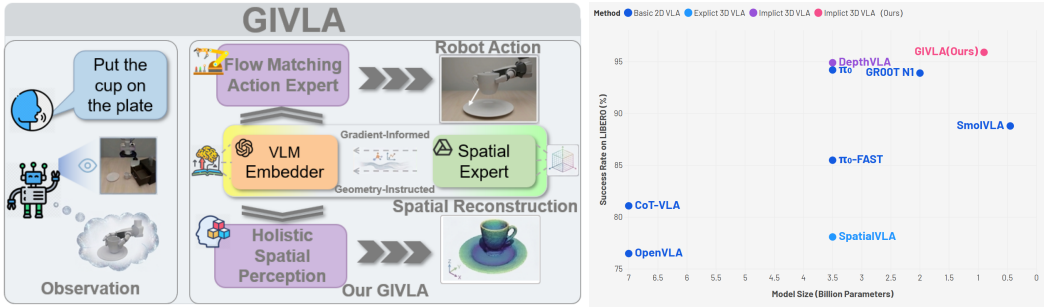
1 Introduction

Recent advances in embodied intelligence encompass robotic navigation [1] and manipulation[2]. More recently, Vision-Language-Action (VLA) models have emerged as a promising paradigm, unifying perception and control through end-to-end multimodal architectures [3, 4, 5, 6, 7]. By leveraging pre-trained Vision-Language Models (VLMs) [8, 9] as decision backbones, VLAs exhibit remarkable cross-scene reasoning abilities. However, a fundamental bottleneck remains: VLMs pre-trained on massive static 2D datasets often lack explicit spatial understanding, such as absolute depth scales and physical layouts, creating a structural gap with the 3D action space required for robotic manipulation.

Spatial representations underpin robotic perception and action, ranging from explicit geometric structures to neural representations and foundation models [10]. Learning spatial representations that support decision-making has also been explored in visual navigation[11]. For VLA-based manipulation, recent studies have explored bridging the gap between 2D visual-semantic representations and the 3D physical world. One prominent direction is **Explicit 3D data injection**[12, 13, 14], which

*These authors contributed equally to this work.
†Corresponding author.

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



(a) GIVLA framework overview.

(b) Performance vs. Efficiency.

Figure 1: **Overview of GIVLA.** (a) We establish a **Geometry-Instruct** and **Gradient-Informed** model between the pre-trained VLM and the spatial geometric expert. Through progressive Geometry-Instruct Model Architecture and Gradient-Informed Training Paradigm , we endow the VLA with holistic spatial perception and percise manipulation. (b) Performance comparison on LIBERO benchmark. Our GIVLA achieves **95.9%** success rate with only **0.9B** parameters, outperforming larger models while maintaining superior parameter efficiency.

typically incorporates 3D point clouds or depth maps directly into the visual encoding stage. While intuitive, these methods are constrained by the scarcity and high acquisition costs of high-quality 3D data. Specifically, the explicit 3D modalities often disrupt the visual-semantic consistency established during large-scale 2D pre-training, leading to significant distribution shifts[15] and degraded reasoning capabilities.

Alternatively, another pathway focuses on **Implicit spatial knowledge injection** [16, 17, 18, 19, 20, 21]. This pathway constructs spatial understanding by coupling geometric spatial experts[22, 23] with the VLA backbone. While this cost-effective methods circumvents the drawbacks of explicit 3D sensing and offers spatial data that is more aligned with the VLM's data distribution, current methods typically rely on shallow feature fitting. Consequently, they often fail to establish a deep structural integration of geometry, leaving the model's spatial reasoning fragile in complex manipulation scenarios.

To overcome the limitations of explicit 3D sensing and the shallowness of existing implicit alignment, we argue that the key lies in the deep structural internalization of geometric priors. We contend that the VLA backbone should be empowered with holistic spatial perception capabilities through the injection of geometric knowledge. Motivated by this insight, we propose **GIVLA**, a Geometry-Instructed and Gradient-Informed framework. GIVLA features a deep-coupled architecture that internalizes geometric priors from a spatial expert into the VLA backbone through progressive feature injection and supervision. Furthermore, GIVLA employs a gradient-informed two-stage training paradigm to ground precise manipulation in holistic spatial perception, effectively leveraging geometric insights to enhance action execution. This design is grounded in our rigorous analysis of training dynamics [24, 25], where we reveal a critical transition in the perception-action relationship: a shift from gradient synergy in early stages to significant gradient conflict during convergence. Our experiments on LIBERO and LIBERO-plus benchmarks demonstrate that GIVLA (0.9B) achieves competitive manipulation accuracy and superior generalization robustness against environmental disturbances.

2 Related Works

2.1 Vision Language Action Model

Vision-Language-Action (VLA) models aim to directly translate linguistic instructions into robotic control policies through end-to-end architectures. Early explorations, such as RT-1[5] and RT-2[4] , demonstrated the potential of multimodal pre-training on large-scale embodied data, while subsequent works like PaLM-E[7] further enhanced reasoning capabilities. With the release of datasets like Open X-Embodiment[26], the field has entered a period of rapid expansion. Foundational models such as OpenVLA[27, 28] and π_0 [29] exhibit remarkable cross-scene generalization by co-training on web-scale Internet data and robot demonstrations. Recent advances, including $\pi_{0.6}$[30], have also

introduced reinforcement learning to bolster performance in real-world environments. Concurrent with the pursuit of model scaling, research into deployment efficiency has gained significant traction. Lightweight models like TinyVLA[6] , SmolVLA [31] , and Evo-1 [32] achieve superior inference speeds by optimizing backbones and action heads. However, regardless of scale, most current VLAs rely on Vision-Language Model (VLM) backbones pre-trained on static 2D web data. This inherent lack of explicit depth scales and physical layouts creates a structural bottleneck for tasks requiring spatial understanding.

2.2 Spatial-Aware VLA

To overcome the limitations of 2D observations, various pathways for enhancing spatial awareness have been proposed. Early explicit methods (3D-VLA[13] , PointVLA[12] , and SpatialVLA [14]) integrate point clouds or real-time depth data to provide geometric constraints. While effective in specific scenarios, these methods depend heavily on 3D sensors and face challenges such as data scarcity and hardware noise. Simultaneously, advancements in 3D reconstruction—transitioning from monocular depth estimation [33] to high-fidelity spatial recovery [34, 23]—have made it feasible to extract robust visual-spatial priors from RGB inputs. This progression has catalyzed a new paradigm of implicit spatial understanding in VLA models, where geometric knowledge from spatial experts is internalized into the policy backbone via techniques like feature fusion or auxiliary task [16, 17], effectively bypassing the complexities of explicit 3D pipelines. However, existing implicit strategies often settle for shallow feature-level fitting, only focusing on action-related spatial information and failing to penetrate the decision-making core of VLA backbones. Furthermore, these approaches typically rely on heavy spatial experts[23], leading to substantial computational overhead and limited deployment efficiency. Therefore, achieving a framework that facilitates deep structural coupling and integrated cognition remains an urgent challenge. Such a framework should move beyond shallow, action-only spatial mapping to penetrate the decision-making core of the VLA backbone, fostering a holistic understanding of global spatial layouts rather than merely fitting task-specific geometric cues. Crucially, this framework should operate in a lightweight manner to ensure real-time viability in robotic deployment.

3 Method

We formulate the embodied manipulation task as a conditional sequence generation problem. At each timestep t, the model receives an RGB observation O_t, a natural language task instruction L, and the robot's proprioceptive state S_t as inputs. The objective of GIVLA is to learn a mapping function f_θ that predicts continuous actions $a_t \in \mathcal{A}$ within the robot's action space.

Distinct from other implicit methods that rely on shallow feature fitting, we explicitly incorporate a perception task into the modeling process to facilitate **deep geometry internalization**. Specifically, GIVLA is designed not only to generate action sequences but also to perform a latent-level reconstruction task $g_\phi(\cdot)$ that recovers the geometric features D_t of the scene:

$$a_t, \hat{D}t = f_\theta(O_t, L, S_t) \tag{1}$$

where $\hat{D}t$ represents the internalized spatial knowledge. Furthermore, we reuse the decoder of a pre-trained spatial geometry expert as a Perception Probe to verify the model's capacity to internalize and reconstruct these spatial structures.

3.1 Geometry-Instructed Model Architecture

To endow the pre-trained VLA with holistic spatial cognition without destroying its inherent semantic reasoning capabilities, we design a Geometry-Instructed architecture. This architecture seamlessly couples a spatial geometric expert (Depth Anything v3[22]) with the VLM backbone (InternVL3[9]) through geometry instruction mechanisms: progressive geometry feature injection at the input and geometry feature supervision at the output. Additionally, a Flow Matching action head is employed to handle the multimodality of action distributions. The sturcture is presented in Figure 2.

