Geometric Action Model for Robot Policy Learning

Jisang Han[1*], Seonghu Jeon[1*], Jaewoo Jung[1,2], René Zurbrügg[2,3], Honggyu An[1], Tifanny Portela[2,3], Marco Hutter[2], Marc Pollefeys[2], Seungryong Kim[1†], and Sunghwan Hong[2,3†]

[1]KAIST AI [2]ETH Zurich [3]ETH AI Center
[*]Equal contribution. [†]Corresponding author.

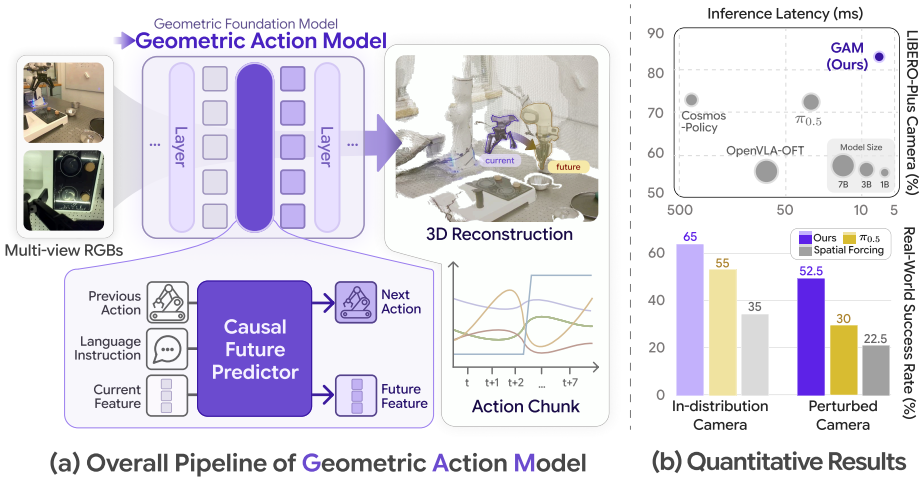


Fig. 1: GAM repurposes geometric foundation models into fast and robust robot policies. (a) GAM jointly predicts future 3D geometry and action chunks within a shared geometric backbone. (b) By leveraging explicit 3D geometric priors, GAM improves robustness and real-world performance while reducing latency and model size compared to existing baselines.

Abstract. Generalist robot policies must follow user instructions while reasoning about how objects, cameras, and robot actions interact in the 3D physical world. Recent vision-language-action models (VLAs) and video world-action models (WAMs) inherit strong semantic or temporal priors from large-scale foundation models, but they still operate primarily on 2D image frames or 2D-derived latent spaces, leaving implicit the 3D geometry required for contact-rich manipulation. We propose the **Geometric Action Model (GAM)**, a language-conditioned manipulation policy that directly repurposes a pretrained geometric foundation model (GFM) as a shared substrate for perception, temporal prediction, and action decoding. GAM splits the GFM at an intermediate layer: the shallow layers serve as an observation encoder, and a causal future predictor inserted at the split layer forecasts future latent tokens conditioned on language, proprioception, and action history. The predicted

future tokens are then routed through the remaining GFM blocks for feature propagation and decoding, allowing a single backbone to produce both future geometry and actions. This design equips the GFM with language-conditioned temporal world modeling through minimal architectural modification while preserving its rich geometric priors. Across a broad suite of simulation and real-robot manipulation benchmarks, GAM is more accurate, more robust, faster, and lighter than current foundation-model-scale baselines.

Keywords: Robot Policy Learning · Geometric Foundation Models · World Models · Visuomotor Control

1 Introduction

A long-standing goal in robotics is to build generalist manipulation policies that can follow natural-language instructions and manipulate arbitrary objects across diverse scenes [6, 19–21]. To achieve this, a general manipulation model must not only recognize objects and parse instructions, but also reason about how the physical world will evolve under its own actions. This requires a unified understanding of language, visual appearance, scene geometry, robot state, and physical dynamics.

Recent progress has therefore increasingly relied on large-scale foundation models as pretrained substrates for robot policies. Vision-language-action models (VLAs) build on vision-language models whose representations are aligned with natural language, and learn to map visual and linguistic tokens to robot actions [6,17,19,21,31,55]. Video world-action models (WAMs) instead leverage pretrained video generation models, using their world prediction priors to jointly model future frames and actions [20, 30]. While these approaches have shown impressive language-conditioned manipulation ability, they are fundamentally in 2D: 3D cues such as depth, scale, and occlusion are left implicit in monocular cues that the action decoder must disentangle on its own, leading to limited generalization across environment changes, especially in robot initial state and camera viewpoint [13, 46].

To overcome this limitation, recent work incorporates 3D geometric information into robot policies. One line learns policies directly on explicit 3D observations such as raw point clouds [15, 52], demonstrating the value of geometry for generalization but typically requiring task-specific encoders trained from scratch. With the emergence of Geometric Foundation Models (GFMs) [23, 42, 43], some works transfer pretrained geometric priors into VLA policies, either distilling selected GFM features into the VLA backbone through representation alignment [22, 38, 50] or attaching a lightweight action head on top of a GFM's final features [14, 32]. These improve spatial awareness, but use the GFM only as a static feature extractor: its multi-layer geometric structure is never repurposed as the policy's own temporal and action-generating substrate.

In this work, we propose **Geometric Action Model (GAM)**, which directly repurposes a GFM as a manipulation policy by using it as a shared medium

for perception, future-state prediction, and action decoding. We show that by jointly predicting future action and geometry, geometric world dynamics can be inherently incorporated into robot policies. This design connects explicit geometric priors with latent temporal world modeling for embodied control.

Specifically, we split the pretrained GFM at an intermediate layer: the shallow layers serve as an observation encoder, while the remaining layers serve as a decoder block. Given the current visual observation, the observation encoder extracts spatially meaningful scene representations. To model how the world evolves over time, we insert a causal transformer at the intermediate layer that predicts future feature representations. This predictor is conditioned on task language, proprioception, and action history by introducing them as additional tokens at each timestep. The predicted future-state tokens are then processed by the remaining GFM decoder together with an action token, allowing the backbone to produce both future geometry and robot actions. An intuitive comparison between existing paradigms and our proposed framework is illustrated in Figure 1.

