Understanding the Impact of Geometric Foundation Models on Vision-Language-Action Models

Yurou Yang[1], Muyuan Lin[1], Roberto Martin-Martin[1,2], Martin Labrie[1], Shreekant Gayaka[1], Cheng-Hao Kuo[1], and Luca Carlone[1,3]

Amazon.com[1] University of Texas at Austin[2] MIT[3]

Abstract. Recent work explores new opportunities at the intersection of vision-language-action models (VLAs) and geometric foundation models (GFMs) for 3D reconstruction, such as VGGT. While the resulting *geometric* VLAs often show improved performance, it remains unclear (i) if modern VLAs already have sufficient geometric understanding to start with, (ii) what is the best architecture to inject geometric understanding into a VLA, and (iii) what is the effect of other design choices that affect geometric VLAs. In this paper we provide a rigorous experimental analysis to shed light on these questions, for a specific choice of VLA (GR00T-N1.5) and GFM (VGGT). Our first contribution is to formalize prior work's intuition that current VLAs lack geometric understanding, by providing a rigorous analysis based on linear probing. The analysis quantifies, for the first time, the "geometric gap" between VLAs and GFMs. Our second contribution is to identify and compare different strategies to bridge GFMs with VLAs. We implement three different architectures, which differ in the way they inject geometry in the VLA, while keeping low-level implementation details as similar as possible, to ensure a fair comparison. Finally, we analyze the impact of non-architectural choices (e.g., training data, number of cameras, reconstruction quality) on the performance of the geometric VLAs.

1 Introduction

Vision-Language-Action models (VLAs) have recently been at the center stage of robotics research for their ability to enable generalist manipulation behaviors from a modest number of expert demonstrations. These models have enabled successful manipulation skills from text instructions (e.g., "pick up the red mug and put it in the sink") and are steadily permeating from academic labs to industrial applications (e.g., [15, 36, 40]). Typical VLAs are composed of two main building blocks: a *vision-language model* (VLM), which parses input images collected by the robot and language instructions from the user, and an *action expert*, which takes the output of the VLM and computes a suitable action (or chunk of actions) for the robot to execute. Intuitively, the VLM helps the robot correlate the user's instructions ("pick up the red mug") with relevant pixels in the image, while the action expert is in charge of establishing the best course of action depending on the robot embodiment and the VLM understanding.

While the combination of VLMs and action expert has demonstrated to be a strategic and powerful architectural choice, VLMs have been observed to be poor at spatial understanding [8, 9]. At the same time, spatial understanding is intrinsically related to manipulation, where distances and geometry make the difference between a successful physical interaction and a failed one. This insight triggered work on *3D* VLAs, where the VLA policy is augmented with depth information from an RGB-D camera [10, 17, 23, 28, 37, 49].

Parallel work in computer vision has pioneered the use of feed-forward transformer architectures for 3D reconstruction. Contrary to traditional SLAM and Structure from Motion approaches, which decouple the problem into multiple stages (e.g., feature extraction, matching, bundle adjustment, dense 3D reconstruction), these new *geometric foundation models* (GFM) reconstruct camera poses, depth maps, and point maps of the observed scene from a collection of images (e.g., [14, 34, 38, 41, 42]). The advantage of these approaches lies in their simplicity, their zero-shot generalization, and their ability to work with uncalibrated cameras. These new methods have created new opportunities to enhance VLAs with geometric understanding without altering their input data. This is particularly important since VLAs are typically trained on RGB inputs and GFMs open the door to adding 3D information without requiring additional data collection. These opportunities have been explored in a number of very recent works [1, 16, 29, 30, 33, 48], which report promising results.

Despite the quick progress in combining VLAs and GFMs in a so-called *geometric VLA*, related work leaves several questions unanswered. First of all, it remains unclear if VLAs already have sufficient geometric understanding and how to quantify a potential gap in 3D understanding. Second, related works propose disparate architectures to inject 3D information from GFMs into VLAs and it is unclear how these architectures compare or what is the insight behind one architecture being better than another. Finally, it is unclear how external factors related to geometry, including number of cameras, size of the training set, and reconstruction quality, impact performance of geometric VLAs.

Contributions. In this paper we provide a rigorous experimental analysis to shed light on the open questions above, for a specific choice of VLA (GR00T-N1.5) and GFM (VGGT). Our first contribution (Section 3) is to **identify and compare different strategies to bridge GFMs and VLAs**. Towards this goal, we analyze related work and identify three different strategies, summarized in Fig. 1. We provide prototype implementations of models using the three different strategies, while keeping low-level implementation details as similar as possible (to ensure an apple-to-apple comparison). Our second contribution (Section 4) is to formalize related work's intuition that current VLAs lack geometric understanding, by providing a **rigorous analysis based on linear probing**. The probing assesses the VLA ability (more specifically, the ability of the VLM within the VLA) to predict dense depth information from images. The analysis quantifies, for the first time, the "geometric gap" between VLAs and GFMs, rigorously grounding observations in related work [29]. Our analysis also provides the surprising finding that a pretrained LLM can predict dense

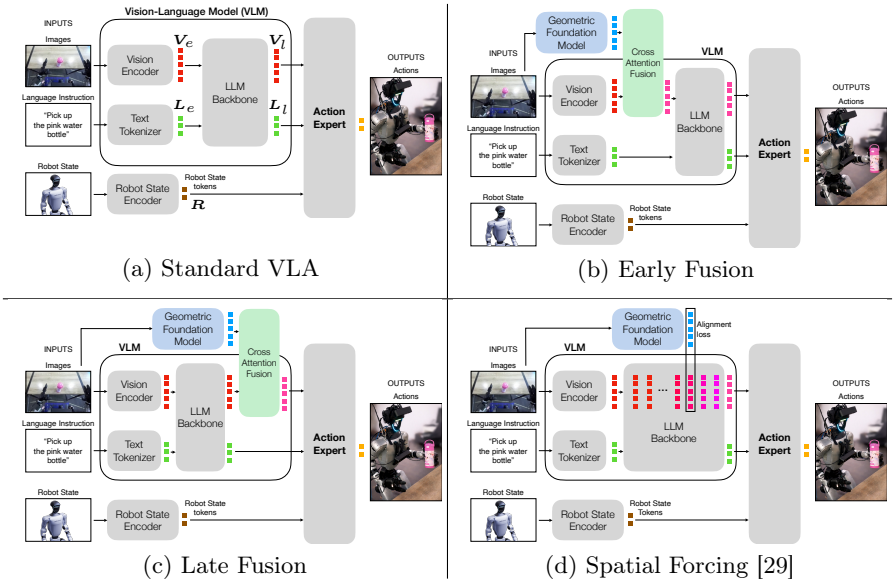


Fig. 1: (a) Standard VLA architecture. (b)-(d) Key strategies to inject tokens from a geometric foundation model (e.g., VGGT) into the VLA.

depth information if it is fed with tokens from a GFM. Our final contribution (Section 5) is to **analyze the effect of key design choices that affect geometric VLAs**. Relevant factors include both architectural aspects (such as the choice of fusion strategy discussed above), as well as non-architectural aspects (such as the number of cameras used by the VLA, the size of the training data, and the quality of the reconstruction from the GFM). We perform our analysis on popular simulation benchmarks, including RoboCasa [32] and LIBERO [31], and we also provide an evaluation on real data, using the Unitree G1 humanoid.