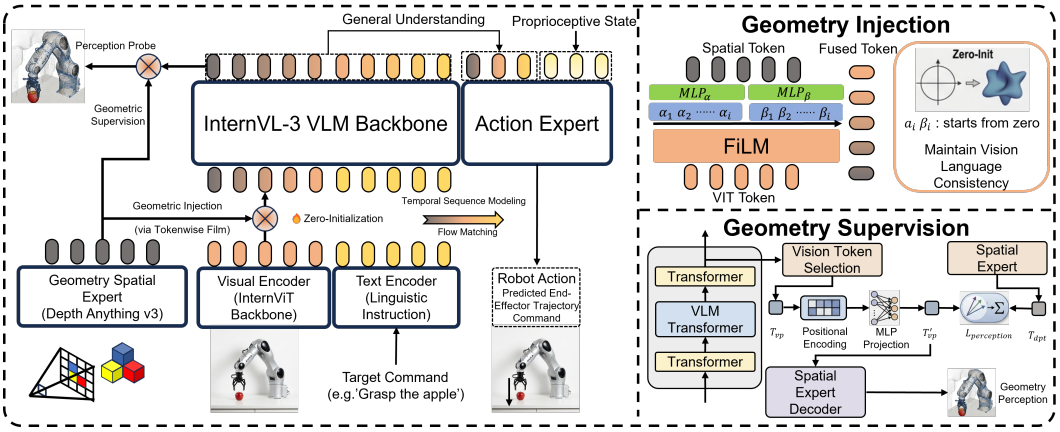


Figure 2: **GIVLA Architecture Overview.** GIVLA achieves deep structural coupling by progressively injecting latent-level geometric priors from a spatial expert (Depth Anything v3 [22]) into the VLM backbone (InternVL3 [9]). The design introduces geometric self-supervision as an optimization "scaffold" and reuses the expert's decoder as a "Perception Probe" to verify the internalization of spatial structures. This coupling enables GIVLA (0.9B) to achieve precise manipulation that transcends shallow feature fitting.

3.1.1 Progressive Geometry Feature Injection

The vision module of InternVL3[9] inherits powerful semantic discrimination from its pre-trained ViT backbone but inherently lacks spatial cognition required for physical manipulation. Conversely, the DA3 expert excels at extracting high-frequency geometric gradients from single RGB images. To achieve a deep integration of spatial and semantic information, we perform geometric feature injection using token-wise FiLM[35] modulation.This mechanism progressively injects the spatial features T_{dpt} extracted by DA3 into the original visual features T_{vit} of the VLA. The fusion process is formulated as:

$$T_{\text{fused}} = (1 + \text{MLP}_{\alpha}(T_{\text{dpt}})) \cdot T_{\text{vit}} + \text{MLP}_{\beta}(T_{\text{dpt}}) \tag{2}$$

Crucially, we employ zero-initialization for the MLP layers. This cold-start injection ensures that the VLM preserves its original pre-trained semantic-action alignment during the early training stages, allowing spatial features to be gradually assimilated without introducing disruptive geometric noise.

3.1.2 Optimization Objectives: Geometry Supervision and Action Generation

To ensure that the VLA internalizes 3D spatial geometry rather than fitting shallow features, we introduce a self-supervised geometric instruction mechanism. We extract vision tokens from the VLM backbone and supervise the model to recover deep spatial representations from these tokens, thereby empowering the Decision Core with holistic spatial awareness. To validate the efficacy of this internalization, we reuse the pre-trained spatial expert's decoder as a 'Perception Probe'. This probe serves to verify that the model can achieve high-quality depth reconstruction from its latent features, confirming the successful integration of geometric knowledge. We define this spatial perception loss ($\mathcal{L}_{\text{perception}}$) as the cosine similarity between the VLM's spatial feature T_{vp} and the expert's feature T_{dpt}:

$$\mathcal{L}_{\text{perception}} = \mathcal{F}_{\cos}(T_{\text{vp}}, T_{\text{dpt}}) \tag{3}$$

Complementary to the perception objective, robotic manipulation is handled by a Flow Matching action head. This module leverages the Task-Space Representations derived from the Geometrically-Grounded Decision Core (comprising the VLM backbone and the integrated spatial expert) to model continuous, multimodal action distributions, ensuring temporal coherence in robot trajectories.The corresponding action loss is defined as:

$$\mathcal{L}_{\text{action}} = \text{MSE}(a_{\text{pred}}, a_{\text{gt}}) \tag{4}$$

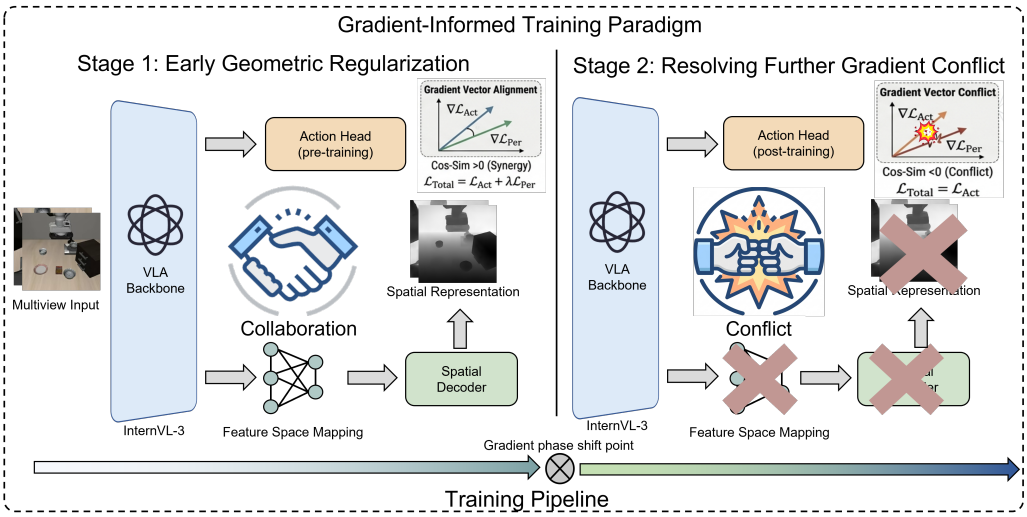


Figure 3: We adopt a **two-stage** progressive training strategy to gradually integrate spatial geometric priors into the VLA model, while fully preserving the strong semantic and action reasoning capabilities inherited from the pre-trained VLM. (We will discuss and analyze the impact of gradient competition on model performance in Figure 5, as well as the rationale for adopting a two-stage approach, in the subsequent experiments.)

3.2 Gradient-Informed Training Paradigm

The integration of holistic spatial perception and precise action decision-making introduces complex training dynamics. We propose a gradient-Informed two-stage progressive training paradigm based on the dual nature of perception and action observed in gradient analysis [24]. We present the training paradigm in Figure 3.

3.2.1 Stage 1: Geometric Regularization for Action Generalization

In the initial stage, we freeze the pre-trained VLM and spatial expert, focusing solely on training the geometric injection layer and the action head. Our gradient analysis reveals that during this phase, the gradients derived from holistic spatial perception and fine-grained action decision-making are positively correlated. The geometric alignment acts as a beneficial explicit geometric regularization. By optimizing the joint objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \alpha \mathcal{L}_{\text{perception}} \tag{5}$$

the combined gradient effectively guides the model parameters toward an optimal basin corresponding to deep geometry internalization, preventing the action head from overfitting to shallow features.

3.2.2 Stage 2: Action-Centric Refinement for Precise Manipulation

As the model establishes a foundational understanding of the 3D environment, the training dynamics shift significantly. In the middle and late stages, a strong gradient conflict emerges between the holistic spatial perception task and the local, robust action manipulation task. Continuing to enforce global spatial alignment forces the network to compromise, severely degrading the precision of robotic manipulation. To resolve this task-level conflict, Stage 2 transitions to a full-parameter end-to-end fine-tuning phase where we abandon the global spatial alignment loss and focus exclusively on action execution:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} \tag{6}$$

While $\mathcal{L}_{\text{perception}}$ is removed, the geometric features continue to be injected. By jointly unfreezing the parameters of spatial expert and the VLM backbone, the model shifts its cognitive focus to the specific geometric details required for precise end-effector trajectories. This gradient-informed transition ensures the preservation of holistic spatial awareness while maximizing manipulation accuracy.