Across diverse simulation and real-world benchmarks, GAM matches or exceeds the success rate of current foundation-model-scale baselines such as VLAs and WAMs while using substantially fewer trainable parameters and substantially faster inference **(55× faster)**, and shows improved generalization to unseen scenarios [13, 24, 28]. GAM especially achieves outstanding performance in camera perturbation settings **(↑9.7%p)**, which requires geometric understanding priors.

Our contributions are as follows:

- We introduce **Geometric Action Model (GAM)**, a manipulation policy that combines temporal world modeling, latent feature-space prediction, and a geometric foundation-model substrate in a single shared-backbone architecture.
- We show that action and geometry can be predicted in a shared token space: a single autoregressive sequence and a single backbone forward pass produce both action tokens and future-scene tokens, decoded by a lightweight action regression head and a depth head.
- We demonstrate that across diverse simulation and real-world manipulation benchmarks, GAM is simultaneously more accurate, more robust, faster, and lighter than current foundation-model-scale alternatives.

2 Related Work

Vision-Language-Action Models.

Vision-language-action models (VLAs) adapt a pretrained vision-language model (VLM) into a robot policy. Early works [21, 55] autoregressively decode discrete action tokens from a finetuned VLM. This paradigm is extended through parallel decoding [19], flow-matching action experts [6, 17, 36], diffusion-based action heads [25], frequency-space action tokenizations [31], and compact open-source VLMs [4, 16]. This line of work extends large generalist foundation models

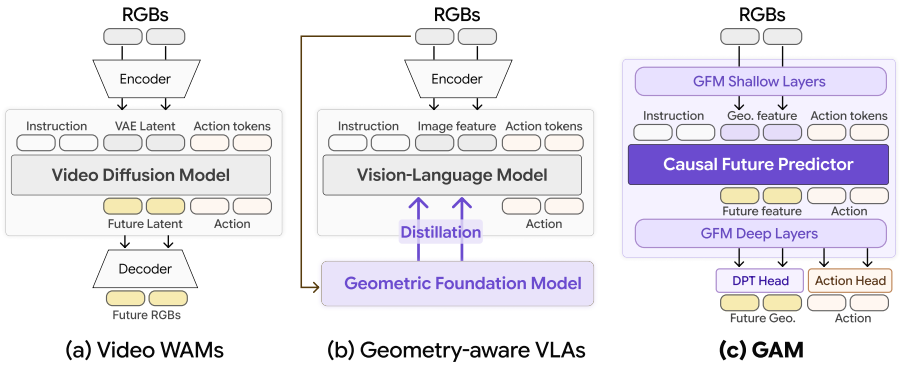


Fig. 2: (a) Video WAMs [20, 30, 49] operate in 2D pixel space, predicting future latents and actions via video diffusion. **(b) Geometry-aware VLAs** [22, 38] predict actions using a VLA with passive feature distillation from an external GFM. **(c) GAM (ours)** unifies perception, geometry prediction, and action decoding by inserting a geometric world model inside a single GFM.

for humanoid and embodied control [5, 9, 41, 48], to spatial-representation variants [33], and to refinements in training and post-training [7, 8, 40]. While these VLAs leverage strong open-vocabulary recognition to produce policies, they rely solely on 2D image priors and therefore lack 3D understanding. To address this, recent works [1, 22, 38] attempt to align intermediate VLA features with geometric foundation model. However, they do not fully exploit the geometric understanding of geometric foundation model backbone.

World Action Models for Robot Manipulation. World action models are trained to predict future states in order to learn a policy. One branch builds on large pretrained video generation models [2], fine-tuning them to jointly predict future frames and actions [20, 30, 49], arguing that the benefit of such co-training stems primarily from training-time supervision rather than test-time future imagination [31, 49]. A second branch keeps the visual backbone frozen and trains a separate temporal predictor in its feature space for planning via model-predictive control [3, 54], with related work using implicit future-latent alignment as an auxiliary training signal [53]. However, these video backbones encode only 2D image-space priors, which do not explicitly resolve depth, scale, or occlusion. GAM instead predicts in the latent space of a geometric foundation model that encodes rich 3D priors, while inheriting from both branches the paradigm of using future prediction as a training signal.

Geometric Foundation Models for Manipulation. Geometric foundation models (GFMs) [18, 23, 42, 44] infer dense 3D structures from multi-view images and have recently served as perceptual substrates for robot policies. Early approaches integrate GFMs with VLAs as frozen feature extractors via representation alignment [22,38], direct encoder replacement [1,32], or point-cloud fusion [39]. Moving beyond static extraction, recent concurrent works adopt GFMs for predictive control: Song et al. [37] utilize the GFM to jointly predict actions and *current*-frame 3D properties, while Xu et al. [47] employ the GFM with a

diffusion policy to co-denoise future action chunks and 3D latents. GAM departs from these paradigms in two ways: (1) action and future-scene predictions are jointly modeled within a single autoregressive token sequence rather than separated heads or diffusion processes, and (2) the GFM's deep blocks are explicitly repurposed to decode predicted *future* tokens rather than merely processing observed ones.

3 Preliminaries: Geometric Foundation Models

A geometric foundation model (GFM) such as VGGT [42] or DA3 [23] is a feed-forward transformer that maps one or more RGB images to dense 3D geometry. Given a sequence of V views $\mathcal{I} = \{I_v\}_{v=1}^{V}$ with $I_v \in \mathbb{R}^{3\times h\times w}$, a GFM produces per-pixel depth $D_v \in \mathbb{R}^{h\times w}$ or 3D point maps $P_v \in \mathbb{R}^{3\times h\times w}$ in a shared world frame, and per-view camera intrinsics and extrinsics $(K_v, \xi_v) \in \mathbb{R}^{3\times 3} \times SE(3)$ via auxiliary heads. Specifically, each view I_v is partitioned into P non-overlapping patches of size $p \times p$ and projected by a patch embedding into a per-view token sequence:

$$\mathbf{z}_v^{(0)} = \left[\mathbf{c}_v,\ \mathbf{x}_v^1, \ldots, \mathbf{x}_v^P\right] \in \mathbb{R}^{(1+P)\times d}, \tag{1}$$

where $\mathbf{c}_v$ is a per-view camera token, $\{\mathbf{x}_v^j\}_{j=1}^{P}$ are patch tokens, and d is the hidden dimension. The full input sequence concatenated across views is $\mathbf{Z}^{(0)} = [\mathbf{z}_1^{(0)}, \ldots, \mathbf{z}_V^{(0)}] \in \mathbb{R}^{V(1+P)\times d}$.