2 Related Work

Vision-Language-Action Models (VLAs) for Manipulation. VLAs map visual observations and natural language instructions to robot actions by extending VLMs with a policy or action head [4–7,13]. These models inherit the semantic grounding and generalization capabilities of large VLMs, enabling robots to perform diverse manipulation tasks from relatively limited demonstration data. However, despite their impressive generalization capabilities, VLAs inherit a key limitation from VLMs: they primarily operate on RGB inputs and focus on semantic understanding [2] rather than explicit spatial reasoning. Prior work has shown that vision-language models often struggle with geometric reasoning tasks such as estimating distances, relative pose, or object placement constraints [8,37]. While recent work attempts to improve spatial reasoning through architectural modifications or large-scale multimodal pretraining [12, 37, 49, 51], it remains

unclear whether these approaches provide the level of geometric understanding required for efficient robotic manipulation.

Geometric Foundation Models. Recent work in computer vision has introduced *geometric foundation models* (GFMs), which learn to infer 3D scene structure directly from images using a single feed-forward transformer-based architecture [27, 43]. Models such as the Visual Geometry Grounded Transformer (VGGT) [42] predict camera poses, depth maps, and dense point maps from one or multiple views without the multi-stage pipelines used in traditional SLAM or structure-from-motion systems. Subsequent work has further developed these models with improved geometric consistency by jointly predicting correlated geometric quantities such as depth, normals, and point maps [43], incorporating additional modalities or supervision to enable metric reconstruction and multi-view consistency [24, 41], or extending these architectures to dynamic scenes and semantic understanding, enabling unified feed-forward modeling of geometry, semantics, and motion [21, 38, 52]. Together, these models provide powerful geometric representations that generalize across environments and can be queried without scene-specific training. These techniques have recently been shown to improve spatial understanding in VLMs [20, 45, 50].

Injecting Geometry into VLAs. Several works augment VLAs with geometric information to address the limited spatial reasoning capabilities of traditional VLM-based policies. Some approaches incorporate explicit geometric inputs such as depth maps or point clouds, for example PointVLA [28] and RVT-2 [17]. Other works attempt to learn spatial structure within the policy itself, through 3D feature learning, affordance reconstruction, or spatial memory mechanisms [10, 23, 39]. More recently, geometric foundation models have been used to inject geometric context into VLAs, either by fusing geometric tokens with visual representations [1, 16, 30, 33] or by aligning vision-language representations with geometric features during training [29]. Despite promising results, existing approaches differ substantially in how geometry is integrated, making it difficult to understand which architectural choices are most effective. In this work, we provide a systematic analysis of several strategies for integrating geometric foundation models into modern VLAs.

3 Bridging VLAs and Geometric Foundation Models

This section reviews the typical VLA architecture and discusses ways to integrate GFMs in a VLA (Section 3.1). Then, we describe our implementation of the "fusion module" that fuses VLA tokens with tokens from the GFM (Section 3.2).

3.1 VLA Architecture and Key Strategies to Inject Geometry

VLA Architecture. A typical VLA is shown in Fig. 1(a). The VLA takes three inputs: a set of images $\mathcal{I}$ (e.g., captured by the robot cameras at the current time t), a language instruction $\mathcal{L}$ (e.g., "pick up the red mug"), and robot state information $\mathcal{R}$ (e.g., joint angles). The inputs are first processed by specific

encoders that produce visual ($\boldsymbol{V}_e$), language ($\boldsymbol{L}_e$), and robot state tokens ($\boldsymbol{R}$), respectively. Visual and language tokens are then passed to the LLM backbone inside the VLM that produces new visual tokens ($\boldsymbol{V}_l$) and language tokens ($\boldsymbol{L}_l$). These tokens, together with the robot state tokens $\boldsymbol{R}$, are passed to the action expert, which computes the set of actions for the next T time steps, $\boldsymbol{a}_{0:T}$.

The VLA is trained from demonstrations, given tuples $(\mathcal{I}, \mathcal{L}, \mathcal{R}, \boldsymbol{a}_{0:T})$, including both input data $(\mathcal{I}, \mathcal{L}, \mathcal{R})$ and the desired action sequence, $\boldsymbol{a}_{0:T}$, given in the demonstration. While different architectures differ in the choice of the VLM, the action expert, and training objective, a popular choice is to use a diffusion policy formulation [11], where the action expert integrates a diffusion transformer. In a diffusion policy formulation, one iteratively adds Gaussian noise to the clean action sequence, $\boldsymbol{a}_{0:T}$, to obtain a noisy sequence $\mathbf{a}^k$, $k = 1, \ldots, N$, such that:

$$\mathbf{a}^k = \alpha_k \boldsymbol{a}_{0:T} + (1 - \alpha_k)\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

The diffusion policy, parametrized by weights θ, is then trained with a flow-matching loss, which teaches the policy how to map back random noise samples to plausible actions. Concretely, the policy predicts the denoising field $v_\theta(\mathbf{a}^k, \boldsymbol{T}, k)$, conditioned on the visual, text, and robot state tokens, $\boldsymbol{T} = (\boldsymbol{V}_l, \boldsymbol{L}_l, \boldsymbol{R})$, and is trained using the following flow-matching loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_k \left[\left\| v_\theta(\mathbf{a}^k, \boldsymbol{T}, k) - (\boldsymbol{\epsilon} - \boldsymbol{a}_{0:T}) \right\|_2^2 \right]. \tag{2}$$

The predicted robot actions are then obtained via (Euler) integration of the field $v_\theta(\mathbf{a}^k, \boldsymbol{T}, k)$; see [4, 6] for details and popular instantiations of these ideas.

