GeoWAM: Geometry Gradient Transfer for World Action Models

Anonymous authors
Paper under double-blind review

Abstract

World action models learn robot policies through video prediction and action generation, but predicting visual appearance does not explicitly encourage representations of how scene geometry evolves. For manipulation, the critical question is not only what a future scene will look like, but how the spatial relationships between the robot and surrounding objects will change. We argue that predicting *geometry trajectories* provides complementary supervision for understanding how scenes evolve during interaction and learning policies that generalize beyond visual appearance. We introduce **GeoWAM**, which augments video–action learning with generative modeling of geometric evolution. A frozen 3D foundation model and a spatiotemporal geometry VAE transform multi-camera video windows into structured geometry latent sequences, which a geometry expert learns to predict through flow matching. Our central mechanism, *geometry gradient transfer*, uses asymmetric attention: the geometry expert reads video and action representations, while neither expert reads geometry. The geometry objective thus backpropagates into the deployed experts, encouraging their representations to support geometric prediction without making action generation depend on geometry tokens. The geometry branch is omitted at deployment, supporting both joint video–action denoising and action generation without future-video rollout. On LIBERO, GeoWAM achieves 98.3% success, improving over a matched baseline by 2.3 percentage points, and attains 65.2% on LIBERO-Plus. Our joint variant reaches 76.49% category-averaged success on LIBERO-Plus. These results support modeling geometric evolution as a training-time source of policy improvement without test-time geometry inference.

1 Introduction

World action models (WAMs) (Wang et al., 2026b; Yuan et al., 2026b; Ye et al., 2026; Kim et al., 2026; Intelligence et al., 2026; Li et al., 2026b) learn robot policies by coupling video prediction with action generation. This coupling turns future observations into a learning signal: beyond matching demonstrated actions, the model must anticipate how an interaction unfolds. Pretrained video models (Wan et al., 2025; Wu et al., 2025) provide a powerful starting point for this approach, but their predictive targets are primarily representations of visual appearance. For manipulation, a plausible future image is only part of the problem. Reaching, grasping, and placing require anticipating how objects move relative to the robot and to one another. The choice of what a world model predicts therefore matters as much as its ability to predict.

Consider placing an object into a drawer. The relevant evolution includes the object approaching the opening, clearing its boundary, and moving into the available space. Video prediction may implicitly capture these changes, yet reconstructing future observations does not directly require the representation to organize them as a geometric sequence. We argue that WAM training should also model *geometry trajectories*: temporally ordered representations of the scene's spatial configuration throughout an interaction. Here, geometry trajectories are implicit representations, not prescribed object poses, explicit motion tracks, or a hand-designed physical simulator. The objective is to complement visual prediction with supervision that makes evolving scene geometry a modeling target in its own right.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Pretrained 3D foundation models (Wang et al., 2025; 2026a; Lin et al., 2026) make such supervision accessible from robot videos without additional geometric annotations. Their multi-view representations provide a basis for describing the spatial structure of an observed scene. Using these representations to supervise future states, rather than only describe the current observation, offers a way to connect geometric knowledge to predictive policy learning. However, constructing a geometric target is not sufficient: its supervision must reach the representations that produce actions. Conversely, making action generation depend on geometric features introduces a requirement to obtain those features at deployment. This motivates our central question: *Can learning to predict geometric evolution improve a world action model without making geometry prediction a prerequisite for acting?*

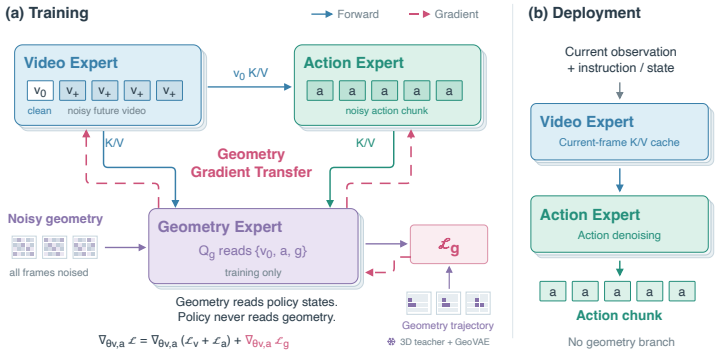


Figure 1: Geometry Gradient Transfer. Geometry attends to video and action, but not vice versa. Its loss backpropagates through their keys and values, improving control representations without geometry inputs at inference.

We introduce **GeoWAM**, which separates learning from geometry from consuming geometry at inference. GeoWAM augments video–action co-training with a geometry expert that predicts structured latent sequences. A frozen 3D foundation model (Wang et al., 2026a) jointly processes multi-camera video windows, and an independently trained spatiotemporal geometry VAE compresses its dense features while preserving camera, temporal, and spatial organization. The resulting sequences serve as targets for geometry flow matching alongside video and action prediction. This construction models the evolution of learned geometric representations without requiring explicit depth or point-map losses during policy training.

The central mechanism is *geometry gradient transfer*. Within a mixture-of-transformers(MoT) architecture, the geometry expert attends to video and action representations, while the video and action experts never attend to geometry. Although information flows toward the geometry expert in the forward pass, as shown in Figure 1, the geometry loss propagates back through the attended keys and values into the deployed experts. Geometric prediction can therefore shape the representations used for control without supplying geometry features to the action predictor. This asymmetry preserves a geometry-independent video–action computation throughout training, allowing the geometry branch to be omitted at deployment without changing that computation. The distinction is between geometry as a predictive training objective and geometry as a required inference-time input.