These tokens are processed by a stack of M transformer blocks $\{f^{(m)}\}_{m=1}^{M}$ employing one of two attention modes. *Frame-wise attention* $f_{\text{frame}}^{(m)}$ operates within each view tokens independently, attending over the $(1+P)$ tokens of a single image. *Global attention* $f_{\text{global}}^{(m)}$ operates jointly over all $V(1+P)$ tokens, fusing information across viewpoints. The hidden state at the m-th transformer block evolves as follows:

$$\mathbf{Z}^{(m)} = f^{(m)}\big(\mathbf{Z}^{(m-1)}\big), \quad f^{(m)} \in \{f_{\text{frame}}^{(m)}, f_{\text{global}}^{(m)}\}. \tag{2}$$

After the transformer forward pass, dense geometry is decoded from multiple intermediate hidden states $\mathbf{Z}^{(m^*)}$, where m^* is one of the selected layers in $\mathcal{S} = \{m_1, m_2, m_3, m_4\}$. The extracted multi-layer hidden states are then fed into the DPT [35] head to estimate per-pixel geometry. Following the original DPT design [35], the head fuses multi-scale transformer features and upsamples them into dense pixel-wise predictions.

4 GAM: Geometric Action Model

Problem Formulation. We consider language-conditioned robot manipulation. At each timestep t, the robot receives a multi-view RGB observation $o_t = \{I_{v,t}\}_{v=1}^{V}$ from V fixed cameras, a proprioceptive state $s_t \in \mathbb{R}^{d_s}$ describing the robot's joint configuration and end-effector pose, and a natural-language task instruction ℓ that is held constant throughout an episode. The policy π_θ must

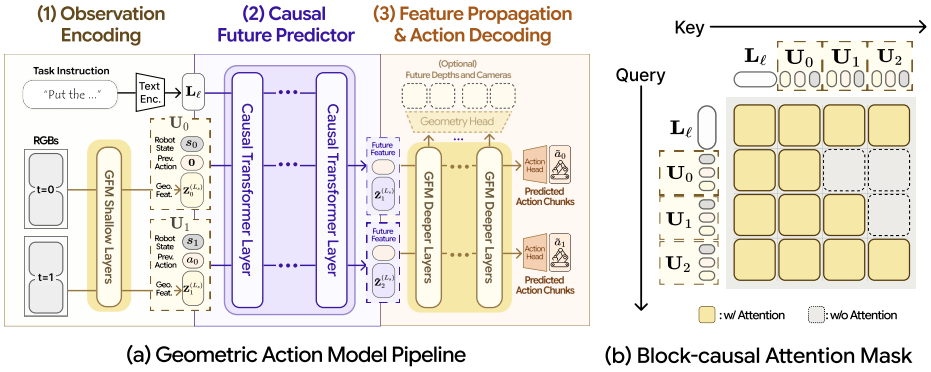


Fig. 3: Main architecture of GAM.

produce an action chunk $\hat{a}_t \in \mathbb{R}^{C \times d_a}$ of length C, encoding the next C delta-pose or joint commands to be executed open-loop before the next observation is acquired. We learn a policy:

$$\pi_\theta : \big(\{o_{t-H+1}, \ldots, o_t\}, \{s_{t-H+1}, \ldots, s_t\}, \{a_{t-H}, \ldots, a_{t-1}\}, \ell\big) \mapsto \hat{a}_t \qquad (3)$$

from a dataset of N expert demonstrations $\mathcal{D} = \{(\tau_i, \ell_i)\}_{i=1}^{N}$, where each trajectory $\tau_i = (o_t, s_t, a_t)_{t=1}^{T_i}$ pairs a sequence of observations, states, and executed action chunks with a fixed instruction ℓ_i. The policy conditions on a context window of H recent timesteps.

Overview. In the following sections, we explain how we transform a pretrained GFM into a language-conditioned world-action model. Our key idea is to split the GFM into two parts and insert a causal temporal predictor between them. This design lets GAM formulate future prediction directly inside the GFM latent space, enabling all predictive computation to be performed in the GFM's geometric representation space.

Concretely, our framework operates in three sequential stages inside the GFM. First, the **observation encoder** (§4.1) repurposes the shallow layers of the GFM to extract latent geometric features from multi-view observations. Next, the **causal future predictor** (§4.2) operates at the split layer, where it combines these geometric features with language, proprioception, and action history to predict future latent tokens. Finally, during **feature propagation and decoding** (§4.3), the predicted future tokens are routed through the remaining deep GFM blocks to simultaneously decode future geometry and the final action chunk $\hat{a}_t$. Figure 3 (a) shows the overall architecture of our model.

4.1 Observation Encoder

We first reuse the shallow layers of the pretrained GFM as the observation encoder. Let L_s denote the split layer where the causal future predictor is inserted. The original GFM transformer stack is then decomposed into an encoder and a decoder:

$$E_{\le L_s} = f^{(L_s)} \circ \cdots \circ f^{(1)}, \qquad D_{>L_s} = f^{(M)} \circ \cdots \circ f^{(L_s+1)}. \qquad (4)$$

Here, the choice of L_s is important because L_s must be deep enough to extract sufficiently rich visual features from the raw observations, yet shallower than the earliest layer used in the DPT head $L_s < m_1$, so that predicted future states can be decoded into future geometries by the DPT heads.

After defining this split layer L_s, for each timestep t' in the context window, we tokenize the multi-view RGB observation $o_{t'} = \{I_{v,t'}\}_{v=1}^{V}$ using the original GFM patch embedding. This produces the initial multi-view token sequence:

$$\mathbf{Z}_{t'}^{(0)} = \left[\mathbf{z}_{1,t'}^{(0)}, \ldots, \mathbf{z}_{V,t'}^{(0)}\right] \in \mathbb{R}^{V(1+P)\times d}, \tag{5}$$

where each view contributes one camera token and P patch tokens. The observation encoder maps these tokens to the split-layer representation $\mathbf{Z}_{t'}^{(L_s)}$. By applying this encoding independently to each timestep in the context window, the output of this stage is a sequence of per-timestep geometric latent states $\{\mathbf{Z}_{t-H+1}^{(L_s)}, \ldots, \mathbf{Z}_{t}^{(L_s)}\}$.