Early Fusion. Now let us consider a first strategy to inject GFM tokens into the baseline VLA described above. This first strategy, that we call "Early Fusion", is illustrated in Fig. 1(b). More specifically, the GFM takes the same images $\mathcal{I}$ as the VLA and produces tokens $\boldsymbol{G}$; in our experiments, we use the VGGT backbone to produce these tokens. Then, visual tokens $\boldsymbol{V}_e$ from the VLA and geometric tokens $\boldsymbol{G}$ from the GFM are fused together to produce new tokens $\texttt{fuse}(\boldsymbol{V}_e, \boldsymbol{G})$, which are passed to the LLM in lieu of the tokens from the vision encoder. Intuitively, the Early Fusion strategy treats the GFM as an additional encoder. This approach has been used in early papers combining VGGT with VLAs [30]; moreover, it resembles injection strategies used to enhance spatial understanding in VLMs [50]. We implement the fusion $\texttt{fuse}(\boldsymbol{V}_e, \boldsymbol{G})$ as a cross-attention layer with attention gating, as discussed in Section 3.2 below. We remark that the use of attention gating was not explicitly mentioned in [30, 50], but we found this aspect to be crucial to the effectiveness of this strategy (see the supplementary material).

Late Fusion. The second strategy, that we call "Late Fusion", is illustrated in Fig. 1(c). Again, the GFM takes the same images $\mathcal{I}$ as the VLA and produces geometric tokens, $\boldsymbol{G}$. However, in this case the geometric tokens are fused with the visual tokens outputted by the LLM, namely $\boldsymbol{V}_l$; the fused tokens, $\texttt{fuse}(\boldsymbol{V}_l, \boldsymbol{G})$, are used as input by the action expert instead of the original LLM tokens, $\boldsymbol{V}_l$. Intuitively, the Late Fusion strategy attempts at enriching the VLM

output with more geometric information before passing it to the action expert. This approach is conceptually similar to [48], but with a simplified architecture. As before, we implement the fusion $\texttt{fuse}(\boldsymbol{V}_l, \boldsymbol{G})$ as a cross-attention layer with attention gating, to keep the implementation as close as possible across architectures.

Spatial Forcing. The third strategy is Spatial Forcing [29] (see Fig. 1(d)). In this case, the architecture itself is the same as the original VLA with the main difference being that, at training time, spatial forcing adds an *alignment loss* to encourage internal tokens of the LLM to align (as measured by cosine similarity) with the GFM tokens. This facilitates the VLM to retain geometric information that could potentially be used by the action expert.

Remark 1 (Our evaluation focuses on GR00T and VGGT). At the time of preparation of this manuscript, no open-source code was available for approaches following the Early Fusion and Late Fusion strategy. Therefore, we developed prototype implementations for both strategies, building on top of the GR00T-N1.5 VLA [3]. For consistency, we also implemented Spatial Forcing [29] using GR00T: while open-source code is available for Spatial Forcing, it is based on OpenVLA [25] and π_0 [6]; we adapted to GR00T for a fair comparison. In all tests, we used VGGT as the geometric foundation model.

3.2 Cross-Attention Fusion

In this section, we briefly discuss the implementation of the fusion module $\texttt{fuse}(\cdot,\cdot)$ mentioned in Section 3.1. We adopt a similar implementation for both the Early Fusion and the Late Fusion strategy. Let us denote with $\boldsymbol{X} \in \mathbb{R}^{P_x \times D_x}$ the VLA visual tokens, where P_x denotes the number of tokens (this typically depends on the number of patches the encoder subdivides the images in) and D_x is the size of the feature associated to each token; for the Early Fusion, we set $\boldsymbol{X} = \boldsymbol{V}_e$, while for the Late Fusion we set $\boldsymbol{X} = \boldsymbol{V}_l$. Also, denote with $\boldsymbol{G} \in \mathbb{R}^{P_g \times D_g}$ the tokens of the geometric foundation model. Below we show how to obtain fused tokens $\tilde{\boldsymbol{X}} = \texttt{fuse}(\boldsymbol{X}, \boldsymbol{G})$ using a cross-attention mechanism with attention gating, such that $\tilde{\boldsymbol{X}} \in \mathbb{R}^{P_x \times D_x}$ has the same size of the VLA tokens, hence remaining compatible with the rest of the architecture.

We follow a standard cross-attention implementation, where tokens are first projected onto a common space using learned projection matrices $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$:

$$\mathbf{Q} = \boldsymbol{X}\,\boldsymbol{W}_Q \in \mathbb{R}^{P_x \times d}, \quad \mathbf{K} \;= \boldsymbol{G}\,\boldsymbol{W}_K \in \mathbb{R}^{P_g \times d}, \quad \mathbf{V} = \boldsymbol{G}\,\boldsymbol{W}_V \in \mathbb{R}^{P_g \times d}. \tag{3}$$

Then, cross-attention is computed as follows:

$$\boldsymbol{Y} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \in \mathbb{R}^{P_x \times d} \tag{4}$$

and the result is projected back to the VLA feature space using a learned projection matrix $\boldsymbol{W}_O$:

$$\boldsymbol{Z} = \boldsymbol{Y}\,\boldsymbol{W}_O \in \mathbb{R}^{P_x \times D_x}. \tag{5}$$

Finally, we apply a learned attention gate $\boldsymbol{A}$ to the cross-attention results and use the gated results as a residual correction term on the VLA tokens:

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{A} \odot \boldsymbol{Z} \tag{6}$$

where $\odot$ denotes the component-wise (Hadamard) product. The gate is initialized close to zero, such that the geometric tokens are gradually added to the VLA tokens; this aspect is important since the rest of the architecture is pretrained without the geometric tokens, and introducing geometric tokens too abruptly prevents the action expert from correctly using them (see the supplementary material). In our implementation, we follow a standard practice and parameterize the linear projections $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_O$ as low-ranking (LoRA) layers [19] and add positional encodings to the VLA and GFM tokens to retain the spatial arrangement of the corresponding tokens. We refer the reader to the supplementary material for more details.

4 Probing Geometric Understanding in VLAs

Key Takeaway. GR00T-N1.5 is not able to retain depth information, which is already lost before the output of the VLM. Late fusion, as expected, can inject depth understanding. Surprisingly, even Early Fusion can inject depth understanding, without requiring finetuning of GR00T's VLM.