Our primary model, **GeoWAM**, generates actions from current-observation video representations without denoising future-video latents at deployment. During training, its geometry expert reads current-observation video tokens and action states, but not future-video tokens. The model thus learns from predictive video and geometry objectives without requiring either future-video rollout or geometry inference to act. We further introduce **GeoWAM-Joint**, an extension that jointly denoises future-video and action latents at deployment and allows the geometry expert to attend to noisy future-video tokens during training. Neither model requires the 3D teacher, geometry VAE, or geometry expert to run for action generation. This extension tests whether geometry gradient transfer also benefits policies that retain visual rollout.

GeoWAM achieves 98.3% success on LIBERO (Liu et al., 2023), exceeding a matched video–action baseline by 2.3 percentage points, and 65.20% category-averaged success on LIBERO-Plus (Fei et al., 2025), without test-time future-video rollout. GeoWAM-Joint further reaches 76.49% on LIBERO-Plus with joint video–action inference. Ablations support the contributions of both geometry gradient transfer and temporally varying supervision. Neither variant requires geometry computation at deployment.

Our contributions are threefold:

- **Predictive geometry supervision.** We use sequences of geometry features from a pretrained 3D foundation model as implicit trajectories of scene geometry evolution, providing predictive targets that complement video–action learning without explicit geometric labels.
- **Geometry gradient transfer.** We introduce an asymmetric expert-attention design that propagates geometry supervision into video and action experts while keeping their forward computation independent of geometry.
- **Policy improvement without test-time geometry.** We evaluate GeoWAM on LIBERO and LIBERO-Plus without future-video rollout, investigate gradient transfer and temporal supervision through controlled ablations, and extend the approach to joint video–action inference with GeoWAM-Joint.

2 Related Work

Video prediction and world action models. Generative policies model the action distribution conditioned on current observations, typically through diffusion (Chi et al., 2023) or flow matching (Black et al., 2024). A related line enriches this learning signal with future prediction: UniPi (Du et al., 2023) generates text-guided video plans that are mapped to actions, GR-1 (Wu et al., 2023) pretrains on video generation and finetunes jointly on image and action prediction, and VPP (Hu et al., 2025) shows that predictive visual representations can directly condition a generalist policy. Unified video–action models such as UVA (Li et al., 2025a) denoise future frames and actions within a single objective, whereas latent world models instead imagine in a compact state space (Hafner et al., 2023). World action models (WAMs) (Wang et al., 2026b) instantiate this idea on pretrained video generators (Wan et al., 2025; Wu et al., 2025) and currently define the state of the art on manipulation benchmarks (Ye et al., 2026; Kim et al., 2026; Li et al., 2026b; Intelligence et al., 2026). Fast-WAM (Yuan et al., 2026b) separates the benefit of video co-training from the cost of test-time imagination and provides the video–action backbone we build on. Across these methods the predicted future is largely a rendering of appearance, GeoWAM retains the architecture and instead changes what the model is trained to predict.

Training-time structured supervision. Structured prediction provides supervision beyond RGB reconstruction, including depth (Li et al., 2025b), point trajectories (Wen et al., 2024; Bharadhwaj et al., 2024), and combined geometric and semantic targets (Zhen et al., 2025; Zhang et al., 2025). Within WAMs, MECo-WAM (Zhang et al., 2026) distills VGGT-derived keyframe relations into a 4D expert and progressively removes policy access to geometry through a decayed read mask, while WAM4D (Li et al., 2026c) predicts future depth through training-time spatial registers. Related methods align intermediate representations with 3D trajectory fields (Yang et al., 2026) or geometric foundation features (Wu et al., 2026). DreamWAM (Yuan et al., 2026a) combines RGB–optical-flow denoising with gated residual branches supervised by geometry and semantic features. GeoWAM instead models compressed multi-view geometry trajectories through an auxiliary flow-matching expert whose attention to video and action carries geometric gradients into both deployed experts. Video and action never attend to geometry, allowing the auxiliary expert to be removed at inference without changing their forward computation.

Geometry representations and feature-space generation. Feed-forward reconstruction models such as DUSt3R (Wang et al., 2024), VGGT (Wang et al., 2025), VGGT-Ω (Wang et al., 2026a), and Depth Anything 3 (Lin et al., 2026) regress dense multi-view geometry and expose features that transfer to downstream tasks. Making such features generative typically requires a compression stage: Gen3R (Huang et al., 2026) introduces a spatiotemporal geometry adapter and aligns its posterior with video-VAE appearance latents, while TRELLIS (Xiang et al., 2025) learns structured latents for 3D assets. A complementary direction abandons pixels for feature-space prediction: DINO-WM (Zhou et al., 2024) plans in the latent space of a frozen visual encoder, V-JEPA 2 (Assran et al., 2025) predicts masked video representations, VFMF (Boduljak et al., 2026) forecasts vision-foundation-model features with autoregressive flow matching, and REPA (Yu et al., 2025) aligns denoising hidden states with pretrained visual features. GeoWAM combines these ingredients for control: a spatiotemporal VAE preserves camera, temporal, and spatial organization under a standard-normal prior and the resulting latents serve as an auxiliary flow-matching target rather than an inference-time input.