Table 1: LIBERO Evaluation

Model	Params	Robo-Pretrain.	Spatial	Object	Goal	Long	Avg. (%)
Baseline 2D VLA							
OpenVLA[27]	7B	Yes	84.70	88.40	79.20	53.70	76.50
CoT-VLA[36]	7B	Yes	87.50	91.60	87.60	69.00	81.10
π_0-FAST[29]	3.5B	Yes	96.40	96.80	88.60	60.20	85.50
π_0[29]	3.5B	Yes	96.80	98.80	95.80	85.20	94.20
SmolVLA[31]	0.45B	No	93.00	94.00	91.00	77.00	88.80
GR00T N1[37]	2B	Yes	94.40	97.60	93.00	90.60	93.90
Evo-1[32]	0.77B	No	92.70	97.70	**96.30**	92.30	94.80
Explicit Spatial-Aware Methods							
SpatialVLA[14]	4B	Yes	88.20	89.90	78.60	55.50	78.10
Implicit Spatial-Aware Methods							
DepthVLA[17]	3.5B	Yes	96.40	**98.00**	95.80	89.20	94.90
GIVLA (Ours)	**0.9B**	**No**	**96.60**	**98.00**	**96.30**	**92.70**	**95.90**

Table 2: LIBERO-plus Evaluation(Robustness Evaluation by Zeroshot)

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Total↑
OpenVLA[27]	0.80	3.50	23.00	8.10	34.80	15.20	28.50	15.60
WorldVLA[38]	0.10	27.90	41.60	43.70	17.10	10.90	38.00	25.00
NORA[39]	2.20	37.00	65.10	45.70	58.60	12.80	62.10	39.00
UniVLA[40]	1.80	46.20	69.60	69.00	81.00	21.20	31.90	43.90
π_0[29]	13.80	6.00	58.80	85.00	**81.40**	79.00	**68.90**	53.60
GIVLA (Ours)	**46.42**	**47.97**	**70.98**	**87.01**	75.41	**86.89**	68.00	**68.95**

4 Experiments

4.1 Simulation Experiments

4.1.1 LIBERO

Setup. To evaluate the effectiveness of our GIVLA, we conduct experiments on the challenging LIBERO benchmark. We perform 10 trials per task and report the average performance over 10 independent training runs. We compare GIVLA with a wide range of representative VLA baselines, including:

- **2D VLA Baselines**: OpenVLA, CoT-VLA, π_0-FAST, SmolVLA, GR00T N1, π_0, Evo-1.
- **Explicit 3D-Aware VLAs**: SpatialVLA .
- **Implicit 3D-Aware VLAs**: DepthVLA, **GIVLA (Ours)**.

All baseline results are taken from their original papers or official implementations to ensure fair and reliable comparisons.

Results. As shown in Table 1, GIVLA achieves an average success rate of 95.90%, outperforming strong baselines such as π_0 (94.20%) and SmolVLA (88.80%). GIVLA consistently exhibits robust and balanced performance across all four task categories. Compared with existing explicit and implicit spatial-aware methods, GIVLA achieves highly competitive results with a much smaller parameter size, demonstrating the efficiency and effectiveness of our proposed progressive self-supervised framework.

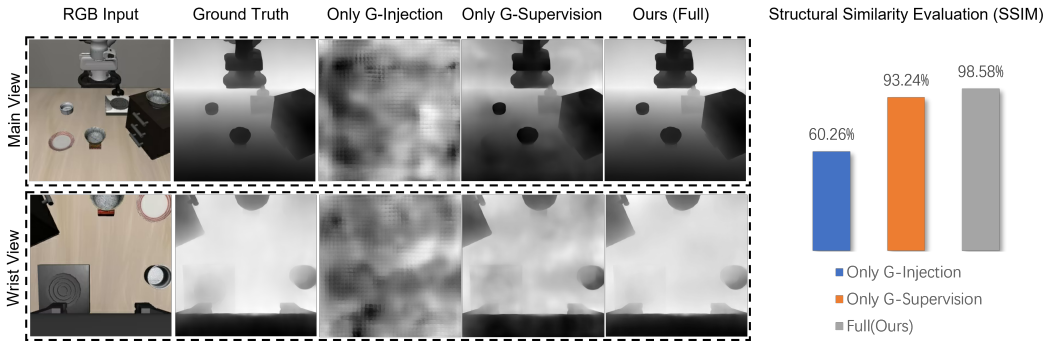


Figure 4: **Spatial Perception Visualization and Structural Similarity Evaluation (SSIM).** (Left) Qualitative depth reconstruction results using our **Perception Probe** across different ablation settings in the LIBERO simulation. (Right) Quantitative evaluation of reconstruction fidelity using the SSIM metric.

4.1.2 LIBERO-plus

Setup. To thoroughly evaluate the robustness of GIVLA under distribution shifts, we train our model on the LIBERO dataset and perform zero-shot evaluation on the challenging LIBERO-plus benchmark. We conduct one trial per task and report the average success rate over ten independent runs to ensure reliability. Under this setup, we compare GIVLA with representative VLA baselines, including OpenVLA,NORA, WorldVLA, UniVLA, π_0, and GIVLA (Ours).

Results. As shown in Table 2, GIVLA achieves an average success rate of **68.95%**, outperforming most existing strong baselines. Across all seven distribution shift categories, GIVLA exhibits robust and stable performance thanks to its comprehensive spatial geometric understanding. Notably, under the challenging shifts of *Camera* perspective and *Robot* initial pose, GIVLA maintains top-tier performance. Meanwhile, its spatial awareness also enhances robustness against other types of disturbances, leading to a strong overall average success rate of 68.95%. Overall, GIVLA demonstrates superior performance on LIBERO-plus, validating the effectiveness and generalization capability of our proposed framework in complex real-world distribution shifts.

4.1.3 Spatial Perception Analysis in Simulation

To assess deep geometry internalization, we utilize the pre-trained spatial expert's decoder as a perception probe to reconstruct depth maps from the latent features. As illustrated in Figure 4, GIVLA (Full) generates high-fidelity spatial structures with a Structural Similarity (SSIM) of **98.58%**, even with a frozen VLM backbone.

Quantitative ablation shows that both components are indispensable: while *G-Supervision* alone yields reasonable layouts (93.24% SSIM), the absence of *G-Supervision* leads to significant structural corruption (60.26% SSIM). These results confirm that our paradigm effectively internalizes geometric knowledge into the latent manifold, enabling the VLA to capture finer structural details. More visualization cases are provided in Appendix C.1 and Figure 8.

4.1.4 Ablation Experiment

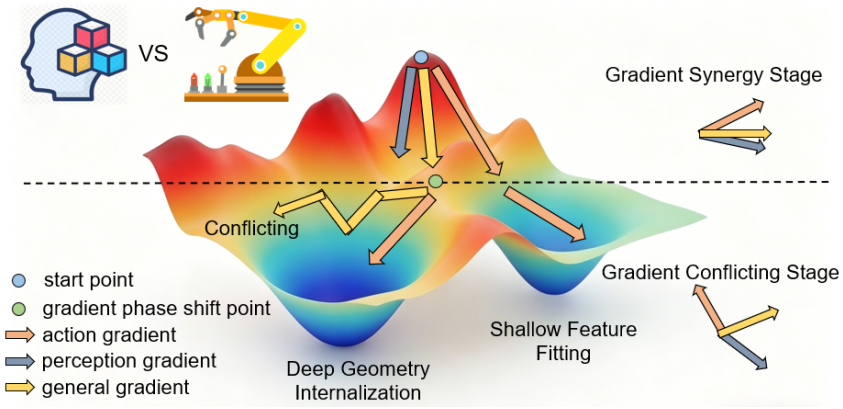
We ablate geometry injection and supervision on LIBERO-Spatial and LIBERO-plus Spatial. As shown in Table 3, FiLM injection alone improves success rates over the geometry-free Base VLA on both benchmarks. Adding Stage-1 geometry supervision further improves LIBERO-plus Spatial performance from 72.72% to 78.28%, while maintaining comparable LIBERO-Spatial performance (96.90% vs. 96.60%). In contrast, retaining geometry supervision during Stage 2 causes severe performance degradation, supporting the use of geometry supervision as an early training scaffold.

4.2 Gradient Analysis

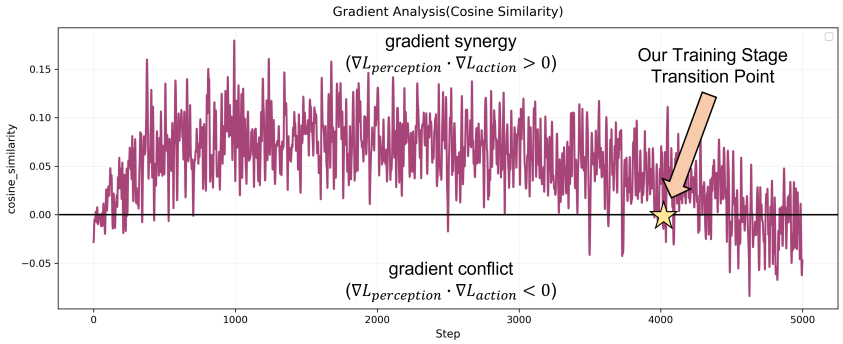
We measure the interaction between geometry supervision and action learning at the FiLM fusion layer. Using the same minibatch and forward computation, we separately compute $g_{\text{perception}} =$

Table 3: Controlled ablations on LIBERO-Spatial and LIBERO-plus Spatial. Geometry injection and geometry supervision are indicated separately. All variants use the same VLA backbone and action head. Success rates are reported in percent.