4.2 Causal Future Predictor

After the observation encoder, GAM performs temporal prediction directly at the split layer L_s, forecasting the next latent geometric state from current and past observations while conditioning on the task instruction, proprioception, and action history. To this end, we insert a causal future predictor g_ϕ between the shallow encoder $E_{\le L_s}$ and the deep decoder $D_{>L_s}$. For each timestep t' in the context window, the encoder provides latent tokens $\mathbf{Z}_{t'}^{(L_s)}$, and we embed the proprioceptive state $s_{t'}$ and previous action $a_{t'-1}$ as tokens:

$$\mathbf{p}_{t'} = \psi_s(s_{t'}), \qquad \mathbf{q}_{t'} = \psi_a(a_{t'-1}), \tag{6}$$

with ψ_s, ψ_a lightweight projection layers, and the instruction ℓ into language tokens $\mathbf{L}_\ell$ with a pretrained text encoder. We then form a per-timestep token block by concatenating the encoded GFM tokens with the proprioception and action-history tokens $\mathbf{U}_{t'} = [\mathbf{p}_{t'}; \mathbf{q}_{t'}; \mathbf{Z}_{t'}^{(L_s)}]$ and prepend the language tokens to obtain $\mathbf{X} = [\mathbf{L}_\ell; \mathbf{U}_{t'-H+1}; \ldots; \mathbf{U}_{t'}]$.

Block-causal self-attention [6, 12] processes the combined sequence $\mathbf{X}$. Let i and j index a query and a key, let $\mathcal{L}$ denote the language prefix, and let $b(\cdot)$ denote a robot token's timestep block. The additive mask is

$$M_{ij} = \begin{cases} 0, & j \in \mathcal{L} \text{ or } \big(i, j \notin \mathcal{L} \ \wedge \ b(j) \le b(i)\big), \\ -\infty, & \text{otherwise.} \end{cases} \tag{7}$$

Language queries therefore see only the language prefix, whereas robot queries see all language tokens and the same or earlier robot blocks. At the final layer of the predictor g_ϕ, we read off the predictions from their respective sequence slots. Specifically, the hidden states corresponding to the geometry slots forecast the latent geometric tokens of the future frame, denoted as $\tilde{\mathbf{Z}}_{t'+1}^{(L_s)}$. Concurrently, the hidden state of the designated previous-action slot is projected to produce a

predicted next action token $\tilde{\mathbf{a}}_{t'} \in \mathbb{R}^d$, in direct analogy to next-token prediction in a causal language model. By jointly forecasting action and geometric latents in this layer, we ensure that action tightly interacts with spatial representations.

This design of introducing a causal transformer predictor g_ϕ allows the pretrained GFM to acquire language-conditioned temporal world modeling with *minimal* architectural modification. Only the inserted g_ϕ needs to learn how to fuse language, proprioception, and action history with GFM latent features. The resulting predictions, $\widetilde{\mathbf{Z}}^{(L_s)}_{t'+1}$ and $\tilde{\mathbf{a}}_{t'}$, are then passed to the remaining GFM blocks for joint geometry and action decoding.

4.3 Feature Propagation and Action Decoding

Following the causal future predictor, the single action token $\tilde{\mathbf{a}}_{t'}$ is replicated V times to form a set of per-view action tokens $\{\tilde{\mathbf{a}}_{v,t'}\}_{v=1}^{V}$, where $\tilde{\mathbf{a}}_{v,t'} = \tilde{\mathbf{a}}_{t'}$. Concatenated with the geometry tokens, they are fed through the remaining GFM blocks $D_{>L_s}$. We perform this *feature propagation* by appending each view's corresponding action token $\tilde{\mathbf{a}}_{v,t'}$ directly to its geometry token sequence for each timestep:

$$\tilde{\mathbf{Z}}^{(M)}_{t'+1} = \big(f^{(M)} \circ \cdots \circ f^{(L_s+1)}\big)\Big(\Big[\big[\tilde{\mathbf{Z}}^{(L_s)}_{1,t'+1};\, \tilde{\mathbf{a}}_{1,t'}\big], \ldots, \big[\tilde{\mathbf{Z}}^{(L_s)}_{V,t'+1};\, \tilde{\mathbf{a}}_{V,t'}\big]\Big]\Big). \tag{8}$$

To prevent future leakage, we extend the predictor's causal mask strategy to the GFM's remaining global attention layers ($f^{(m)}_{\text{global}}$).

Finally, the propagated features are decoded by two heads. The lightweight action head h_{act} aggregates action tokens over the context window to regress the executable action chunk $\hat{a}_{t'}$, while the original GFM depth head h_{depth} decodes geometry tokens into action-aligned future depth maps. The GFM's deep blocks, originally pretrained to decode shallow features into 3D geometry, are thus repurposed here as the decoder of the world model's predicted future.

4.4 Training and Inference

The policy is trained end-to-end by minimizing an action loss and a future-prediction loss:

$$\mathcal{L}_{\text{future}} = \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}, \qquad \mathcal{L}_{\text{total}} = \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \mathcal{L}_{\text{future}}, \tag{9}$$

where the λ factors balance each term and $\mathcal{H} = \{t-H+1, \ldots, t\}$ is the context window. The *action* loss $\mathcal{L}_{\text{act}}$ is an ℓ_1 regression between the decoded action chunk $\hat{a}_{t'}$ and the expert action $a_{t'}$ over all $t' \in \mathcal{H}$. The future loss $\mathcal{L}_{\text{future}}$ combines latent future-feature alignment with future-depth supervision. Specifically, the *future-feature* loss $\mathcal{L}_{\text{feat}}$ anchors the predictor g_ϕ to temporal geometric transitions by aligning predicted future tokens $\tilde{\mathbf{Z}}^{(L_s)}_{t'+1}$ with the actual next frame $\mathbf{Z}^{(L_s)}_{t'+1}$ extracted from the frozen GFM:

$$\mathcal{L}_{\text{feat}} = \sum_{t' \in \mathcal{H}} \left\| \tilde{\mathbf{Z}}^{(L_s)}_{t'+1} - \mathbf{Z}^{(L_s)}_{t'+1} \right\|_1. \tag{10}$$

The *future-depth* loss $\mathcal{L}_{\text{depth}}$ grounds the predicted future in valid 3D structure by supervising the decoded depth $\tilde{D}_{t'+1} = h_{\text{depth}}(\tilde{\mathbf{Z}}_{t'+1}^{(m^*)})$ using depth head h_{depth} against ground-truth future depth $D_{t'+1}$. Following the Depth Anything 3 training objective [23], we use its scale-invariant depth and gradient-matching penalties for this supervision.