3 METHOD

3.1 OVERVIEW

At each control step s, GeoWAM receives multi-camera observations $o_s = \{o_s^c\}_{c=1}^{C}$, a language instruction ℓ, and proprioception p_s, and predicts the next H actions as a chunk $a_s \in \mathbb{R}^{H \times d_a}$. We learn the policy $\pi_\theta(a_s \mid o_s, \ell, p_s)$ from demonstrations, using a window $\mathcal{I}_s$ of current and future observations to supervise appearance and geometric evolution.

As illustrated in Figure 2, GeoWAM augments the video–action mixture-of-transformers (MoT) backbone of Fast-WAM (Yuan et al., 2026b) with a training-only geometry expert to learn from scene geometry trajectories. We first introduce the flow-matching formulation in Section 3.2. Section 3.3 describes how a frozen 3D foundation model and a pretrained geometry VAE construct these trajectories as supervision. Section 3.4 presents asymmetric attention for transferring geometry gradients into the video and action experts without making their predictions depend on geometry inputs. Finally, Section 3.5 details the training objective and the inference procedures for GeoWAM and its Joint variant, both of which omit geometry computation at deployment.

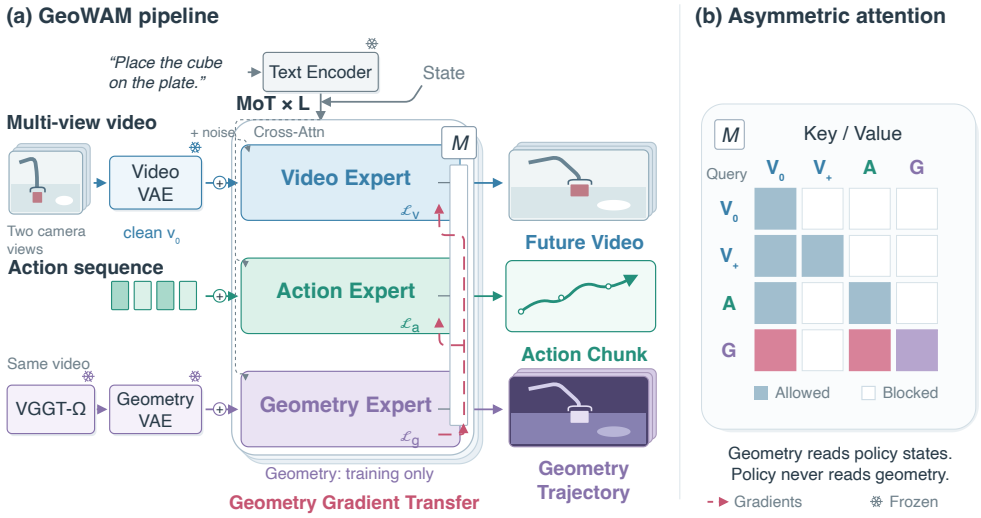


Figure 2: **GeoWAM overview.** GeoWAM jointly learns video, action, and geometry prediction, transferring gradients from geometry trajectories derived from a frozen 3D teacher and geometry VAE into the video and action experts without requiring geometry computation at deployment.

3.2 PRELIMINARIES

Flow Matching. We use flow matching to model video, geometry, and actions in their respective latent or action spaces. For each modality $m \in \{v, g, a\}$, a clean target x_m is interpolated with Gaussian noise at level $\sigma \in [0, 1]$:

$$x_m^\sigma = (1 - \sigma)x_m + \sigma\epsilon_m, \qquad u_m = \epsilon_m - x_m, \qquad \epsilon_m \sim \mathcal{N}(0, I), \tag{1}$$

An expert v_{θ_m} learns the path velocity u_m by minimizing

$$\mathcal{L}_{\mathrm{CFM}}(\theta_m) = \mathbb{E}\left[\left\| v_{\theta_m}(x_m^\sigma, \sigma, c_m) - u_m \right\|^2\right], \tag{2}$$

where c_m denotes the conditioning available to each expert under the attention mask defined below, and the expectation is over training examples, noise levels, and Gaussian noise. We follow the backbone's shifted noise schedule $\sigma = \rho(r)$, $r \sim \mathcal{U}(0, 1)$. At inference, Euler integration from $\sigma_0 = 1$ to $\sigma_K = 0$ transforms Gaussian noise into a prediction:

$$x_{m,k+1} = x_{m,k} + (\sigma_{k+1} - \sigma_k)\, v_{\theta_m}(x_{m,k}, \sigma_k, c_m)\,. \tag{3}$$

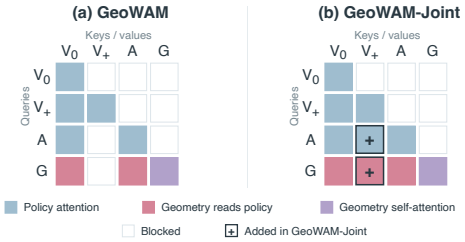


Figure 3: **Asymmetric attention masks.** Both variants allow geometry to attend to video and action without reciprocal access, while GeoWAM-Joint additionally exposes future-video tokens to the action and geometry experts.