Model	Geometry injection	Geometry supervision		LIBERO Spatial	LIBERO-plus Spatial
		Stage 1	Stage 2		
Base VLA	✗	✗	✗	87.90	69.30
FiLM injection only	✓	✗	✗	**96.90**	72.72
Always-on supervision	✓	✓	✓	2.30	0.50
GIVLA (Ours)	✓	✓	✗	96.60	**78.28**



(a) Conceptual illustration of the optimization trajectory.



(b) Cosine similarity between geometry and action gradients.

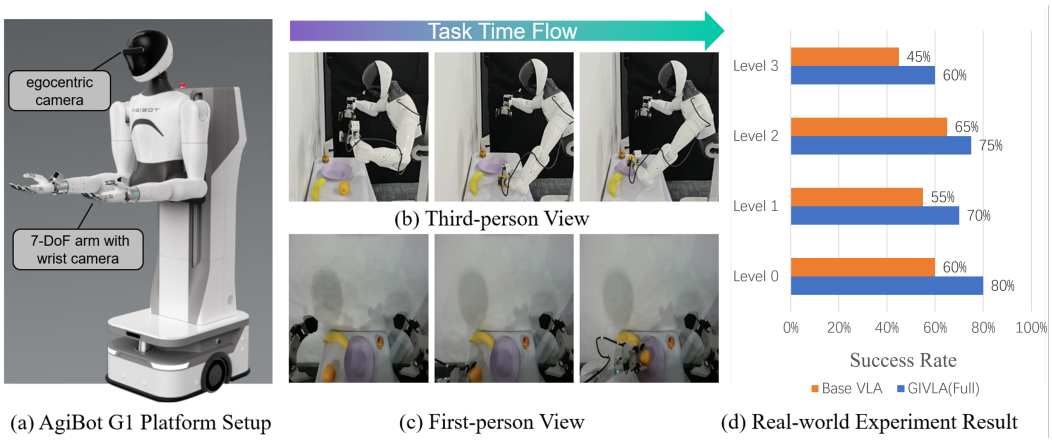
Figure 5: Gradient analysis of GIVLA. (a) Conceptual illustration of geometry-guided early training followed by action-centric refinement. (b) Gradient cosine similarity at the FiLM fusion layer during a 5,000-step diagnostic run with the Stage-1 configuration. The marked 4,000-step position denotes the fixed transition used in the reported training schedule.

$\nabla_{\theta_{\text{FiLM}}} \mathcal{L}_{\text{perception}}$ and $g_{\text{action}} = \nabla_{\theta_{\text{FiLM}}} \mathcal{L}_{\text{action}}$, and evaluate their cosine similarity:

$$c = \frac{g_{\text{perception}}^{\top} g_{\text{action}}}{\|g_{\text{perception}}\|_2 \|g_{\text{action}}\|_2 + \epsilon}. \tag{7}$$

Positive values indicate aligned gradient directions, whereas negative values indicate conflicting directions.

Figure 5b shows predominantly positive alignment during early training, followed by weaker alignment and increasingly negative interactions toward the end of the run. This observation motivates using geometry supervision as an early optimization scaffold before action-only refinement, as illustrated conceptually in Figure 5a. The controlled ablations in Table 3 support this design: temporary



(a) AgiBot G1 Platform Setup (c) First-person View (d) Real-world Experiment Result

Figure 6: **Real-world Experiments on AgiBot G1.** (a) **AgiBot G1 Platform Setup**: The AgiBot G1 humanoid robot with an egocentric head camera and dual 7-DoF arms. (b) **Third-person View**: Synchronized visual sequence from an external observer's perspective. (c) **First-person View**: Synchronized visual sequence from the robot's onboard camera. (d) **Real-world Experiment Result**: Quantitative performance across four difficulty levels (L0–L3).

geometry supervision improves LIBERO-plus Spatial performance, whereas retaining it throughout refinement severely degrades performance. Additional gradient analyses across learning rates and loss weights are provided in Appendix B.4.

4.3 Real-world Experiments

4.3.1 Real-world Manipulation

To evaluate the practical effectiveness of GIVLA, we conduct a set of challenging "pick-and-place" tasks using the AgiBot G1 platform, which is equipped with dual 7-DoF arms and 1-DoF grippers.

Setup. We define four difficulty levels based on environmental clutter and object stability for the task of placing a lemon into a bowl: (1) **L0**: Stable object state without clutter; (2) **L1**: Unstable object state without clutter; (3) **L2**: Stable object state with clutter interference; (4) **L3**: Unstable object state with clutter interference. Both models are fine-tuned using the same demonstration data, with 25 trajectories per difficulty level. We compare GIVLA with a Base VLA that uses the same InternVL3 backbone and Flow Matching action head but excludes DA3 geometric features, FiLM-based geometry injection, and geometry supervision. Each model is evaluated over 80 trials, with 20 trials per difficulty level. This real-world evaluation primarily assesses deployment feasibility under scarce demonstration data, rather than maximizing task performance.

Results Analysis. As shown in Figure 6, GIVLA achieves promising success rates despite the extreme scarcity of demonstration data. The model demonstrates significant potential for precision manipulation, effectively handling unstable object poses and showing reactive grasping adaptability when objects are moved. These results support the deployment feasibility of GIVLA in this limited-data real-world setting.

4.3.2 Spatial Perception Analysis in Real-world

We further verify whether the spatial cognition capability is successfully activated in the real-world domain. We collect RGB-D data from real task scenes and compare the depth maps generated by our model against the Ground Truth (GT) and the original DA3 expert predictions. We employ the Structural Similarity Index (SSIM) as a quantitative metric to evaluate the spatial alignment. The

Table 4: Quantitative comparison of spatial understanding (SSIM).

Method	DA3 (Expert)	GIVLA (Ours)
SSIM (%)	51.64	50.16

quantitative results in Table 4 indicate that GIVLA effectively internalizes the expert's geometric knowledge within the VLM's latent space using only RGB input. While a numerical gap persists between the models and the raw depth GT, GIVLA achieves performance closely approaching the native spatial expert. This suggests that the VLA's spatial awareness is primarily bounded by the expert's capability. Detailed qualitative depth comparisons and further discussions regarding sensor noise are provided in Appendix C.2 , Figure 9 and Figure 10.

5 Conclusion

In this paper, we introduced **GIVLA**, a geometry-instructed and gradient-informed framework that achieves robust spatial awareness in VLAs without relying on massive 3D datasets. Our work reveals two fundamental characteristics of embodied vision-language models: the **activation of latent spatial potential** within VLM backbones and the intrinsic **perception-action duality** that governs training dynamics. By managing the transition from gradient synergy to conflict, GIVLA effectively bridges the gap between semantic reasoning and precise 3D manipulation. Despite its effectiveness and high parameter efficiency, GIVLA is currently subject to the performance upper bound of the upstream geometric expert. Furthermore, as an end-to-end paradigm, it lacks explicit safety constraints in action generation, which remains a critical requirement for human-robot interaction in unstructured environments. Future exploration into automated dynamic optimization and the integration of safety-aware control will further enhance the reliability of spatial-aware VLAs.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key R&D Program of China (2024YFB4708900), and in part by the National Natural Science Foundation of China (62225309, U24A20278, 62361166632, 62503317), the Explorers Program of Shanghai (25TS1415100), the Shanghai Municipal Special Program for Basic Research on General AI Foundation Models (2025SHZDZX025G01), the Natural Science Foundation of Shanghai (25ZR1402232), the China Postdoctoral Science Foundation (2025M771709), and the Postdoctoral Fellowship Program of CPSF (GZC20251180).

References

[1] Qiming Liu, Xinru Cui, Zhe Liu, and Hesheng Wang. Cognitive navigation for intelligent mobile robots: A learning-based approach with topological memory configuration. *IEEE/CAA Journal of Automatica Sinica*, 11(9):1933–1943, 2024.

[2] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024.

[3] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry

Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[6] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation, 2025.

[7] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023.

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2024.

[9] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

[10] Tianchen Deng, Yue Pan, Shenghai Yuan, Dong Li, Chen Wang, Mingrui Li, Long Chen, Lihua Xie, Danwei Wang, Jingchuan Wang, Javier Civera, Hesheng Wang, and Weidong Chen. What is the best 3d scene representation for robotics? from geometric to foundation models. *arXiv preprint arXiv:2512.03422*, 2025.

[11] Qiming Liu, Haoran Xin, Zhe Liu, and Hesheng Wang. Integrating neural radiance fields end-to-end for cognitive visuomotor navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11200–11215, 2024.

[12] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models, 2025.

[13] 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, 2024.

[14] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model, 2025.

[15] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024.

[16] 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, 2025.

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

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

[19] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction, 2025.

[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, 2025.

[21] 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, 2025.

[22] 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, 2025.

[23] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer, 2025.

[24] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.