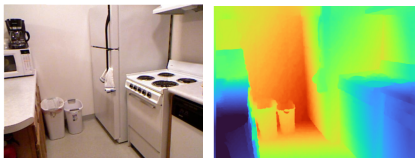
Before investigating the impact of injecting geometric tokens into a VLA, we ask a basic question: do VLAs already understand geometry? We rigorously answer this question using a technique known as *linear probing*. In linear probing, one attaches a simple MLP (the linear probe) to a layer of a frozen network, and then trains the MLP to solve a task (in our case, monocular depth estimation) to probe the information content of that layer: intuitively if the layer contains enough information (e.g., about geometry), then the task can be completed successfully, otherwise it cannot. Qualitative probing results have been reported in related work (e.g., [29]), but we provide the first *quantitative* evaluation of the geometric understanding gap between VLAs and geometric foundation models.

Probing Setup. We train the linear probe for 10 epochs on the NYU Depth V2 dataset (we use the version of the dataset from [44]). The dataset includes pairs of RGB and depth images, including 24,000 training pairs, and 645 pairs for validation. For probing purposes, we use a Scale-Invariant Logarithmic (SILog) loss following standard practice [46]. To evaluate probing performance, we compute the RMSE depth error (↓), as well as the δ_1 score (↑), which measures the fraction of pixels with predictions within 25% of the ground truth depth.

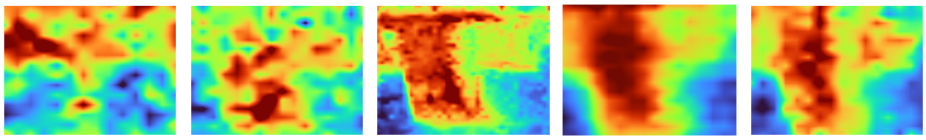
Probing Results. With reference to Fig. 1(a), we start by probing the output of the vision encoder, as well as the output of the overall GR00T VLM. Again, in both cases, the VLA is frozen at the original pre-trained weights and the linear probe is the only module that is trained. The first two rows in Table 1 report the corresponding scores. For reference, we also perform linear probing

	RMSE [m] (↓)	δ_1 (↑)
GR00T - vision encoder probe	0.92	0.51
GR00T - VLM probe	0.73	0.63
VGGT probe	**0.41**	**0.89**
Early Fusion probe	0.44	0.88
Late Fusion probe	0.45	0.87

Table 1: RMSE depth errors and δ_1 scores for the linear probing experiments.



(a) RGB **(b)** GT depth



(c) VLA 1 **(d)** VLA 2 **(e)** VGGT **(f)** Early **(g)** Late

Fig. 2: Sample linear probing results: (a) RBG input, (b) ground truth (GT) depth, (c) depth predicted by the GR00T vision encoder probe, (d) GR00T VLM probe, (e) VGGT probe, (f) Early Fusion probe, and (g) Late Fusion probe. The poor performance of the VLA probes support the intuition that VLAs lose geometric understanding, while *geometric* VLAs recover most of it thanks to the VGGT tokens.

using the output of the VGGT backbone and report the results in the third row of Table 1: intuitively the VGGT tokens definitely encode geometric understanding and we can think about the gap between the performance of the VGGT probe and the VLA probes as a measure of the "geometric gap", which quantifies the amount of geometric information loss in the VLA. From Table 1, we observe that the depth understanding in VGGT is substantially higher compared to the VLA, with the RMSE almost doubling between VGGT and the VLA. The table also shows that the depth information is already lost after the vision encoder.[1]

The results so far confirm that using VGGT has the potential to supplement the VLA with a better depth understanding. To further reinforce this possibility, we perform linear probing on the Early Fusion and Late Fusion models, to show that those architectures are indeed able to inject geometric information into the VLA. In this case, since we do not have pre-trained weights for the cross-attention fusion module, we train both the fusion module and the linear probe. In both cases, we probe the output of the VLM, to make sure geometric information is retained. Note that Spatial forcing does not entail architectural modifications hence it is excluded from this comparison. The results are reported in the last two rows of Table 1. The Late Fusion architecture is able to retain geometric information. This is somewhat expected, since the VGGT tokens are injected near the probe: intuitively, the attention gate in (6) can prioritize VGGT tokens and provide relevant geometric information to the linear probe. More surprisingly, even the Early Fusion architecture is able to correctly inject geometric informa-

[1] The attentive reader might find surprising that the probe at the output of the VLM has more depth information than the output of the vision encoder (intuitively, if the depth information is lost after the vision encoder, how can it be recovered at the output of the VLM?). This is an artifact of the analysis: the hidden dimension for vision encoder probe is 1024, while for the VLM and VGGT probes is 2048; therefore, the vision encoder probe is impacted by the lower capacity of the MLP.

tion, achieving depth performance almost on par with VGGT. This is nontrivial since we are injecting the VGGT tokens *before* the LLM and the LLM is frozen: nevertheless, the LLM is still able to use the VGGT tokens to produce dense depth predictions. We provide qualitative depth prediction results in Fig. 2. We remark that these are the results of a *linear* probe, hence —even when accurate— they are more noisy than the ones produced by, e.g., a DPT [42]. Next, we are ready to assess whether the additional geometric information in the Early and Late Fusion models is actually helpful to improve manipulation.[2]

5 Impact of Key Design Choices

In this section, we provide a rigorous evaluation to assess the impact of key design choices on the performance of the geometric VLAs in Section 3. After introducing the evaluation protocol (Section 5.1), we investigate the impact of the fusion strategy (Section 5.2), the training data size (Section 5.3), the number of cameras (Section 5.4), and the VGGT reconstruction quality (Section 5.5).

5.1 Experimental Protocol

Benchmarks. We evaluate performance on two simulated benchmarks (RoboCasa and LIBERO) and a real benchmark (using a Unitree G1 humanoid).

RoboCasa [32] is a large-scale simulation benchmark. The benchmark includes 8 tasks: PnPCabToCounter, PnPCounterToCab, PnPCounterToMicrowave, PnPCounterToSink, PnPCounterToStove, PnPMicrowaveToCounter, PnPSinkToCounter, PnPStoveToCounter. For instance, PnPCabToCounter indicates a Pick-and-Place (PnP) task that moves an object from the cabinet to the kitchen counter. For each task, the benchmark provides 5 different episodes, each one corresponding to a different kitchen scenario and a different object to grasp. For each of the 5 episodes, we repeated 15 trials, where the appearance of the scene is randomized, for a total of 600 manipulation experiments. We follow the standard evaluation protocol and use three cameras (left, right, and wrist cameras).