3.3 Implicit Geometry Dynamics as Supervision

We use a frozen 3D foundation model, VGGT-Ω (Wang et al., 2026a), to jointly processes all cameras and frames of $\mathcal{I}_s$ in time-major camera order, so every feature is window-conditioned. And we retain four layers of dense features $F \in \mathbb{R}^{C\times T\times 4\times 196\times 2048}$, concatenate levels per spatial position, and normalize with fixed training-set statistics: $\widetilde{F} = \big(F - \mu^F\big)/\max\big(s^F, 10^{-6}\big)$, where $\mu^F, s^F \in \mathbb{R}^{4\times 2048}$ pool over training windows, cameras, times, and patch locations. We train a spatiotemporal geometry VAE whose encoder predicts a diagonal Gaussian posterior

$$q_\phi(z \mid \widetilde{F}) = \mathcal{N}\Big(\mu_\phi(\widetilde{F}), \mathrm{diag}\big(\exp \ell_\phi(\widetilde{F})\big)\Big), \qquad z = \mu_\phi + \exp(\ell_\phi/2) \odot \epsilon, \tag{4}$$

and whose decoder reconstructs all four teacher levels as $\widehat{F} = \widehat{\widetilde{F}} \odot s^F + \mu^F$. Training minimizes

$$\mathcal{L}_{\mathrm{VAE}} = \mathrm{SmoothL1}_\beta\Big(\widehat{\widetilde{F}}, \widetilde{F}\Big) + \Big\langle 1 - \cos(\widehat{F}, F)\Big\rangle + \beta_{\mathrm{KL}}(k)\, \overline{D}_{\mathrm{KL}}(q_\phi \| \mathcal{N}(0, I)), \tag{5}$$

with no depth, point-map, or camera supervision and no alignment to the video-VAE posterior. The latent keeps the frame and patch grid at 96 channels per position, $85.3\times$ fewer scalars than F. Freezing the teacher and the VAE, we standardize the posterior mean to obtain the geometry target: $g = \big(\mu_\phi(\widetilde{F}) - \mu^G\big)/s^G, \mu^G, s^G \in \mathbb{R}^{96}$, which flow matching predicts. The sequence g provides supervision for geometric evolution beyond a static observation. Its features incorporate temporal and cross-view context from the full teacher window, which is used only during training and is not required by the deployed policy.

3.4 Asymmetric Geometry Gradient Transfer

Write V_0, V_+, G, and A for clean current-observation video, noisy future-video, noisy geometry (all T frames), and noisy action tokens. All experts share one masked attention operation,

$$O_e = \mathrm{softmax}\left(\frac{Q_e\,[K_v; K_g; K_a]^\top}{\sqrt{d}} + M_e\right)[U_v; U_g; U_a], \qquad e \in \{v, g, a\}, \tag{6}$$

where Q_e, K_e, U_e are the queries, keys, and values of expert e, $d = 128$ is the head dimension, and M_e is an additive mask. Figure 3 summarizes the blockwise visibility for both variants.

This asymmetry keeps video and action predictions independent of geometry tokens while allowing geometric supervision to shape both experts. Specifically, $\mathcal{L}_g$ backpropagates through the video and action keys and values read by the geometry expert:

$$\frac{\partial \mathcal{L}_g}{\partial \theta_{va}} = \sum_{l=1}^{L}\left[\frac{\partial \mathcal{L}_g}{\partial K_v^l}\frac{\partial K_v^l}{\partial \theta_{va}} + \frac{\partial \mathcal{L}_g}{\partial U_v^l}\frac{\partial U_v^l}{\partial \theta_{va}} + \frac{\partial \mathcal{L}_g}{\partial K_a^l}\frac{\partial K_a^l}{\partial \theta_{va}} + \frac{\partial \mathcal{L}_g}{\partial U_a^l}\frac{\partial U_a^l}{\partial \theta_{va}}\right], \tag{7}$$

a cross-expert gradient path through shared attention rather than weight sharing or an injected update. Because no video or action query attends to geometry at any layer, their activations depend on it only through parameters; by induction over layers, deleting the geometry tokens and attention rows leaves the video and action forward computation unchanged, so the geometry expert is dropped at inference. To isolate this pathway we rescale geometry's incoming gradients by

$$R_\alpha(x) = \mathrm{sg}(x) + \alpha\big(x - \mathrm{sg}(x)\big), \tag{8}$$

where sg is stop-gradient: α=1 is the full method, and α=0 blocks geometry gradients from reaching the deployed experts while leaving forward values and attention edges intact.

3.5 Training Objective and Inference

Noise assignment. We sample the video and action noise levels independently and use the action noise level for geometry. Each modality receives an independent Gaussian noise tensor:

$$\sigma_v = \rho(r_v), \quad \sigma_a = \sigma_g = \rho(r_a), \quad r_v, r_a \sim \mathcal{U}(0,1), \quad \epsilon_v, \epsilon_a, \epsilon_g \sim \mathcal{N}(0, I). \tag{9}$$

The current-observation video latent remains clean and is excluded from the video loss. All T geometry frames, including the initial frame, are noised and contribute to the geometry loss; no clean geometry anchor is provided.