[25] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization, 2019.

[26] Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov,

Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025.

[27] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024.

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

[29] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026.

[30] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a vla that learns from experience, 2025.

[31] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025.

[32] Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment, 2025.

[33] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2020.

[34] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data, 2024.

[35] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[36] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models, 2025.

[37] NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025.

[38] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model, 2025.

[39] Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks, 2025.

[40] 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, 2025.

[41] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models, 2025.

[42] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[44] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[45] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024.

[46] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[47] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2025.

[48] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[49] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models, 2024.

[50] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, Song Han, and Yao Lu. Vila-u: a unified foundation model integrating visual understanding and generation, 2025.

[51] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines, 2020.

[52] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025.

[53] Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, Nadine Chang, Karan Sapra, Amala Sanjay Deshmukh, Tuomas Rintamaki, Matthieu Le, Ilia Karmanov, Lukas Voegtle, Philipp Fischer, De-An Huang, Timo Roman, Tong Lu, Jose M. Alvarez, Bryan Catanzaro, Jan Kautz, Andrew Tao, Guilin Liu, and Zhiding Yu. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models, 2025.

[54] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. Paligemma 2: A family of versatile vlms for transfer, 2024.

[55] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2, 2024.

Table 5: Two-stage training configuration of GIVLA.

Configuration	Stage 1	Stage 2
Training steps	4,000	150,000
Learning rate	2.5×10^{-5}	1×10^{-5}
Geometry-loss weight α	0.5	0
VLM backbone	Frozen	Trainable
Spatial expert	Frozen	Trainable
Geometry injection	Enabled	Enabled
Action head	Trainable	Trainable

A Code Availability

The core implementation of the GIVLA model, including training scripts, model architectures, and documentation for reproducibility, is available at the supplementary material. Also you can find our project at `https://github.com/Alexander-bugwriter/GIVLA`.

B Training Details

B.1 Optimization Configuration

We use AdamW with a weight decay of 10^{-3} and an action-head dropout rate of 0.2. Training uses BF16 precision with Accelerate and DeepSpeed ZeRO-2, a batch size of 8, and 448×448 input images. Each stage uses a 1,000-step warmup followed by cosine learning-rate decay, with gradient clipping at 1.0. Stage-specific settings are summarized in Table 5.

B.2 Two-Stage Training Schedule

Stage 1 jointly optimizes action prediction and geometry alignment while keeping the pretrained VLM and spatial expert frozen. After 4,000 steps, Stage 2 loads the resulting model weights, removes geometry supervision, and jointly unfreezes the VLM and spatial expert. The optimizer and scheduler are reinitialized for 150,000 steps of action-only refinement. Geometry injection remains active throughout both stages.

B.3 Hardware and Computational Efficiency

LIBERO training is performed on a single NVIDIA H20 GPU. For physical robot experiments, training uses two NVIDIA GeForce RTX 4090 GPUs (48 GB variants), and inference is performed on a single NVIDIA GeForce RTX 4070 GPU. The measured peak inference memory usage is 2.74 GB, with a latency of 353.3 ms per inference.

B.4 Gradient Sensitivity to Hyperparameters

We examine nine combinations of learning rate and geometry-loss weight using the gradient measurement protocol described in Section 4.2. Across all nine configurations, the mean gradient cosine similarity decreases from the first 20% to the last 20% of training. Larger learning rates produce more pronounced negative alignment, while the influence of α varies across configurations. These results show how the balance between geometric alignment and action learning changes with the optimization settings.

Persistent geometry supervision. We additionally analyze the configuration that retains geometry supervision with $\alpha = 0.5$ during Stage 2. Combining the recorded trajectories from the two stages yields a mean gradient cosine similarity of -0.1486, a standard deviation of 0.1284 across logged observations, and a negative-cosine fraction of 91.45%. Over the final 20% of the trajectory, the mean decreases to -0.2624, with 96.59% of observations being negative. Together with the corresponding ablation results, this persistent negative alignment supports removing geometry supervision before full-model action refinement.

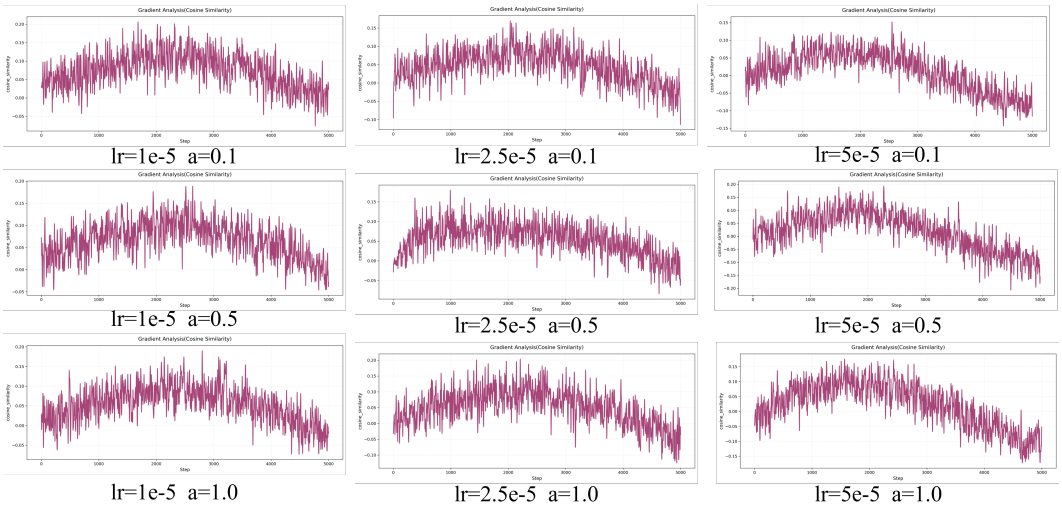


Figure 7: Geometry–action gradient cosine similarity under nine hyperparameter configurations. Columns correspond to learning rates of 1×10^{-5}, 2.5×10^{-5}, and 5×10^{-5}; rows correspond to geometry-loss weights $\alpha = 0.1$, 0.5, and 1.0. Each curve covers a 5,000-step diagnostic run with the Stage-1 configuration. Gradients are measured at the FiLM fusion layer.

C Spatial Cognition Visualizations

C.1 Visualization of Ablation in Simulation

To intuitively demonstrate how GIVLA internalizes geometric knowledge from the spatial expert, we visualize the predicted spatial features of different ablation variants. We utilize the raw depth maps generated by the pre-trained spatial expert (Depth-Anything-v3) as the reference (pseudo-ground truth) and project the internal representations of GIVLA onto the spatial manifold.

As illustrated in Figure 8, the visualization results reveal distinct behaviors across different configurations:
Geometry-Injection Only (without Supervision): When geometric features are injected via FiLM layers without the two-stage alignment, the model primarily captures low-level textures and shallow edges. It fails to form a coherent structural understanding of the scene, indicating a loss of holistic spatial perception.

Geometry-Supervision Only (without injection): While this variant exhibits a degree of global scene layout awareness, the representations are coarse. Specifically, the geometric details in critical manipulation areas—such as object boundaries and gripper-object interfaces—are blurred or misaligned.

GIVLA (Full: Injection + Supervision): By simultaneously employing geometric feature injection and gradient-informed alignment, GIVLA achieves a superior degree of spatial internalization. The visualized manifold not only captures the global depth hierarchy but also maintains high-fidelity details in interaction-related regions.

These visualizations empirically confirm that our progressive training paradigm effectively activates the latent spatial manifold of VLMs, enabling the model to bridge the gap between abstract visual features and precise physical geometry.

C.2 Visualization of Real-world Experiment

The quantitative results in Table 4 lead to several key insights: (1) **Expert-Bounded Perception:** Both DA3 and GIVLA exhibit a numerical gap compared to the raw depth GT. This indicates that the spatial awareness of the VLA is primarily constrained by the performance upper bound of the pre-trained geometric expert, rather than the VLA's internal representation capacity.
(2) **High-Fidelity Knowledge Internalization:** GIVLA achieves an SSIM of 50.16%, closely