LIBERO [31] is a standard simulation benchmark and includes four tasks: LIBERO-SPATIAL, LIBERO-OBJECT, LIBERO-GOAL, and LIBERO-100. LIBERO-100 includes 90 short-horizon tasks (LIBERO-90) and 10 long-horizon tasks (LIBERO-LONG). Each task is evaluated over 500 trials. We follow the standard evaluation protocol, including the use of both primary and wrist-mounted cameras.

We also benchmark on real robot data using a *Unitree G1 humanoid*. The G1 benchmark focuses on evaluating pick-and-place performance for 3 classes of objects: bottle, ball, and box. We repeated 90 tests across the three objects, randomizing object location, object orientation, and object instances within each

[2] Inside the supplementary material we provide an additional experiment where we probe surface normal prediction. Surface normals describe how a robot can interact with an object by encoding local contact geometry and feasible force directions. The results are consistent with the findings in this section, further confirming the lack of geometric understanding in GR00T and the potential to regain it using VGGT.

class (e.g., from a small wooden box to a larger cereal box for the "box" class). In each test, there is a single object on the table and the text prompt first commands the robot to pick up the object, and then to drop if off on the table. To position the target objects in a repeatable manner across experiments, we overlaid a grid on the table surface and randomly generated the object's grid coordinates and orientation. We use the same configuration across all models.

Performance Metrics. We use success rate as key performance metric, but in certain experiments we also break down the success rate across different grasping stages. For each model, we evaluate the models at the following checkpoints/epochs: 1, 5, 10, 15, 18, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, and report the best success rate across checkpoints. For each result, we compute significance levels using the two-sided McNemar's test (more details in the supplementary material): we observed that the randomness in the action expert can produce fluctuations in the results (even after fixing the seed for the scenario generation), hence distinguishing randomness from actual performance differences is key (see the supplementary material for numerical results).

Experimental Considerations. We fix the random seeds for repeatability, without any attempt at tuning the seed for performance. We also fix the noise generation in the diffusion transformer of GR00T. Experimental settings are identical across all models. We train all models using an NVIDIA A100 GPU cluster with 320 GB GPU Memory for 100 epochs. Finetuning takes 2-3 days (depending on the model), while mid-training (Section 5.3) takes around 1 week.

Implementation Details. When finetuning GR00T, we follow the standard protocol and only train the action expert starting from pre-trained weights. For the Early Fusion and the Late Fusion model, we train the fusion module and the action expert and keek everything else frozen. For Spatial Forcing, we apply the alignment loss to layer 9 (out of 13) of the GR00T LLM and only finetune the linear projection between the vision encoder and the LLM. We also attempted to finetune the LLM in Spatial Forcing, but obtained worse results (see the supplementary material).

5.2 Impact of Fusion Strategy

Key Takeaway. Task-level finetuning of geometric VLAs combining GR00T-N1.5 and VGGT does not lead to a statistically significant increase in the success rate. The Early Fusion geometric VLA appears to be more promising in real world experiments.

In this section, we compare the three geometric VLA models from Section 3.

RoboCasa Results. Table 2 reports success rates across the 8 RoboCasa tasks, including both the implemented geometric VLAs (Early Fusion, Late Fusion, and Spatial Forcing) and baselines from the literature. For each, we also report p values for statistical significance, all computed with respect to the GR00T-N1.5 baseline. For each geometric VLA, we finetune on the specific task using the demonstrations provided by RoboCasa (around 300 demonstrations per task).

Moreover, we finetune and test GR00T-N1.5 to ensure a fair comparison. We postpone the discussion of the last row of the table to Section 5.3.

	CabToCtr	CtrToCab	CtrToMicrowave	CtrToSink	CtrToStove	MicrowaveToCtr	SinkToCtr	StoveToCtr	Average
DP3 [47]	4.0	2.0	6.0	0.0	0.0	6.0	0.0	0.0	2.3
Pi0 [6]	28.0	18.0	36.0	70.0	36.0	22.0	16.0	44.0	33.8
Pi0-Fast [35]	30.0	48.0	20.0	56.0	64.0	46.0	62.0	60.0	48.3
RS-CL [26]	**60.0**	68.0	40.0	68.0	72.0	48.0	68.0	54.0	59.0
DP-VLA [18]	10.0	32.0	56.0	30.0	22.0	18.0	56.0	62.0	35.8
GR00T N1 [4]	20.0	36.0	13.0	10.0	24.0	16.0	33.0	29.0	22.6
Video Policy [22]	48.0	52.0	22.0	48.0	54.0	28.0	56.0	70.0	47.3
GR00T-N1.5	42.7	**74.7**	73.3	93.3	77.3	58.7	65.3	88.0	71.7
Early Fusion	32.0 (p=0.186)	69.3 (p=0.289)	65.3 (p=0.377)	88.0 (p=0.388)	73.3 (p=0.664)	62.7 (p=0.742)	80.0 (p=0.019)	86.7 (p=1.000)	69.7 (p=0.399)
Late Fusion	46.7 (p=0.710)	72.0 (p=0.625)	**74.7** (p=1.000)	85.3 (p=0.146)	69.3 (p=0.307)	**69.3** (p=0.115)	69.3 (p=0.689)	81.3 (p=0.267)	71.0 (p=0.806)
Spatial Forcing	29.3 (p=0.123)	68.0 (p=0.227)	66.7 (p=0.499)	76.0 (p<0.001)	72.0 (p=0.454)	60.0 (p=1.000)	**84** (p=0.007)	**90.7** (p=0.804)	68.3 (p=0.154)
Early Fusion (mid-trained)	52.0 (p=0.281)	72.0 (p=0.727)	69.3 (p=0.700)	**94.7** (p=1.000)	**80.0** (p=0.815)	68.0 (p=0.189)	81.3 (p=0.023)	84.0 (p=0.607)	**75.2** (p=0.104)

Table 2: VLA performance comparison on the RoboCasa benchmark. Green indicates best result per column; yellow indicates second best (ties share the same color). p-values are computed against the GR00T-N1.5 baseline. Results for the first 7 rows are borrowed from the corresponding paper.