Objective. We jointly optimize flow-matching objectives for video, action, and geometry using the interpolation path in Eq. 1. An auxiliary cosine loss encourages the predicted geometry velocity to match the direction of its target, giving the combined objective

$$\mathcal{L} = \lambda_v \mathcal{L}_v + \lambda_a \mathcal{L}_a + \lambda_g \mathcal{L}_g + \lambda_{\cos} \mathcal{L}_{\cos}, \qquad \mathcal{L}_m = \mathbb{E}\Big[w(\sigma_m) \Big\langle \big(\widehat{u}_m - u_m\big)^2 \Big\rangle_m \Big], \tag{10}$$

where $m \in \{v, a, g\}$ indexes the three modalities and $\langle \cdot \rangle_m$ averages over the coordinates included in each loss. We set $\lambda_v = \lambda_a = \lambda_g = 1$ and $\lambda_{\cos} = 0.2$ and update all three experts and the proprioception encoder, while keeping the video VAE, text encoder, 3D teacher, and GeoVAE frozen.

Inference without future rollout. We encode the current observation, run the video expert once on the current-frame tokens, and cache the resulting keys and values; actions are denoised from Gaussian noise with K Euler steps of Eq. 3, after which we execute a prefix of the predicted chunk and replan. GeoWAM-Joint instead retains action's access to V_+ (Figure 3b) and jointly denoises future video and actions, restoring the clean current-observation latent after each update. The geometry expert is omitted in both inference settings.

4 Experiments

We evaluate whether predicting implicit geometry trajectories improves robot manipulation without requiring geometry computation at deployment. Our experiments examine task success on LIBERO (Liu et al., 2023), robustness to distribution shifts on LIBERO-Plus (Fei et al., 2025), and the contributions of cross-expert gradient transfer and temporally varying geometry targets.

Table 1: **Manipulation performance on LIBERO.** Success rates (%) across four task suites and their average. Embodied PT. denotes embodied-policy pretraining; bold indicates the best score in each column.

Method	Embodied PT.	Spatial	Object	Goal	Long	**Average**
OpenVLA (Kim et al., 2024)	Yes	84.70	88.40	79.20	53.70	76.50
π_0 (Black et al., 2024)	Yes	96.80	98.80	95.80	85.20	94.10
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	Yes	98.80	98.20	98.00	92.40	96.90
LingBot-VA (Li et al., 2026a)	Yes	98.50	99.60	97.20	**98.50**	98.50
Motus (Bi et al., 2026)	Yes	96.80	99.80	96.60	97.60	97.70
No-rollout WAMs without embodied-policy pretraining						
Fast-WAM (Yuan et al., 2026b)	No	98.20	**100.00**	97.00	95.20	97.60
MECo-WAM (Zhang et al., 2026)	No	98.80	**100.00**	98.20	95.80	98.20
GeoWAM (ours)	No	98.40	99.60	**98.80**	96.40	98.30
Additional WAM variants						
Fast-WAM-Joint	No	**99.60**	99.40	98.20	96.80	98.50
GeoWAM-Joint (ours)	No	99.20	99.40	**98.80**	97.20	**98.65**
Fast-WAM-IDM	No	98.80	97.80	97.80	97.60	98.00
Fast-WAM w/o video co-training	No	89.20	99.20	95.40	90.00	93.50

4.1 Experimental Setup

Datasets and metrics. We train on LIBERO demonstrations and evaluate on its Spatial, Object, Goal, and Long suites (Liu et al., 2023). Each suite contains ten tasks, and we evaluate 50 trials per task, giving 2,000 rollouts per evaluation seed. We report success rates for each suite and their unweighted average. To assess robustness beyond the training distribution, we evaluate the LIBERO-trained checkpoints on LIBERO-Plus (Fei et al., 2025) without additional training. This benchmark contains 10,030 task variants spanning camera viewpoints, robot initial states, language instructions, lighting, background textures, sensor noise, and object layouts. We evaluate each variant once per seed and report the category-wise success rates and their macro-average.

Implementation details. We initialize the video backbone from Wan2.2-TI2V-5B (Wan et al., 2025) and initialize compatible action and geometry transformer weights by interpolation (Yuan et al., 2026b). The frozen video VAE encodes the cameras separately before their latents are packed along the width dimension. The frozen VGGT-Ω teacher jointly processes the multi-camera window; a separately pretrained, frozen geometry VAE produces a $2 \times 96 \times 9 \times 14 \times 14$ posterior-mode target before channel normalization and packing. Only the three experts and the proprioception encoder are updated during policy training. We use AdamW with a peak learning rate of 10^{-4}, weight decay 10^{-2}, and $(\beta_1, \beta_2) = (0.9, 0.95)$ for ten epochs. The learning rate warms up over the first 5% of optimizer steps and decays with a cosine schedule to 10^{-6} and gradients are clipped at norm 1.0. All experiments run on 4× A800 GPUs with BF16 and gradient checkpointing.

4.2 Main Experimental Results

Manipulation performance on LIBERO. Table 1 compares GeoWAM with published VLA and WAM results, distinguishing the use of embodied-policy pretraining. GeoWAM achieves **98.30%** average success, compared with 97.60% for Fast-WAM and 98.20% for MECo-WAM. GeoWAM also remains within 0.20 points of LingBot-VA and Fast-WAM-Joint (both 98.50%), while avoiding embodied-policy pretraining and future-video denoising, respectively. GeoWAM-Joint achieves **98.65%** average success, the highest average among the listed methods, exceeding GeoWAM by 0.35 points and the reported Fast-WAM-Joint result by 0.15 points. For the latter, our matched video–action baseline uses GeoWAM's per-camera video encoding: GeoWAM improves its average success by 2.30 points and its Long score by 5.80 points (Section 4.3).