During inference, key-value caching maintains the historical context online. Each step therefore processes only the new observation o_t and previous action a_{t-1} in a single feed-forward pass.

5 Experiments

5.1 Implementation Details

We use DA3-Giant [23] fine-tuned on Track4World [26] as the backbone. We insert a 12-layer causal predictor with width $d_g = 1024$ at layer $L_s = 12$, where alternating attention begins. For the task instruction, we extract language tokens using a frozen T5 encoder [34]. The policy predicts $C = 8$ step action chunks in a $d_a = 7$ end-effector action space from $d_s = 7$ proprioceptive states. We optimize with AdamW using a constant learning rate. In both training stages, the GFM blocks through L_s and the DPT depth head remain frozen, while the causal predictor, action head, and GFM blocks after L_s are updated. We set $\lambda_{\text{act}} = 3$ and define $\mathcal{L}_{\text{future}}$ using $\lambda_{\text{feat}} = 1$ and $\lambda_{\text{depth}} = 3$.

Two-stage Training. GAM is first pre-trained with a context horizon of $H = 4$ on 784K single-arm robot trajectories from RoboCasa365 [29], MimicGen [27], and OpenX-Embodiment [10]. All pre-training sources are converted to a common observation format and a 7-DoF end-effector action space while retaining their original task instructions. We include only OXE subsets compatible with this control interface and the manipulation-task subset of RoboCasa365. Future-depth supervision is source-dependent: OXE uses teacher pseudo-depth from the pretrained GFM, whereas MimicGen and RoboCasa365 use re-rendered simulator depth. GAM is then post-trained on each downstream benchmark with $H = 1$. Simulation benchmarks use ground-truth future depth, while real-world post-training uses pseudo-depth targets from the pretrained GFM because metric future-depth annotations are unavailable. Both stages optimize the same objective and update the same parameter set: pre-training learns transferable temporal geometry from multi-step contexts, whereas post-training adapts action decoding to each downstream data distribution using single-step inputs. At inference, GAM requires neither a target future observation nor depth supervision; it predicts and executes the eight-step action chunk open-loop from the current RGB observations, robot state, task instruction, and previous action in a single forward pass. Dataset sampling, optimization, and augmentation details are provided in the supplementary material.

Table 1: Evaluation results on LIBERO and LIBERO-Plus. Success rates are reported in % with absolute performance drops from LIBERO to LIBERO-Plus shown in parentheses. Color highlights denote the top three performing methods within each column: first, second, and third.

Method	Size	Orig.	Plus	Cam.	Robot	Lang.	Light	BG	Noise	Layout
VLAs										
$\pi_{0.5}$ [17]	3.3B	96.9	84.6 (↓12.3)	72.0	76.6	86.5	96.1	95.2	86.7	86.0
OpenVLA-OFT [19]	7B	97.1	69.6 (↓27.5)	56.4	31.9	79.5	88.7	93.3	75.8	74.3
RIPT-VLA [40]	7B	97.5	68.4 (↓29.1)	55.2	31.2	77.6	88.4	91.6	73.5	74.2
π_0 [44]	3.3B	91.3	69.3 (↓22.0)	61.0	40.8	63.7	89.3	84.1	80.1	75.9
π_0-FAST [31]	3.3B	85.5	61.6 (↓23.9)	65.1	21.6	61.0	73.2	73.3	74.4	68.8
UniVLA [7]	8.5B	95.2	42.9 (↓52.3)	1.8	46.2	69.5	69.0	81.0	21.2	31.9
NORA [16]	3B	87.9	39.0 (↓48.9)	2.2	37.0	65.1	45.7	58.6	12.8	62.1
OpenVLA [21]	7B	76.5	15.6 (↓60.9)	0.8	3.5	23.0	8.1	34.8	15.2	28.5
WAMs										
Cosmos-Policy [20]	2B	98.5	82.4 (↓16.1)	73.4	63.3	89.3	98.9	83.5	89.3	84.0
Fast-WAM [51]	6B	97.6	50.0 (↓47.5)	16.4	44.5	68.9	78.2	53.7	37.7	60.7
WorldVLA [8]	7B	79.1	25.0 (↓54.1)	0.1	27.9	41.6	43.7	17.1	11.0	38.0
Geometry-aware VLAs										
Spatial Forcing [22]	3.3B	94.0	25.7 (↓58.3)	0.1	0.3	26.8	66.0	45.9	0.1	59.8
ROCKET [38]	3.3B	95.3	47.5 (↓46.6)	30.9	75.6	29.3	69.2	47.0	25.4	62.0
GAM										
GAM (Ours)	1.4B	97.6	85.5 (↓12.1)	83.1	70.0	84.8	97.2	94.3	95.3	79.1

5.2 Experimental Setup

Simulation Benchmarks. We evaluate generalization across distinct axes using two simulation benchmarks. Specifically, we train our policy on **LIBERO** [24], a lifelong single-arm manipulation benchmark spanning diverse spatial layouts, object identities, and task goals. We then test zero-shot out-of-distribution robustness on **LIBERO-Plus** [13]. This benchmark introduces controlled environmental perturbations across dimensions such as camera viewpoint, lighting, and backgrounds. We report additional results in the supplementary material.

Real-Robot Setup. We train on four tasks (∼200 demonstrations each) using wrist-mounted ZED and external RealSense views (Figure 4), with GFM pseudo-depth targets. Each task has 10 ID and 10 OOD trials; OOD translates the external camera by 85 cm and rotates it by 45°. The supplementary material provides the full protocol.

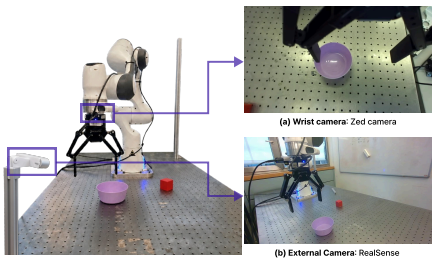


Fig. 4: Real-world camera setup.

Baselines. We compare GAM with **VLAs** [6,7,16,17,19,21,31,40] and **WAMs** [8,20,51]. We also include **geometry-aware VLAs** [22,38]. The real-robot baselines are $\pi_{0.5}$ [17] and Spatial Forcing [22]. We use a matched evaluation protocol and either re-evaluate available checkpoints or report published benchmark results.