From a quick glance at the table, the results show that in average the performance of the geometric VLAs is subpar compared to the GR00T baseline (e.g., 69.7% for Early Fusion vs. 71.2% for the baseline). However, closer inspection of the p values suggests that the baseline results are just not significantly different from the baseline: a p value larger than $p = 0.05 - 0.1$ indicates the differences are not statistically significant and might be the result of randomness. We remark that each p value in the "Average" column is not the average of the p values of that row, but it is recomputed accounting for the success rates of the runs across all tasks. We draw similar conclusions from testing on the LIBERO benchmark, see the supplementary material for further details. In conclusion, basic finetuning of geometric VLAs does not fundamentally change success rate in simulated benchmarks. As we will show in the next sections, the conclusion drastically changes when increasing the amount of training data or the sensor configuration.

	Approach	Grasp	Lift	Placement	Overall
GR00T-N1.5	57.78	51.92	85.19	**86.96**	22.22
Early Fusion	**84.44** (p<0.001)	**60.53** (p=0.824)	89.13 (p=1.000)	65.85 (p=0.180)	**27.78** (p=0.511)
Late Fusion	57.78 (p=0.855)	59.62 (p=1.000)	**93.55** (p=1.000)	79.31 (p=0.625)	25.56 (p=0.710)

Table 3: VLA performance comparison on (real) Unitree G1 benchmark. The table reports the overall success rate, as well as the breakdown of the success rate in terms of percentage of cases where the robot successfully approached, grasped, lifted, and placed the target object. Green indicates best result per column; yellow indicates second best.

Unitree G1 Results. Table 3 reports the results of the real manipulation benchmark on the Unitree G1. Since each experiment is time consuming, we focus on the Early Fusion and the Late Fusion models. In this case, we also provide a breakdown of the performance across grasping stages: the Approach stage is successful when the robot approaches and touches the object of interest; the Grasp stage is successful when the robot is able to grasp the object; the Lift stage is successful when the object is correctly lifted above the table; the Placement stage is successfully when the object is correctly placed back on the table. If all the stages are successful, the overall episode is considered successful.

Several considerations are in order. First of all, the overall success rate is higher than baseline (22.22%) for both the Early Fusion (27.78%) and Late Fusion model (25.56%). However, the results again lack strong statistical significance, with p values above 0.1. At the same time, the success rate of the approach stage is statistically better than baseline for the Early Fusion model, achieving 84.44% success rate compared to 57.78% of the GR00T baseline. This might indicate that Early Fusion is able to leverage geometric information to better identify the target object and more precisely move the gripper towards it. In average, Grasp and Lift performance is better for the Early and Late Fusion models compared to baseline, while the placement is slightly worse, but again these observations are assigned relatively large p values. In practice, we observed the Early Fusion approach to more reliably grasp small objects (e.g., the small ball and the small box in our benchmark), which was very challenging for the GR00T baseline instead. The performance gain of the Late Fusion approach is more modest. We conjecture that the VLM (and, in particular, the LLM within the LLM) is more flexible in processing new information (as the one provided by the Early Fusion) —an observation confirmed by the plasticity observed in the probing experiment of Section 4; on the other hand, the action expert is relatively "rigid" and unable to make good use of additional data sources (as in the Late Fusion). Next, we go deeper into our investigation and focus on other aspects that make the advantage of using GFMs more pronounced. We mostly focus on the Early Fusion model, mostly due to the fact that it stands out in real experiments, and we postpone details about other models to the supplementary material.

5.3 Impact of Training Data Scaling

Key Takeaway. Early Fusion's performance largely improves and becomes better than baseline (in a statistically significant manner) after training it on a larger amount of data before finetuning on the specific task.

This section shows that we can draw stronger conclusions about the impact of geometric information if we scale up the training data. For this experiment, we first trained the Early Fusion model on the entire RoboCasa dataset (all 8 tasks) starting from the GR00T's pretrained weights. The basic idea is that this "mid-training" allows the network to better adjust to the VGGT tokens and make better use of the resulting geometric information. After mid-training, we finetune on a specific RoboCasa task, as done in the previous section. The corresponding results are reported in the last row of Table 2. Interestingly, now the Early Fusion approach with mid-training becomes substantially better than baseline, with p values indicating increased statistical significance. To further demonstrate that the advantage is due to VGGT rather than other potential information leakage induced by the mid-training, The supplementary material shows that Early Fusion outperforms GR00T even when both are mid-trained using the same protocol. This observation is important because it suggests that

Method	CabToCtr	CtrToCab	CtrToMicrowave	CtrToSink	CtrToStove	MicrowaveToCtr	SinkToCtr	StoveToCtr	Average
GR00T-N1.5	**12.0**	8.0	33.3	**28.0**	13.3	12.0	26.7	4.0	17.2
Early Fusion	10.7 (p=1.000)	**9.3** (p=1.000)	**38.7** (p=0.503)	17.3 (p=0.115)	**29.3** (p=0.008)	**18.7** (p=0.302)	**34.7** (p=0.307)	**13.3** (p=0.039)	**21.5** (p=0.030)

Table 4: RoboCasa results with single-camera (no mid-training). Green indicates best result per column. p values are computed against GR00T-N1.5 (single camera).

scaling up the training of geometric VLAs may unlock their full potential. Such a large-scale training is indeed possible since it would not require additional training data.

5.4 Impact of the Number of Cameras

Key Takeaway. Even without mid-training, Early Fusion's performance becomes statistically better than baseline when using a single-camera setup.

While we have observed that mid-training already makes the Early Fusion model statistically better than baseline, one might also ponder if —without mid-training— there are factors that could make VGGT more useful for manipulation. One may conjecture that GR00T might already be able to infer depth information from multiple camera views. On the other hand, the advantage of using a geometric foundation model might be more pronounced when using a single camera, since in that case the VLA would not be able to obtain dense geometric understanding via multi-view geometry. To test this hypothesis, we finetune both the Early Fusion model and the GR00T baseline using a single camera (in particular, the left camera) in Robocasa. Table 4 reports success rates for both single-camera models. For both, as expected, the overall performance drastically drops, confirming that multi-view inputs are key to successful grasping. More interestingly, the advantage of Early Fusion becomes more pronounced (21.5% vs. 17.2% for the baseline) and statistically significant ($p < 0.05$), confirming our intuition. This might suggest that, in sensor-constrained settings, geometric foundation models provide a viable option to boost performance.

5.5 Impact of VGGT Performance

Key Takeaway. While the Early Fusion model's success rate is largely task dependent, the success rate is also correlated with VGGT's performance (i.e., smaller depth errors in VGGT correlate with highest success rates).

Finally, we test the impact of the VGGT reconstruction performance on the success rate of the resulting geometric VLA. Intuitively, we expect that in cases where VGGT is out-of-distribution or exhibits larger errors, such errors would propagate to the VLA as well. Conversely, scenarios with more accurate VGGT performance might achieve better success rates. Fig. 3(a) plots —for each of the 5 episodes of the 8 RoboCasa tasks— the success rate for that episode versus the average (RMSE) depth error of the VGGT reconstruction.