Table 2: **Robustness under distribution shifts on LIBERO-Plus.** Success rates (%) across seven perturbation categories; Avg. is their unweighted mean. Emb. PT. denotes embodied-policy pretraining. Bold and underlined values indicate the best and second-best scores in each column.

Type	Method	Emb. PT.	Cam.	Robot	Lang.	Light	Bg.	Noise	Layout	**Avg.**
VLA	UniVLA (Bu et al., 2025)	Yes	1.80	46.20	69.60	69.00	81.00	21.20	31.90	45.81
	OpenVLA-OFT (Kim et al., 2025)	Yes	56.40	31.90	79.50	88.70	**93.30**	<u>75.80</u>	74.20	71.40
	π_0 (Black et al., 2024)	Yes	13.80	6.00	58.80	85.00	<u>81.40</u>	**79.00**	68.90	56.13
	π_0-FAST (Pertsch et al., 2025)	Yes	**65.10**	21.60	61.00	73.20	73.20	74.40	68.80	62.47
WAM	WorldVLA (Cen et al., 2025)	Yes	0.10	27.90	41.60	43.70	17.10	10.90	38.00	25.61
WAMs without test-time future-video rollout										
WAM	Fast-WAM (Yuan et al., 2026b)	No	16.26	44.13	66.82	79.77	52.60	38.54	61.38	51.36
	DreamWAM-uncond (Yuan et al., 2026a)	No	30.39	59.23	77.29	85.90	64.68	56.65	69.97	63.44
	GeoWAM (ours)	No	25.89	56.32	87.38	95.36	62.92	53.84	74.69	65.20
WAMs with joint future-video and action denoising										
WAM	Fast-WAM-Joint (Yuan et al., 2026b)	No	39.59	60.90	92.32	94.57	57.62	58.59	80.52	69.16
	DreamWAM (Yuan et al., 2026a)	No	53.78	<u>63.61</u>	**94.80**	**96.67**	71.56	67.15	<u>80.72</u>	<u>75.47</u>
	GeoWAM-Joint (ours)	No	<u>58.16</u>	**74.00**	<u>93.36</u>	<u>96.23</u>	63.85	68.02	**81.84**	**76.49**

Robustness on LIBERO-Plus. Table 2 compares robustness under seven perturbation categories, separating no-rollout WAMs from joint-generation variants. Without adapting to LIBERO-Plus, the primary GeoWAM model achieves **65.20%** macro-average success, exceeding the reported Fast-WAM and DreamWAM-uncond results by 13.84 and 1.76 percentage points, respectively. The GeoWAM-Joint extension reaches **76.49%**, the highest reported average among the methods in the table, compared with 69.16% for Fast-WAM-Joint and 75.47% for DreamWAM. These gains are achieved with fewer types of auxiliary supervision. DreamWAM combines RAFT-derived optical flow, Depth Anything V3 geometry features, and DINOv2 semantic features, whereas GeoWAM augments video–action learning with a single auxiliary target: implicit geometry trajectories derived from VGGT-Ω. GeoWAM achieves higher reported macro-average success in both the no-rollout and joint settings, by 1.76 and 1.02 percentage points. This suggests that a unified geometry trajectory can provide effective supervision without separate motion and semantic prediction objectives, although the cross-paper comparison does not isolate the effect of supervision type. Within GeoWAM, the Joint extension improves the macro-average by 11.29 points, with the largest category gain in camera viewpoints (25.89% to 58.16%). The improvement therefore supports the Joint extension as a robustness-oriented deployment option, without isolating the effect of rollout alone.

Behavior under perturbed robot initial states. Figure 4 shows two paired LIBERO-Plus rollouts under Robot Initial States perturbations. Each pair uses the same task variant and initial simulator state for Fast-WAM and GeoWAM-Joint. Despite the shared starting observation, their subsequent interactions differ: the highlighted end-effector–object configurations at steps 74 and 52 accompany unsuccessful Fast-WAM rollouts and successful GeoWAM-Joint rollouts, respectively. These

examples illustrate how the policies behave when the robot starts from a perturbed configuration, complementing the category-level success rates.

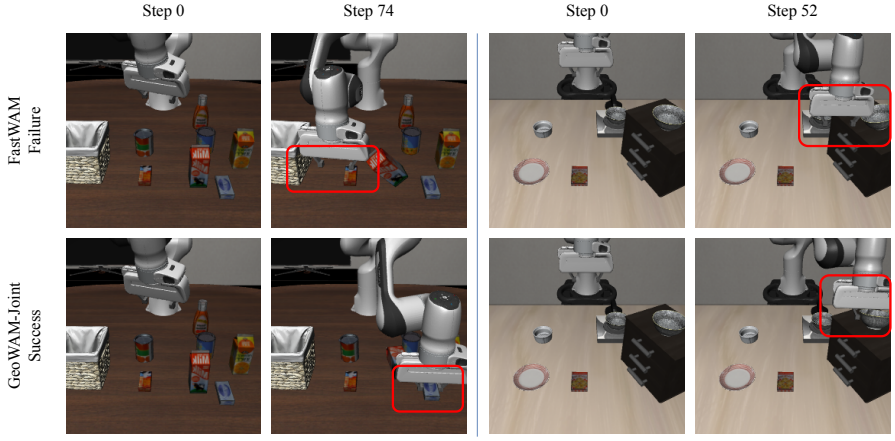


Figure 4: **Paired behavior under robot initial-state perturbations on LIBERO-Plus.** The left and right column pairs show two examples, each at step 0 and a later executed-action step (74 and 52, respectively). Rows compare Fast-WAM (no rollout) and GeoWAM-Joint from the same initial state within each example. Red boxes highlight the end-effector–object interaction regions; success and failure labels refer to the outcomes of the complete rollouts.