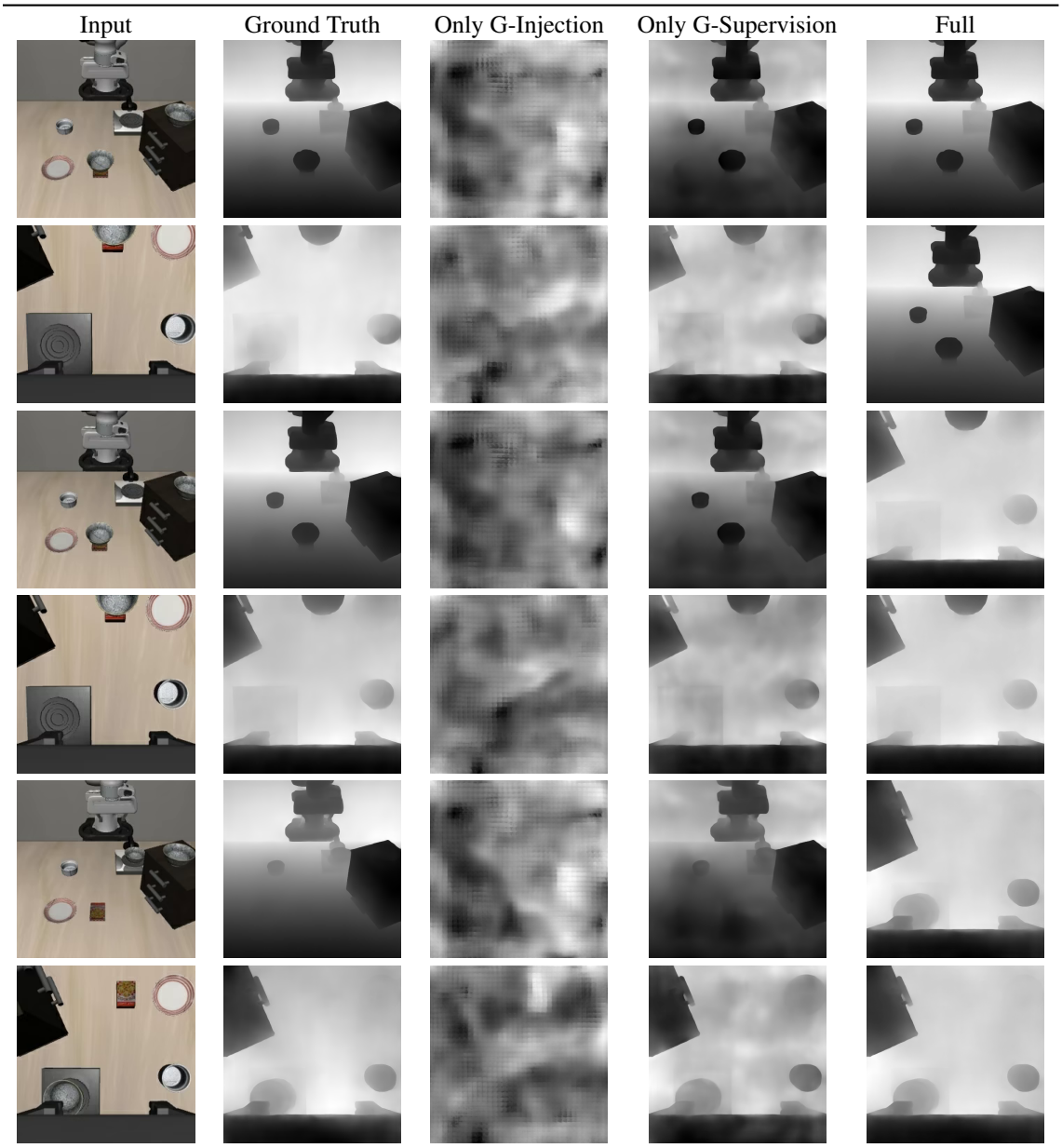


Figure 8: Visualization of general spatial perception

approaching the performance of the native DA3 expert (51.64%) while using significantly fewer specialized parameters and only RGB input. This demonstrates that our first-stage alignment effectively unlocks the VLM's latent spatial potential, allowing it to nearly perfectly internalize the expert's geometric prior.

(3) **Noise Robustness and Semantic Clarity:** Visualization (see Appendix) reveals that the real-world GT is heavily corrupted by sensor noise and environmental reflections. In contrast, GIVLA generates smoother depth maps with clearer object boundaries. The slight numerical difference between GIVLA and DA3 does not hinder the model's ability to provide high-fidelity spatial guidance for complex manipulation tasks.

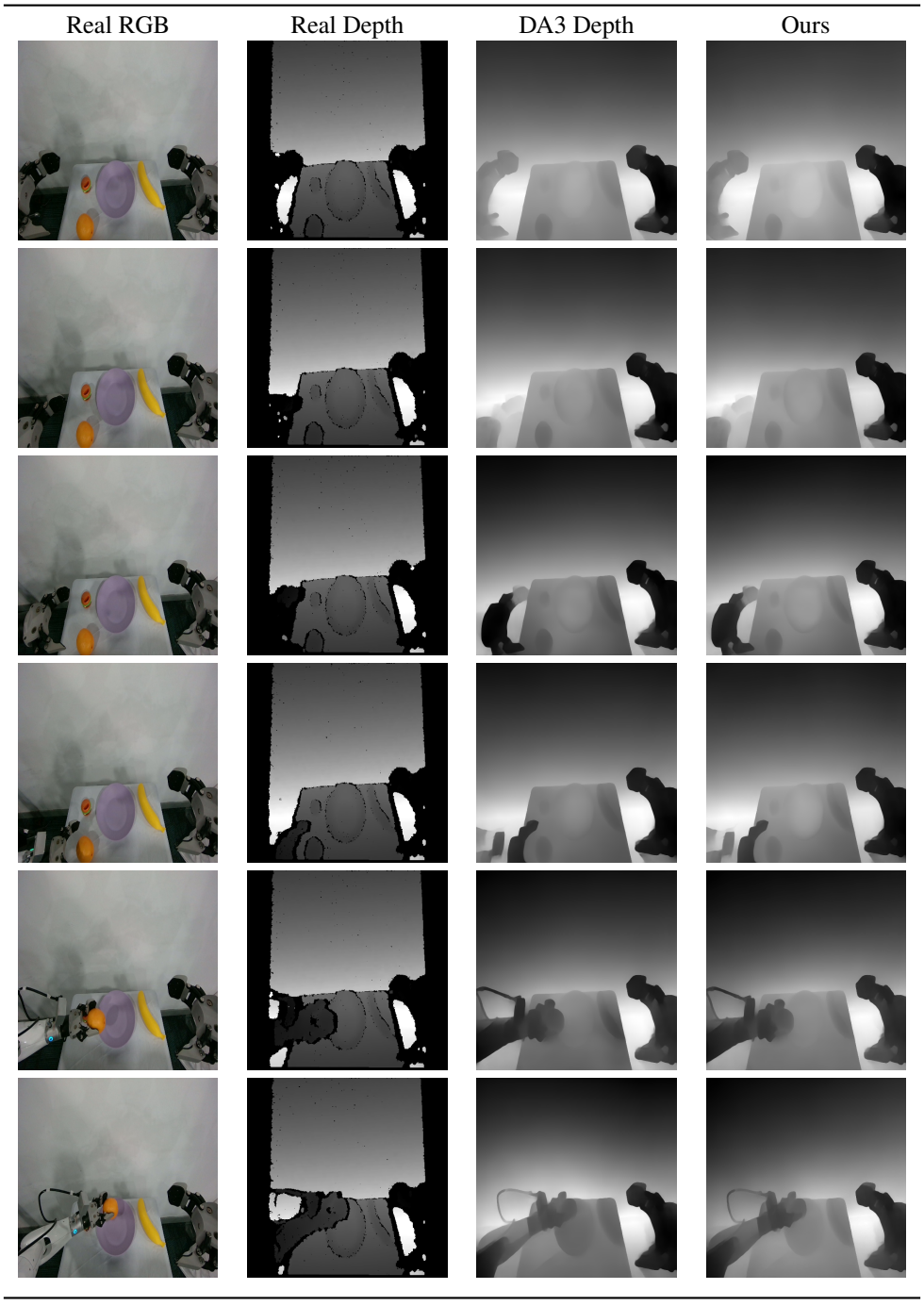


Figure 9: Visualization of real-world spatial perception

D Details on Experiments

D.1 LIBERO

Task Setup. LIBERO is a benchmark designed to evaluate lifelong learning in robot manipulation, focusing on the transfer of both declarative and procedural knowledge. The tasks in LIBERO are categorized into four primary task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. Each suite aims to test different aspects of robot learning, including how well