5.3 Main Results

Simulation Results. Table 1 shows that GAM remains competitive on the saturated LIBERO benchmark and outperforms competing baselines on LIBERO-Plus, including a **9.7 percentage-point gain** under camera perturbation. Existing geometry-aware VLAs use GFM representations only partially, whereas GAM incorporates the GFM throughout its predictive pathway. The result supports end-to-end geometric integration for out-of-distribution control.
Real-world Results. Figure 5 shows that GAM outperforms both baselines in ID and OOD trials and remains robust under camera perturbation. The transfer from simulation to physical execution shows that integrating the GFM throughout policy training improves real-world robustness.

5.4 Ablation Study

Post-training Component Analysis. Table 2 summarizes the effects of pretraining, the future loss $\mathcal{L}_{\text{future}}$, and context horizon on the Object suite of LIBERO and LIBERO-Plus. The small marginal effect of $\mathcal{L}_{\text{future}}$ after pretraining should not be interpreted as evidence that future prediction is unnecessary. Pretraining supplies most of the temporal geometric dynamics: with $\mathcal{L}_{\text{future}}$ enabled, pretraining raises the success rates from 98.4/73.4 to 99.6/89.7 on LIBERO and LIBERO-Plus. When pretraining

Table 2: Future-loss and context ablation.

Pretrain	$\mathcal{L}_{\text{future}}$	H	Orig. SR (%)	Plus SR (%)
✓	✓	1	**99.6**	**89.7**
✓	✓	2	97.2	84.4
✓	✓	4	98.2	85.1
✓	✗	1	98.6	89.5
✗	✓	1	98.4	73.4
✗	✗	1	93.6	50.0

Table 3: Layer ablation.

Split layer L_s	Orig. (%)	Plus (%)
0	5.4	1.8
12	**99.6**	**70.1**
19	95.6	63.4
27	1.2	1.6
33	0.0	0.0
39	0.0	0.0

Table 4: Direct action-token ablation.

Variant	Orig.	Plus
Direct-action supervision	98.4	84.1
GAM (Ours)	**99.6**	**89.7**

Table 5: Matched and deployment latency.

Method	Size	Matched	Deploy.
OpenVLA-OFT [19]	7B	70.1 ms	–
$\pi_{0.5}$ [17]	3.3B	29.2 ms	–
Cosmos Policy [20]	2B	382.4 ms	–
GAM (Ours)	**1.4B**	**17.5 ms**	**6.9 ms**

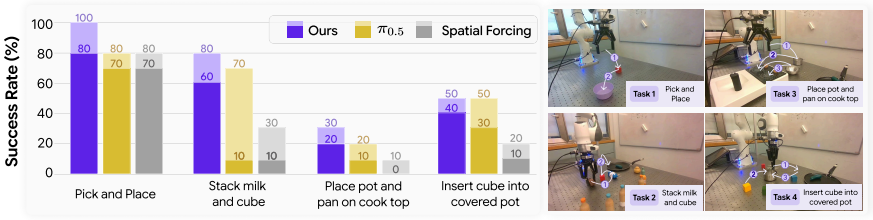


Fig. 5: Real-world robot tasks and results. Each task is evaluated under both in-distribution (light bars) and out-of-distribution (dark bars) camera settings. The four evaluated tasks are illustrated on the right.

is unavailable, $\mathcal{L}_{\text{future}}$ raises the success rates from 93.6/50.0 to 98.4/73.4, showing that future supervision can learn dynamics directly from the task data. After pretraining, removing $\mathcal{L}_{\text{future}}$ changes the success rates only from 99.6/89.7 to 98.6/89.5, so the loss mainly acts as an auxiliary signal during task adaptation. The horizon ablation shows that $H = 1$ is sufficient and more robust than longer histories, consistent with prior observations that extended context can introduce spurious correlations [11, 45].

Split Layer L_s Selection. Table 3 evaluates the depth of future predictor by shifting the split layer L_s and re-initializing the predictor. This split-layer analysis uses only $\mathcal{L}_{\text{feat}}$ because $\mathcal{L}_{\text{depth}}$ is not equally applicable to every split layer; the experiment therefore isolates where latent future prediction should be inserted. Our default choice of $L_s = 12$ achieves peak performance, validating it as the optimal seam between frame-wise and cross-view attention. While layer 19 remains competitive, inserting the predictor too early ($L_s = 0$) or late ($L_s \in \{27, 33, 39\}$) causes total performance collapse. This confirms that forecasted tokens require sufficient interaction through deep layers to properly integrate into the pretrained 3D geometric prior.

Action Decoding Path. Table 4 compares GAM with a variant that applies action supervision directly to the causal predictor's action token without propagating it through the remaining GFM blocks. On the Object suites of LIBERO and LIBERO-Plus, deep GFM decoding improves the success rates from 98.4/84.1 to 99.6/89.7. Together, the ablations show that pretraining, future supervision, and deep GFM decoding each improve generalization: pretraining supplies broad temporal dynamics, future supervision learns or preserves task-relevant dynamics, and the deep decoder refines action tokens through pretrained geometric representations.

5.5 Analysis

Inference Speed and Model Size. Table 5 separates matched-runtime measurements from GAM's optimized deployment setting. Under the same setting, GAM takes 17.5 ms, compared with 70.1 ms for OpenVLA-OFT and 382.4 ms for Cosmos Policy. GAM is therefore 21.9× faster than Cosmos Policy under

matched conditions. Enabling CUDA Graphs only for GAM's optimized deployment reduces its latency to **6.9 ms (≈145 Hz)**, corresponding to a maximum deployment speedup of 55.4× relative to the reported Cosmos Policy latency. Further measurement details are provided in the supplementary material. GAM combines this latency with a 1.4B-parameter model by using single-pass prediction instead of multi-step diffusion denoising.

Robustness to Viewpoint and Scene Variation. Figure 6 further breaks down the camera-perturbation results by difficulty level of LIBERO-Plus [13]. GAM achieves consistently higher success rates than all baselines at every level, and the advantage remains clear even under the strongest perturbations.