The depth error is averaged across all images and cameras in the episode. First of all, Fig. 3(a) shows that the depth errors are relatively small, confirming the outstanding generalization capabilities of VGGT, which performs well despite the relatively low-resolution camera renderings (qualitative results in Fig. 3(b)). More importantly, we can inspect the correlation between depth quality and success rate. While the success rate is largely impacted by other factors beyond the VGGT reconstruction quality (such as the task complexity), we can see that the right-hand side of the plot shows a decreasing trend, meaning that lower RMSE corresponds to higher success rates. More formally, we can measure the correlation between depth reconstruction error and success rate: using the data in Fig. 3(a), we compute the Spearman correlation coefficient and obtain $\rho = -0.202$, which confirms a mild negative correlation between the variables. This observation is important since it suggests that finetuning VGGT in the target environment might further top-off the performance of the geometric VLA.

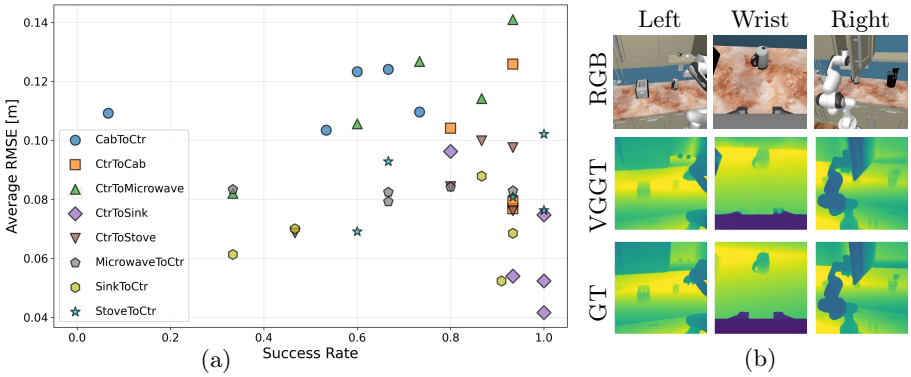


Fig. 3: (a) VGGT depth error vs. manipulation success rate in RoboCasa. (b) Sample RGB images and predicted vs. ground-truth depth, for the left, wrist, right cameras.

6 Limitations

Despite our best effort to provide a rigorous evaluation of the impact of geometric foundation models on VLAs, this work is not without limitations. First of all, the results are limited to our choice of VLA (GR00T-N1.5) and GFM (VGGT). There are many other VLAs and GFMs, and despite the presence of architectural similarities, we cannot generalize our conclusions beyond the combination we tested. Second, most of the results are on simulation benchmarks (except our analysis on the Unitree G1). While this is common practice in the field, it might not fully reflect real performance (indeed we observed that models using GFMs show stronger performance on real data). This issue is compounded by the fact that some of the simulation benchmarks are saturated, hence limiting the margin of improvement that can be shown. Third, while we tried to follow design choices used in related work, we did not ablate the impact of other choices (e.g., which layer to apply Spatial Forcing to, or alternative fusion strategies not relying on cross-attention); the scope of our experimental analysis is already very broad,

with some of the training experiments taking close to a week to complete, and exploring the entire design space is simply impractical. Finally, while we believe that adding statistical significance (and avoiding tuning random seeds) is crucial to draw rigorous conclusions, the corresponding p values are not binary, and still leave a margin for errors; we acknowledge this uncertainty in our evaluation.

7 Conclusions

We provide an analysis of the impact of injecting geometric information via Geometric Foundation Models (GFMs) into modern VLAs. First, we introduced three different strategies to inject geometric knowledge in VLAs, providing a unifying lens over the recent literature. Second, we quantified the gap in geometric understanding between a VLA (GR00T-N1.5) and a GFM (VGGT) using linear probing, and show such a gap can be filled by architectures leveraging GFMs. Third, we discussed factors impacting performance, including both architectural choices (i.e., choice of fusion strategy) as well as non-architectural choices (including the size of the training data, the number of cameras, and the quality of the VGGT reconstruction). As a result, this work provides non-trivial insights into the performance of novel VLAs integrating geometric foundation models, that we believe can trigger new and impactful research in this area.

References

1. Abouzeid, A., Mansour, M., Sun, Z., Song, D.: GeoAware-VLA: Implicit geometry aware vision-language-action model. arXiv preprint arXiv:2509.14117 (2025)
2. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al.: Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691 (2022)
3. Bjorck, J., Blukis, V., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L.J., Fang, Y., Fox, D., et al.: GR00T-N1.5: An improved open foundation model for generalist humanoid robots. `https://research.nvidia.com/labs/gear/gr00t-n1_5/` (2025)
4. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: GR00T-N1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
5. Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
6. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A Vision-Language-Action flow model for general robot control. In: Robotics: Science and Systems (RSS) (2025)
7. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817 (2022)
8. Cai, W., Ponomarenko, I., Yuan, J., Li, X., Yang, W., Dong, H., Zhao, B.: Spatialbot: Precise spatial understanding with vision language models. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 9490–9498. IEEE (2025)
9. Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F.: SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14455–14465 (June 2024)
10. Chen, S., Garcia, R., Laptev, I., Schmid, C.: Sugar: Pre-training 3d visual representations for robotics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18049–18060 (2024)
11. Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. In: Robotics: Science and Systems (RSS) (2023)
12. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
13. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: PaLM-E: an embodied multimodal language model. In: Proceedings of the 40th International Conference on Machine Learning. pp. 8469–8488 (2023)
14. Fang, X., Gao, J., Wang, Z., Chen, Z., Ren, X., Lyu, J., Ren, Q., Yang, Z., Yang, X., Yan, Y., Lyu, C.: Dens3R: A foundation model for 3d geometry prediction. arXiv preprint arXiv:2507.16290 (2025)
15. Figure AI: Helix: A vision-language-action model for generalist humanoid control (February 20 2025), `https://www.figure.ai/news/helix`, accessed: 2026-01-21