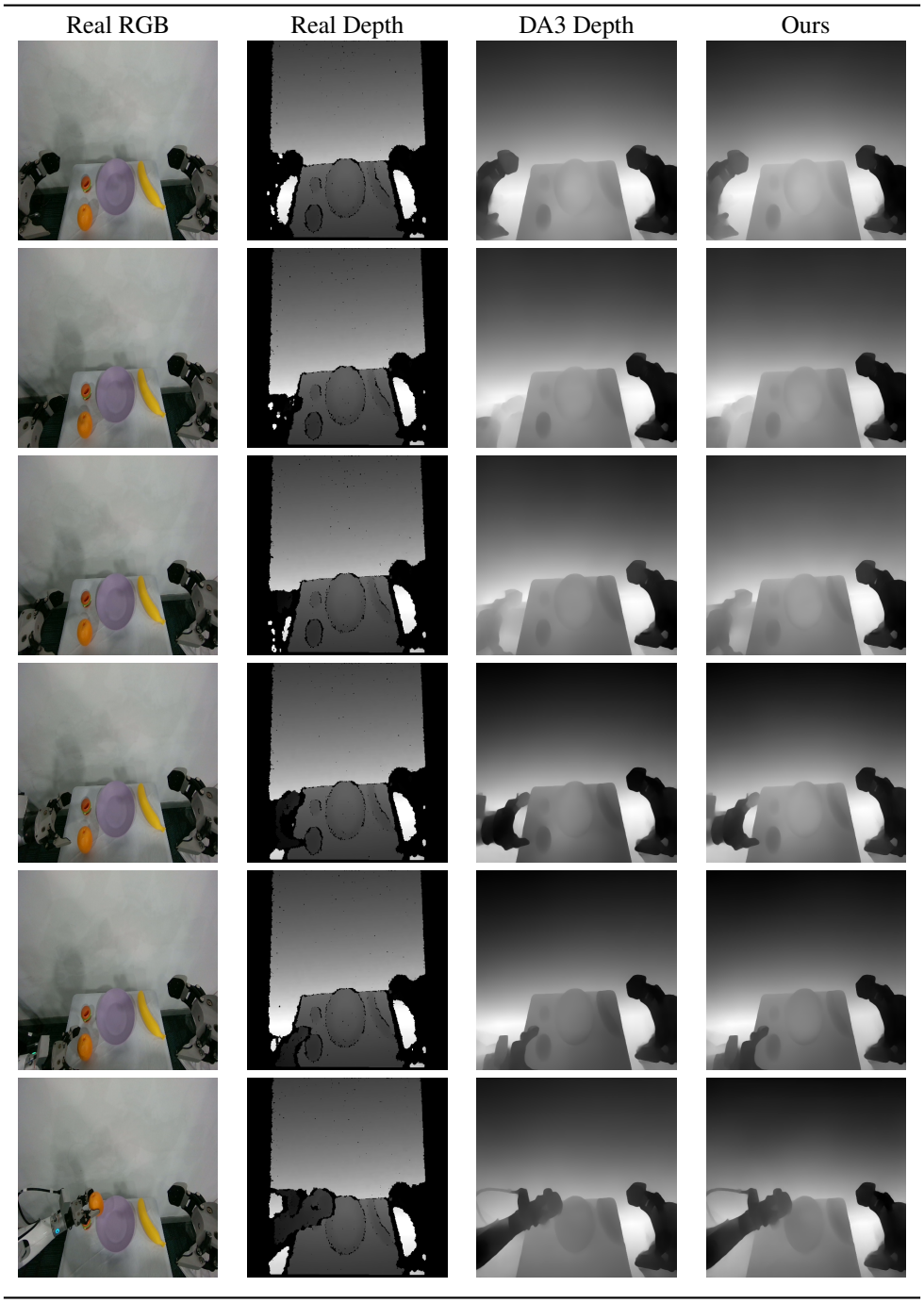


Figure 10: Visualization of real-world spatial perception

the robot can generalize learned knowledge to new situations, transfer spatial and object-specific information, and adapt to diverse task goals.

1. **LIBERO-Spatial.** This suite tests the robot's ability to understand and manipulate spatial relationships between objects. In tasks from this suite, the robot is tasked with manipulating objects (such as placing a bowl on a plate), where the challenge lies in learning how objects relate spatially in the environment.

2. **LIBERO-Object.** Tasks in this suite test the robot's ability to recognize and manipulate different objects. The primary challenge is in learning to handle various objects, each with different shapes, sizes, and properties.
3. **LIBERO-Goal.** These tasks focus on goal-directed manipulation where the robot must learn to achieve specific task goals, such as placing objects in predefined locations. The goal is usually fixed, but the actions to reach the goal may vary.
4. **LIBERO-Long.** This suite combines elements from the previous three, introducing more complex tasks that require a combination of declarative and procedural knowledge. The LIBERO-100 suite is more diverse, with a larger set of tasks, enabling a thorough evaluation of the robot's ability to perform a wide range of tasks and generalize across various environments. The suite includes both short-horizon tasks and long-horizon tasks, with 90 tasks requiring quick decision-making and 10 tasks designed to test long-term planning and execution.

The goal of these tasks is not only to test the robot's ability to perform specific tasks but also to understand how well the robot can transfer learned skills across different task settings, handling new object types, new goals, and different spatial arrangements.

D.2 LIBERO-plus

Task Setup. LIBERO-Plus [41] is an enhanced diagnostic benchmark built upon the original LIBERO to systematically evaluate the robustness of Vision-Language-Action (VLA) models. While standard benchmarks often report aggregate success rates under ideal conditions, LIBERO-Plus exposes model vulnerabilities by introducing controlled perturbations across multiple environmental and task dimensions. The benchmark comprises 10,030 tasks spanning seven perturbation factors with twenty-one low-level components.

D.2.1 Perturbation Dimensions

LIBERO-Plus evaluates VLA performance under seven common single-dimension perturbations, designed to test the limits of multimodal understanding:

- **Objects Layout**: This dimension tests robustness against object-level disturbances, including adding confounding distractor objects and shifting the target object's initial position or orientation.
- **Camera Viewpoints**: Evaluates view-free representation by systematically varying the third-person camera's distance, spherical position (azimuth and elevation), and orientation.
- **Robot Initial States**: Tests kinematic reasoning by applying random perturbations to the robot arm's initial joint angles.
- **Language Instructions**: Employs Large Language Models (LLMs) to rewrite instructions into longer, conversational forms or commonsense-based descriptions to test semantic understanding and filtering.
- **Light Conditions**: Varies illumination intensity, direction, specular highlights, and the presence of shadows to evaluate visual understanding under diverse lighting.
- **Background Textures**: Modifies the scene theme (e.g., wall textures) and working surface appearance (e.g., tabletop materials) using a curated collection of textures.
- **Sensor Noise**: Injects real-world sensor imperfections such as motion blur, Gaussian blur, zoom blur, fog, and glass distortions to evaluate robustness under degraded input quality.

D.2.2 Compositional Generalization Gap

A unique feature of LIBERO-Plus is the analysis of *compositional generalization*, defined as the model's ability to handle joint perturbations from multiple dimensions. Statistical analysis reveals a consistent negative compositionality gap, suggesting that co-occurring shifts act as coupled noise sources that significantly degrade performance beyond the effects of independent perturbations. This highlights the entanglement in learned representations and the lack of mechanisms in current models to capture higher-order dependencies.

Table 6: Task Instructions for the Four Evaluated LIBERO Suites.

Task Category	Task Instruction
LIBERO-Spatial	Pick up the black bowl between the plate and the ramekin and place it on the plate. Pick up the black bowl next to the ramekin and place it on the plate. Pick up the black bowl from table center and place it on the plate. Pick up the black bowl on the cookie box and place it on the plate. Pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate. Pick up the black bowl on the ramekin and place it on the plate. Pick up the black bowl next to the cookie box and place it on the plate. Pick up the black bowl on the stove and place it on the plate. Pick up the black bowl next to the plate and place it on the plate. Pick up the black bowl on the wooden cabinet and place it on the plate.
LIBERO-Object	Pick up the alphabet soup and place it in the basket. Pick up the cream cheese and place it in the basket. Pick up the salad dressing and place it in the basket. Pick up the BBQ sauce and place it in the basket. Pick up the ketchup and place it in the basket. Pick up the tomato sauce and place it in the basket. Pick up the butter and place it in the basket. Pick up the milk and place it in the basket. Pick up the chocolate pudding and place it in the basket. Pick up the orange juice and place it in the basket.
LIBERO-Goal	Open the middle drawer of the cabinet. Put the bowl on the stove. Put the wine bottle on top of the cabinet. Open the top drawer and put the bowl inside. Put the bowl on top of the cabinet. Push the plate to the front of the stove. Put the cream cheese in the bowl. Turn on the stove. Put the bowl on the plate. Put the wine bottle on the rack.
LIBERO-Long	Put both the alphabet soup and the tomato sauce in the basket. Put both the cream cheese box and the butter in the basket. Turn on the stove and put the moka pot on it. Put the black bowl in the bottom drawer of the cabinet and close it. Put the white mug on the left plate and put the yellow and white mug on the right plate. Pick up the book and place it in the back compartment of the caddy. Put the white mug on the plate and put the chocolate pudding to the right of the plate. Put both the alphabet soup and the cream cheese box in the basket. Put both moka pots on the stove. Put the yellow and white mug in the microwave and close it.

Table 7: LIBERO suite Results

	Spatial	Object	Goal	Long	Average
Mean $\pm$ Std	96.60 ± 0.70	98.00 ± 0.80	96.30 ± 1.10	92.70 ± 2.10	95.90 ± 0.70

D.3 Success Rates and Run-to-Run Variability

Tables 7 and 8 report the mean success rates and sample standard deviations of GIVLA across ten independent stochastic training runs on LIBERO and LIBERO-plus, respectively. All runs follow the same training and evaluation protocol. For LIBERO-plus, models trained on the standard LIBERO mixture are evaluated directly under the seven perturbation categories without additional training or adaptation. These statistics complement the mean success rates reported in the main text by quantifying run-to-run variability. We report uncertainty estimates for GIVLA; corresponding estimates are unavailable for external baselines whose results are taken from published sources. All success rates are expressed as percentages.