4.3 Ablation Study

Table 3 examines how gradient connectivity, temporal supervision, and geometry anchoring affect the primary no-rollout model. Matched FastWAM uses the same per-camera video-VAE encoding and latent packing as GeoWAM, providing a baseline for evaluating the addition of geometry supervision. All variants generate actions without a geometry branch at deployment, so their performance differences arise from training rather than additional inference-time geometry inputs.

Table 3: **LIBERO ablations in the no-rollout setting.** Success rates (%) over 50 trials per task and ten tasks per suite.

Method	Spatial	Object	Goal	Long	Avg.
Matched FastWAM	97.40	**99.60**	96.40	90.60	96.00
GeoWAM-Detach	97.60	98.80	96.00	91.20	95.90
Static-Geometry Transfer	96.40	99.20	97.80	95.40	97.20
Decayed-Anchor GeoWAM	**99.00**	99.40	97.60	94.80	97.70
GeoWAM	98.40	**99.60**	**98.80**	**96.40**	**98.30**

Cross-expert gradients drive policy improvement. GeoWAM-Detach retains the geometry objective and forward attention connections but blocks gradients through the video/action keys and values read by geometry. Its performance remains close to Matched FastWAM and below GeoWAM, indicating that adding a geometry expert alone is insufficient to recover the gain. This comparison supports the role of geometry supervision in shaping the deployed experts through attention.

Temporal geometry targets strengthen supervision. Static-Geometry Transfer repeats the initial geometry latent across time while preserving the target shape and token count. It improves average success over Matched FastWAM but falls short of GeoWAM, suggesting that static geometric structure provides useful supervision and that modeling its evolution adds further value. GeoWAM improves over this control on every suite, consistent with the benefit of learning how scene structure changes during interaction.

A clean-anchor curriculum offers no overall advantage. Decayed-Anchor GeoWAM begins with a clean initial geometry slice and gradually removes this conditioning during training, adjusting attention visibility and loss masking accordingly. Although it improves Spatial performance, it underperforms fully noised GeoWAM on average and on the Long suite. A clean geometric reference may simplify the auxiliary prediction task, but these results show no corresponding overall benefit for action learning.

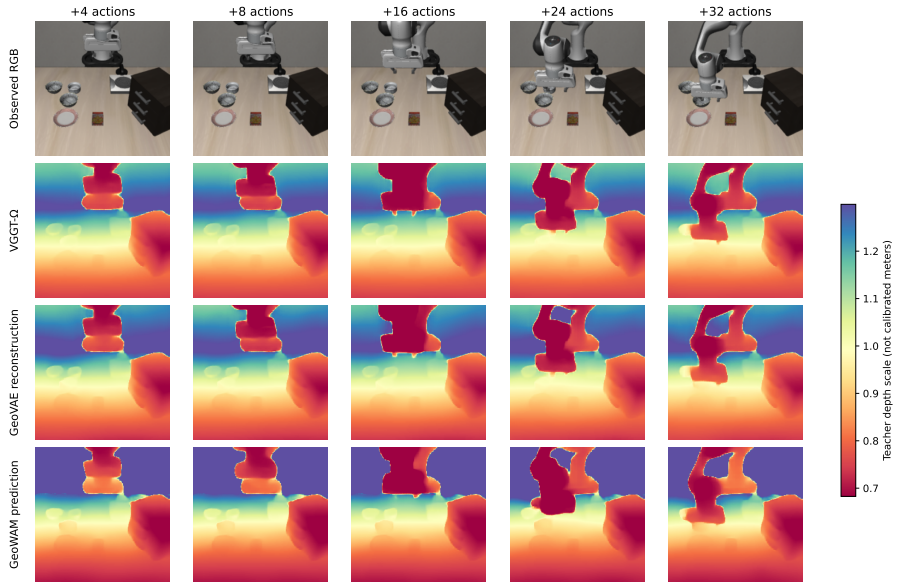


Figure 5: **Diagnostic prediction of scene geometry.** Main-camera observations and depth maps. Rows show observed RGB, frozen VGGT-Ω depth, depth decoded from GeoVAE-reconstructed teacher features, and depth decoded from geometry latents generated by GeoWAM-Joint.

4.4 Geometry Prediction and Mechanism Analysis

The auxiliary expert provides a way to inspect the geometric trajectories learned during policy training, beyond their effect on task success. For diagnostic inference, generated geometry latents are inverse-normalized, decoded into the four teacher feature levels, and passed through the frozen VGGT-Ω dense prediction head to obtain depth sequences. This requires neither a newly fitted depth probe nor a depth loss during GeoWAM training. Figure 5 presents a main-camera example from GeoWAM-Joint, comparing the observed future RGB frames, raw-teacher depth, geometry-VAE reconstruction depth, and predicted depth over the same action window. Teacher depth is a reference prediction rather than ground-truth metric depth, and this additional diagnostic path is not part of action deployment.