16. Ge, S., Zhang, Y., Xie, S., Zhang, W., Zhou, M., Wang, Z.: VGGT-DP: Generalizable robot control via vision foundation models. arXiv preprint arXiv:2509.18778 (2025)
17. Goyal, A., Blukis, V., Xu, J., Guo, Y., Chao, Y.W., Fox, D.: RVT-2: Learning precise manipulation from few demonstrations. arXiv preprint arXiv:2406.08545 (2024)
18. Han, B., Kim, J., Jang, J.: A dual process VLA: Efficient robotic manipulation leveraging VLM. arXiv preprint arXiv:2410.15549 (2024)
19. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
20. Hu, W., Lin, J., Long, Y., Ran, Y., Jiang, L., Wang, Y., Zhu, C., Xu, R., Wang, T., Pang, J.: G^2VLM: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2026)
21. Hu, Y., Cheng, C., Yu, S., Guo, X., Wang, H.: VGGT4D: Mining motion cues in visual geometry transformers for 4d scene reconstruction. arXiv preprint arXiv:2511.19971 (2025)
22. Hu, Y., Guo, Y., Wang, P., Chen, X., Wang, Y.J., Zhang, J., Sreenath, K., Lu, C., Chen, J.: Video prediction policy: A generalist robot policy with predictive visual representations. arXiv preprint arXiv:2412.14803 (2024)
23. Jia, Y., Liu, J., Chen, S., Gu, C., Wang, Z., Luo, L., Lee, L., Wang, P., Wang, Z., Zhang, R., et al.: Lift3d foundation policy: Lifting 2d large-scale pre-trained models for robust 3d robotic manipulation. arXiv preprint arXiv:2411.18623 (2024)
24. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Rota Bulò, S., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)
25. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: OpenVLA: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
26. Kim, T., Lee, J., Koo, M., Kim, D., Lee, K., Kim, C., Seo, Y., Shin, J.: Contrastive representation regularization for vision-language-action models. arXiv preprint arXiv:2510.01711 (2025)
27. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with MASt3R. In: European Conf. on Computer Vision (ECCV). vol. 15130, pp. 71–91 (2024)
28. Li, C., Wen, J., Peng, Y., Peng, Y., Feng, F., Zhu, Y.: Pointvla: Injecting the 3d world into vision-language-action models. arXiv preprint arXiv:2503.07511 (2025)
29. Li, F., Song, W., Zhao, H., Wang, J., Ding, P., Wang, D., Zeng, L., Li, H.: Spatial forcing: Implicit spatial representation alignment for vision-language-action model. arXiv preprint arXiv:2510.12276 (2025)
30. Lin, T., Li, G., Zhong, Y., Zou, Y., Du, Y., Liu, J., Gu, E., Zhao, B.: Evo-0: Vision-language-action model with implicit spatial understanding. arXiv preprint arXiv:2507.00416 (2025)
31. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. arXiv preprint arXiv:2306.03310 (2023)

32. Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., Zhu, Y.: Robocasa: Large-scale simulation of everyday tasks for generalist robots. In: Robotics: Science and Systems (RSS) (2024)
33. Ni, Z., He, Y., Qian, L., Mao, J., Fu, F., Sui, W., Su, H., Peng, J., Wang, Z., He, B.: Vo-dp: Semantic-geometric adaptive diffusion policy for vision-only robotic manipulation. arXiv preprint arXiv:2510.15530 (2025)
34. Peng, H., Li, H., Dai, Y., Lan, Y., Luo, Y., Qi, T., Zhang, Z., Zhan, Y., Zhang, J., Xu, W., Liu, Z.: Omnivggt: Omni-modality driven visual geometry grounded transformer. arXiv preprint arXiv:2511.10560 (2025)
35. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. arXiv preprint arXiv:2501.09747 (2025)
36. Physical Intelligence: Physical intelligence (π) (2026), `https://www.pi.website/`, accessed: 2026-01-21
37. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al.: SpatialVLA: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830 (2025)
38. Rojas, S., Armando, M., Ghamen, B., Weinzaepfel, P., Leroy, V., Rogez, G.: HAMSt3R: Human-aware multi-view stereo 3d reconstruction. arXiv preprint arXiv:2508.16433 (2025)
39. Steiner, R., Millane, A., Tingdahl, D., Volk, C., Ramasamy, V., Yao, X., Du, P., Pouya, S., Sheng, S.: mindmap: Spatial memory in deep feature maps for 3d action policies. arXiv preprint arXiv:2509.20297 (2025)
40. Tesla, Inc.: Ai & robotics (2026), `https://www.tesla.com/en_eu/AI`, accessed: 2026-01-21
41. Wang, H., Agapito, L.: AMB3R: Accurate feed-forward metric-scale 3d reconstruction with backend. arXiv preprint arXiv:2511.20343 (2025)
42. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novony, D.: VGGT: Visual geometry grounded transformer. arXiv preprint arXiv:2503.11651 (2025)
43. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3d vision made easy. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 20697–20709 (2024)
44. Wofk, D., Ma, F., Yang, T.J., Karaman, S., Sze, V.: FastDepth: Fast Monocular Depth Estimation on Embedded Systems. In: IEEE International Conference on Robotics and Automation (ICRA) (2019)
45. Wu, D., Liu, F., Hung, Y.H., Duan, Y.: Spatial-MLLM: Boosting MLLM capabilities in visual-based spatial intelligence. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
46. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. arXiv: 2406.09414 (2024)
47. Ze, Y., Zhang, G., Zhang, K., Hu, C., Wang, M., Xu, H.: 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. arXiv preprint arXiv:2403.03954 (2024)
48. Zhang, Z., Li, H., Dai, Y., Zhu, Z., Zhou, L., Liu, C., Wang, D., Tay, F.E.H., Chen, S., Liu, Z., Liu, Y., Li, X., Zhou, P.: From spatial to actions: Grounding vision-language-action model in spatial foundation priors. arXiv preprint arXiv:2510.17439 (2025)
49. Zhen, H., Qiu, X., Chen, P., Yang, J., Yan, X., Du, Y., Hong, Y., Gan, C.: 3D-VLA: A 3d vision-language-action generative world model. arXiv preprint arXiv:2403.09631 (2024)

50. Zheng, D., Huang, S., Li, Y., Wang, L.: Learning from videos for 3D world: Enhancing MLLMs with 3D vision geometry priors. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
51. Zhu, C., Wang, T., Zhang, W., Pang, J., Liu, X.: LLaVA-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. arXiv preprint arXiv:2409.18125 (2024)
52. Zust, L., Cabon, Y., Marrie, J., Antsfeld, L., Chidlovskii, B., Revaud, J., Csurka, G.: PanSt3R: Multi-view consistent panoptic segmentation. arXiv preprint arXiv:2506.21348 (2025)