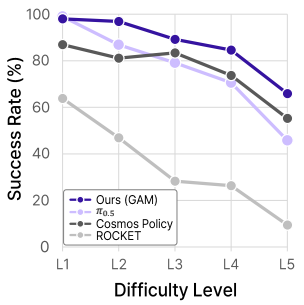
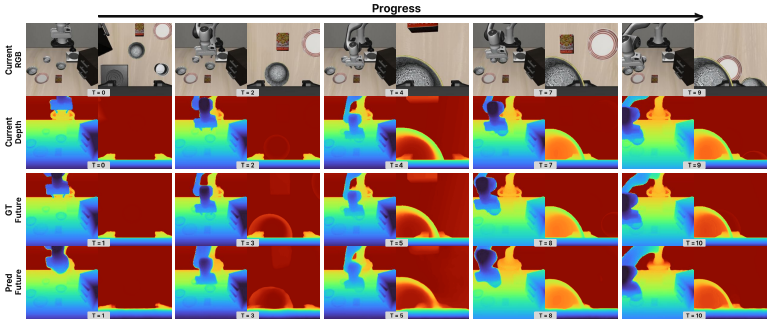


Fig. 6: Success rate vs. camera perturbation difficulty.

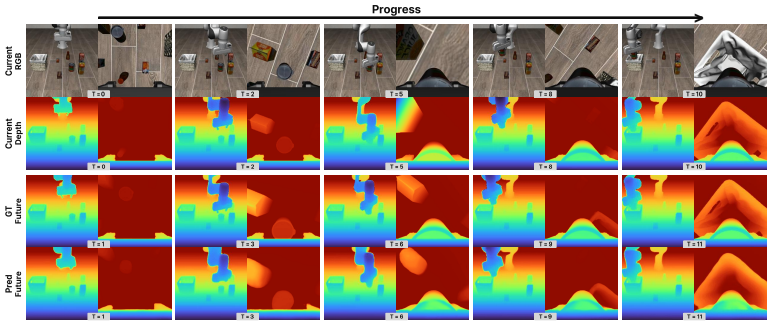
Qualitative Future-Depth Prediction. Figure 7 compares the current RGB and depth with the ground-truth and predicted future depth at successive stages of representative tasks from all four LIBERO suites. The predictions preserve the static scene layout and depth ordering while tracking changes around the manipulated object and target region, including objects approaching plates, baskets, and cabinets. The predicted changes follow the corresponding ground-truth future depth instead of merely reproducing the current depth, providing qualitative evidence that $\mathcal{L}_{\text{future}}$ learns temporally coherent geometric dynamics.

6 Conclusion and Limitation

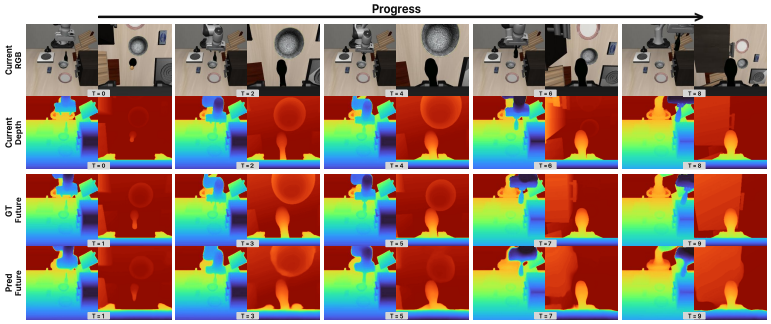
We introduced Geometric Action Model, which unifies geometry and action prediction with temporal world modeling inside a single shared GFM. By inserting a causal transformer between the GFM's shallow and deep layers, GAM autoregressively decodes actions and future geometries, resolving the spatial ambiguities of traditional foundation-model substrates. Across extensive simulation and real-world benchmarks, GAM achieves superior accuracy, faster inference, and strong out-of-distribution robustness to environmental perturbations. These results show that explicit geometric priors can complement latent world modeling, particularly when robot policies must remain robust to viewpoint changes. The framework also has limitations. Its language reasoning and commonsense capabilities are bounded by the frozen text encoder; integrating a large language model or an external reasoning module is a natural next step.



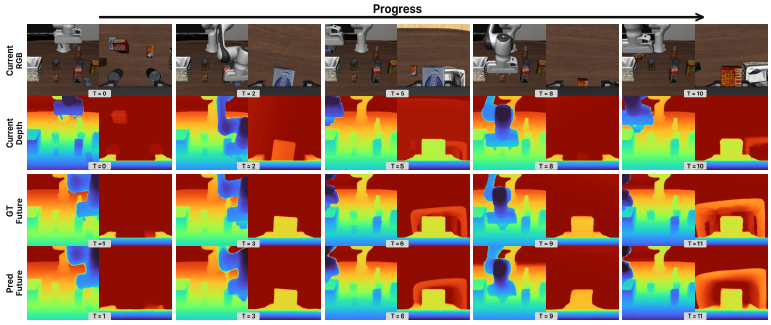
(a) LIBERO-Spatial: bowl from table center to plate.



(b) LIBERO-Object: tomato sauce to basket.



(c) LIBERO-Long: cream cheese and butter to basket.



(d) LIBERO-Goal: wine bottle on cabinet.

Fig. 7: Future depth predictions from GAM on representative LIBERO tasks.

References

1. Abouzeid, A., Mansour, M., Sun, Q., Sun, Z., Song, D.: Geoaware-vla: Implicit geometry aware vision-language-action model. arXiv preprint arXiv:2509.14117 (2025)
2. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
3. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: others. 2025. qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 **4**(5) (1)
5. 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)
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. arXiv preprint arXiv:2410.24164 (2024)
7. Bu, Q., Yang, Y., Cai, J., Gao, S., Ren, G., Yao, M., Luo, P., Li, H.: Univla: Learning to act anywhere with task-centric latent actions. arXiv preprint arXiv:2505.06111 (2025)
8. Cen, J., Yu, C., Yuan, H., Jiang, Y., Huang, S., Guo, J., Li, X., Song, Y., Luo, H., Wang, F., et al.: Worldvla: Towards autoregressive action world model. arXiv preprint arXiv:2506.21539 (2025)
9. Cheang, C., Chen, S., Cui, Z., Hu, Y., Huang, L., Kong, T., Li, H., Li, Y., Liu, Y., Ma, X., et al.: Gr-3 technical report. arXiv preprint arXiv:2507.15493 (2025)
10. Collaboration, O.E., O'Neill, A., Rehman, A., Gupta, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models. arXiv preprint arXiv:2310.08864 **1**(2) (2023)
11. De Haan, P., Jayaraman, D., Levine, S.: Causal confusion in imitation learning. Advances in neural information processing systems **32** (2019)
12. Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H.W.: Unified language model pre-training for natural language understanding and generation. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
13. Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., et al.: Libero-plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626 (2025)
14. 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)
15. Huang, W., Chao, Y.W., Mousavian, A., Liu, M.Y., Fox, D., Mo, K., Fei-Fei, L.: Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. arXiv preprint arXiv:2601.03782 (2026)
16. Hung, C.Y., Sun, Q., Hong, P., Zadeh, A., Li, C., Tan, U., Majumder, N., Poria, S., et al.: Nora: A small open-sourced generalist vision language action model for embodied tasks. arXiv preprint arXiv:2504.19854 (2025)