Table 8: LIBERO-plus Perturbation Results

	Camera	Robot	Language	Light	Background	Noise	Layout	Total
Mean ± Std	46.42 ± 2.04	47.97 ± 2.65	70.98 ± 4.29	87.01 ± 2.22	75.41 ± 3.14	86.89 ± 3.51	68.00 ± 3.44	68.95 ± 2.14

E Details of compared models

E.1 2D Vision-Language-Action (VLA) Baselines

OpenVLA [27]: OpenVLA is a large-scale Vision-Language-Action (VLA) model with 7B parameters. Its backbone utilizes the Llama 2[42] language model, integrated with a visual encoder that fuses features from DINOv2[43] and SigLIP[44] to enable robust spatial reasoning and semantic understanding. In terms of action generation, the model employs a discrete autoregressive algorithm. It was pretrained on the OpenX-Embodiment dataset[26], comprising 970,000 real-world robot demonstrations. The core methodology involves synthesizing Internet-scale vision-language data with robotic demonstrations.

π_0 [29]: π_0 is a 3.5 billion (3.5B) parameter model based on the pretrained PaliGemma[45] Vision-Language Model. For action generation, the model departs from traditional binning approaches by adopting a continuous action algorithm based on Flow Matching[46], enabling the generation of high-frequency and dexterous motor commands. It was trained on diverse large-scale datasets across multiple robotic platforms. Methodologically, it builds a flow-matching architecture atop the pretrained VLM and utilizes a two-stage training strategy: an initial broad pretraining phase on cross-embodiment datasets, followed by optional high-quality fine-tuning for specific tasks such as laundry folding.

π_0-fast [29]: π_0-FAST is a 3.5 billion (3.5B) parameter model that shares the same backbone and underlying PaliGemma[45] VLM as π_0. Unlike the base version, this model utilizes a discrete autoregressive algorithm for action generation. Its primary innovation is the introduction of Frequency-domain Action Sequence Tokenization (FAST), which employs the Discrete Cosine Transform (DCT) to map continuous action trajectories into the frequency domain. By concentrating core information into a few low-frequency coefficients, the method compresses actions into compact discrete tokens. This replaces conventional binning schemes, effectively reducing training time by a factor of five without sacrificing performance.

WorldVLA [38]: WorldVLA is an autoregressive action world model that unifies the understanding and generation of both actions and images, initialized from the Chameleon model[47]. The system integrates three tokenizers: image (VQ-GAN), text (BPE), and action. The action tokenizer discretizes continuous movements into 256 bins represented by 7 tokens. It utilizes a selective attention masking mechanism to optimize sequence prediction and leverages world-model-based future image prediction to learn environmental physics, creating a synergy between action modeling and visual understanding.

NORA [39]: NORA is a 3B parameter model based on the Qwen-2.5-VL[48] multimodal backbone. For action generation, it employs a discrete autoregressive algorithm equipped with a FAST+ tokenizer. Trained on 970,000 real-world demonstrations, NORA aims to minimize computational overhead while leveraging the strong visual-semantic reasoning capabilities of the Qwen series to enhance action anchoring, providing a practical solution for real-time robotic environments.

UniVLA [40]: The UniVLA framework is built upon Prismatic[49] 7B and utilizes the DINOv2[43] feature space for visual perception. The model generates actions using a discrete autoregressive algorithm and learns primarily from Internet-scale video data. At the core of the framework is a Latent Action Model based on VQ-VAE, which extracts task-centric latent action representations from paired video frames within the DINOv2 feature space. These quantized latent actions serve as pseudo-labels to train the vision-language policy, allowing the model to leverage diverse data sources, including human videos, for efficient cross-embodiment learning.

CoT-VLA[36]: CoT-VLA is a 7 billion (7B) parameter model based on the VILA-U[50] multimodal foundation model. The model employs a discrete autoregressive algorithm for action generation and was trained using robotic demonstrations from the OpenX-Embodiment dataset[26]. It introduces a Visual Chain-of-Thought (CoT) mechanism combined with a hybrid attention design: before predicting action sequences, the model first generates future sub-goal images in an autoregressive

manner using causal attention as an intermediate reasoning step, followed by the generation of short action sequences via a full attention mechanism. The training process incorporates additional action-free video datasets, such as EPIC-KITCHENS[51], to enhance logical reasoning.

SmolVLA[31]: SmolVLA is an extremely lightweight model with 0.45B parameters, utilizing the SmolVLM-2[52] as its backbone. SmolVLA uses a Flow Matching-based action expert for continuous action-chunk generation. It also supports asynchronous inference to improve deployment responsiveness. The model focuses on lowering deployment barriers, supporting training on a single GPU and deployment on consumer-grade GPUs or CPUs. To improve responsiveness, it introduces an asynchronous inference stack that decouples perception and action prediction from execution.

GR00T-n1[37]: GR00T N1 is a 2B parameter model specifically designed for humanoid robots. It features a dual-system architecture: System 2 consists of the Eagle-2 VLM[53] for environment interpretation, while System 1 is a Diffusion Transformer module. For action generation, the model uses a continuous algorithm based on the Diffusion Transformer. By training on a heterogeneous mixture of real trajectories, human videos, and synthetic data, the two systems are tightly coupled to generate fluid motor actions in real-time.

Evo-1[32]: Evo-1 is a lightweight 0.77B parameter model using a cropped version of the native multimodal InternVL3[9]. Action generation is performed via a continuous algorithm based on a Cross-Modulated Diffusion Transformer. A key characteristic of Evo-1 is its ability to achieve high performance without extensive pretraining on robot data. Through optimized integration modules and a two-stage training paradigm, the model progressively aligns action with perception, preserving the original representation power of the VLM.

E.2 Explicit 3D VLA

SpatialVLA [14]: SpatialVLA is a 4B parameter model that adopts PaliGemma2[54] as its vision-language backbone. Action generation is implemented through a discrete autoregressive algorithm using Adaptive Action Grids. Relying on 1.1 million real-world robot clips, it introduces Ego3D positional encodings to inject 3D spatial context from depth information into input observations. Additionally, the proposed Adaptive Action Grids discretize continuous robot motion based on data distribution, allowing for efficient fine-tuning on new robotic setups by simply re-discretizing the pre-learned grids.

E.3 Implicit 3D VLA

DepthVLA[17]: DepthVLA is a 3.5B parameter model based on the PaliGemma VLM[45]. For action generation, it utilizes a continuous diffusion algorithm. The architecture features a Mixture-of-Transformers design that unifies the VLM, a depth transformer[55], and action experts within an end-to-end model, where all three components share a unified attention mechanism. DepthVLA explicitly introduces spatial awareness through a pretrained depth prediction module, specifically aimed at enhancing robustness in tasks requiring precise spatial reasoning.

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 abstract and introduction accurately summarize our core contributions, including the GIVLA framework and the discovery of perception-action gradient conflicts.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We provide a dedicated discussion on limitations regarding model scaling, lack of explicit safety constrain and the performance upper bound of upstream experts in conclusion.

   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: Our work is primarily empirical and does not introduce new theoretical theorems or formal mathematical proofs.

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: We provide comprehensive architectural details, hyperparameter settings, and training paradigms in Method section and Appendix B.

   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: We provide the complete codebase, including model architecture, training scripts, and detailed instructions for reproducibility in our supplementary material.

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: Detailed experimental configurations, including optimization schedules and hardware setups, are specified in Experiment section and Appendix D.

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: [Yes]

Justification: We report the average success rates across multiple independent training runs to ensure statistical reliability.

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: We disclose the specific usage of NVIDIA H20 ,RTX 4090 GPUs and RTX 4070 GPUs for simulation and real-world experiments in Appendix B.3.

   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: Our research follows the standard ethical practices for robotic learning and does not involve any sensitive data or unethical procedures.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the benefits of parameter-efficient VLA for accessible robotics and explicitly address the safety risks of end-to-end models in human-centric environments in Conclusion section.

    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: Our model is focused on low-level robotic manipulation in laboratory settings and does not pose a high risk for standard societal misuse.

    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: All utilized foundation models and benchmarks (InternVL-3, Depth-Anything-v3, LIBERO) are properly cited and used according to their respective licenses.

    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 GIVLA model and its accompanying training scripts are new assets provided in the repository.

    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: Our research did not involve human subjects or crowdsourced data collection.

    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: Our study is limited to autonomous robotic benchmarks and physical hardware experiments without human involvement.

    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: LLMs were only used for standard writing and grammatical editing, which does not impact the core methodology or scientific rigor.

    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.