17. Intelligence, P., 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)
18. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)
19. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645 (2025)
20. Kim, M.J., Gao, Y., Lin, T.Y., Lin, Y.C., Ge, Y., Lam, G., Liang, P., Song, S., Liu, M.Y., Finn, C., et al.: Cosmos policy: Fine-tuning video models for visuomotor control and planning. arXiv preprint arXiv:2601.16163 (2026)
21. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
22. 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)
23. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
24. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
25. Liu, S., Wu, L., Li, B., Tan, H., Chen, H., Wang, Z., Xu, K., Su, H., Zhu, J.: Rdt-1b: a diffusion foundation model for bimanual manipulation. In: International Conference on Learning Representations. vol. 2025, pp. 29982–30009 (2025)
26. Lu, J., Xu, J., Hu, W., Zhu, R., Zhao, C., Yeung, S.K., Shan, Y., Liu, Y.: Track4world: Feedforward world-centric dense 3d tracking of all pixels. arXiv preprint arXiv:2603.02573 (2026)
27. Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., Fox, D.: Mimicgen: A data generation system for scalable robot learning using human demonstrations. arXiv preprint arXiv:2310.17596 (2023)
28. 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. arXiv preprint arXiv:2406.02523 (2024)
29. Nasiriany, S., Nasiriany, S., Maddukuri, A., Zhu, Y.: Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. arXiv preprint arXiv:2603.04356 (2026)
30. Pai, J., Achenbach, L., Montesinos, V., Forrai, B., Mees, O., Nava, E.: mimic-video: Video-action models for generalizable robot control beyond vlas. arXiv preprint arXiv:2512.15692 (2025)
31. 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)
32. Qian, Q., Zhao, G., Zhang, G., Wang, J., Xu, R., Gao, J., Zhao, D.: Gp3: A 3d geometry-aware policy with multi-view images for robotic manipulation. arXiv preprint arXiv:2509.15733 (2025)
33. 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)

34. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
35. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021)
36. Shi, L.X., Ichter, B., Equi, M., Ke, L., Pertsch, K., Vuong, Q., Tanner, J., Walling, A., Wang, H., Fusai, N., et al.: Hi robot: Open-ended instruction following with hierarchical vision-language-action models. arXiv preprint arXiv:2502.19417 (2025)
37. Song, Z., Li, Q., Zhou, J., Yuan, Z., Chen, T., Lin, L., Wang, G.: Robotic manipulation is vision-to-geometry mapping ($f(v) \rightarrow g$): Vision-geometry backbones over language and video models. arXiv preprint arXiv:2604.12908 (2026)
38. Sun, G., Du, T., Feng, K., Luo, C., Ding, X., Shen, Z., Wang, Z., He, Y., Li, A.: Rocket: Residual-oriented multi-layer alignment for spatially-aware vision-language-action models. arXiv preprint arXiv:2602.17951 (2026)
39. Sun, L., Xie, B., Liu, Y., Shi, H., Wang, T., Cao, J.: Geovla: Empowering 3d representations in vision-language-action models. arXiv preprint arXiv:2508.09071 (2025)
40. Tan, S., Dou, K., Zhao, Y., Krähenbühl, P.: Interactive post-training for vision-language-action models. arXiv preprint arXiv:2505.17016 (2025)
41. Team, G.R., Abdolmaleki, A., Abeyruwan, S., Ainslie, J., Alayrac, J.B., Arenas, M.G., Balakrishna, A., Batchelor, N., Bewley, A., Bingham, J., et al.: Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. arXiv preprint arXiv:2510.03342 (2025)
42. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
43. Wang, J., Chen, M., Zhang, S., Karaev, N., Schönberger, J., Labatut, P., Bojanowski, P., Novotny, D., Vedaldi, A., Rupprecht, C.: Vggt-ω. arXiv preprint arXiv:2605.15195 (2026)
44. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Scalable permutation-equivariant visual geometry learning. arXiv e-prints pp. arXiv–2507 (2025)
45. Wen, C., Lin, J., Darrell, T., Jayaraman, D., Gao, Y.: Fighting copycat agents in behavioral cloning from observation histories. Advances in Neural Information Processing Systems **33**, 2564–2575 (2020)
46. Wilcox, A., Ghanem, M., Moghani, M., Barroso, P., Joffe, B., Garg, A.: Adapt3r: Adaptive 3d scene representation for domain transfer in imitation learning. arXiv preprint arXiv:2503.04877 (2025)
47. Xu, C., Li, H., Cheng, S., Hu, J., Fan, H., Feng, Z., Liu, S.: Action-geometry prediction with 3d geometric prior for bimanual manipulation. arXiv preprint arXiv:2602.23814 (2026)
48. Yang, J., Tan, R., Wu, Q., Zheng, R., Peng, B., Liang, Y., Gu, Y., Cai, M., Ye, S., Jang, J., et al.: Magma: A foundation model for multimodal ai agents. In: Proceedings of the computer vision and pattern recognition conference. pp. 14203–14214 (2025)
49. Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y.L., Zhu, C., Xiang, J., et al.: World action models are zero-shot policies. arXiv preprint arXiv:2602.15922 (2026)

50. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940 (2024)
51. Yuan, T., Dong, Z., Liu, Y., Zhao, H.: Fast-wam: Do world action models need test-time future imagination? arXiv preprint arXiv:2603.16666 (2026)
52. 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)
53. Zheng, R., Wang, J., Reed, S., Bjorck, J., Fang, Y., Hu, F., Jang, J., Kundalia, K., Lin, Z., Magne, L., et al.: Flare: Robot learning with implicit world modeling. arXiv preprint arXiv:2505.15659 (2025)
54. Zhou, G., Pan, H., LeCun, Y., Pinto, L.: Dino-wm: World models on pre-trained visual features enable zero-shot planning. arXiv preprint arXiv:2411.04983 (2024)
55. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)