The GeoVAE reconstructions closely preserve the teacher's broad scene layout and changing robot silhouette in this example. The predicted sequence also captures the arm's movement toward the tabletop and the resulting changes in foreground geometry. However, the predicted background is more uniform, and differences remain in depth values and local contours. This example illustrates that the auxiliary representation retains decodable information about future scene structure, while also exposing errors beyond those introduced by VAE reconstruction.

5 Conclusion and Limitations

In this paper, We presented GeoWAM, which uses implicit geometry trajectories to enrich video–action learning through asymmetric gradient transfer. By allowing the geometry objective to shape video and action representations without making their predictions depend on geometry tokens, GeoWAM improves policy learning while preserving geometry-free deployment. Results on LIBERO and LIBERO-Plus, together with the gradient-detachment and static-target ablations, support the value of both the transfer pathway and temporally varying geometric supervision. GeoWAM also accommodates joint video–action inference, extending its benefits to policies that retain future rollout. The approach nevertheless adds training cost and inherits errors from the frozen geometry teacher and its compression model. Our evaluation is limited to simulated manipulation, and robustness remains uneven across perturbations; real-world transfer and consistency across training seeds remain to be established. These limitations motivate more efficient geometric supervision and broader evaluation across environments and embodiments.

AI USE STATEMENT

We used generative AI tools to improve the clarity and readability of the manuscript, assist with data processing and related code. The authors reviewed the AI-assisted text and analyses and checked the relevant code and processed data for correctness and consistency with the intended experimental procedures. All final methodological decisions and scientific conclusions were made by the authors, who take responsibility for the complete content of this paper.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *ECCV*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *CVPR*, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Gabrijel Boduljak, Yushi Lan, Christian Rupprecht, and Andrea Vedaldi. VFMF: World modeling by forecasting vision foundation model features. In *ICML*, 2026.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. In *RSS*, 2025.

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

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2023.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *ICML*, 2025.

Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. Gen3R: 3D scene generation meets feed-forward reconstruction. In *CVPR*, 2026.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model, 2024. *arXiv preprint arXiv:2406.09246*, 2024.

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning, 2026. *arXiv preprint arXiv:2601.16163*, 2026.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *RSS*, 2025a.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, et al. WAM4D: Fast 4D world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026c.

Yixuan Li, Yuhui Chen, Mingcai Zhou, and Haoran Li. QDepth-VLA: Quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025b.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *CVPR*, 2026a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024.

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, et al. World action models: The next frontier in embodied ai, 2026. *arXiv preprint arXiv:2605.12090*, 2026b.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *RSS*, 2024.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3D representation for consistent world modeling. In *ICLR*, 2026.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D latents for scalable and versatile 3D generation. In *CVPR*, 2025.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, et al. 4d-wam: Infusing spatiotemporal awareness into world action models through trajectory fields. *arXiv preprint arXiv:2608.08023*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. DreamWAM: Beyond RGB future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026a.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026b.

Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4D geometric priors for inference-efficient world action models. *arXiv preprint arXiv:2607.05468*, 2026.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. TesserAct: Learning 4D embodied world models. In *ICCV*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

SUPPLEMENTARY MATERIAL

This appendix provides representation and architecture details (Appendix A), training and inference specifications (Appendix B), evaluation protocols (Appendix C), ablation definitions (Appendix D), and geometry diagnostics (Appendix E).

A REPRESENTATION AND ARCHITECTURE DETAILS

A.1 GEOMETRY VAE AND FIXED STATISTICS

Our LIBERO windows contain two cameras and nine frames at action offsets $(0, 4, \ldots, 32)$, aligned with a 32-step action chunk. The teacher receives images in time-major camera order and returns dense features from layers $\{4, 11, 17, 23\}$, excluding camera and register tokens. Each 224×224 image yields a 14×14 grid of 2048-dimensional features per layer. The dense stem concatenates the four feature levels into 8192 channels per spatial position. With cameras temporarily folded into the batch, a 3×3 Conv2d maps 8192 channels to 1024, followed by a temporal Conv3d mapping 1024 to 512. Encoder stages use widths $(512, 256, 128, 128)$ with two residual blocks per stage. The posterior head maps 128 channels to 192, split into 96-channel mean and log variance. There is no temporal or spatial downsampling in this configuration. A corresponding decoder reconstructs all four 2048-dimensional feature levels. The latent has approximately $8192/96 = 85.3$ times fewer scalars than the teacher feature tensor while preserving its frame and spatial indices.

Before the posterior head, cross-camera attention reshapes the hidden states to $[BT, C \cdot 14 \cdot 14, 128]$ and uses eight attention heads with learned camera, row, and column embeddings. A corresponding module appears at the decoder's low-dimensional input. Attention output projections and final feed-forward projections are zero-initialized. This permits cross-view exchange without imposing equality between camera latents. Temporal convolutions use left padding, but GroupNorm aggregates over the temporal axis and the teacher uses full-window context.

For feature channel d of teacher layer l, statistics pool over training windows, cameras, times, and patch locations:

$$\mu^F_{l,d} = \mathbb{E}_{w,c,t,p} F_{w,c,t,l,p,d}, \qquad (s^F_{l,d})^2 = \mathbb{E}_{w,c,t,p} F^2_{w,c,t,l,p,d} - (\mu^F_{l,d})^2. \tag{11}$$

Distributed calibration accumulates sums, squared sums, and counts in float64 over disjoint window subsets; standard deviations are floored at 10^{-6}. Latent statistics are computed analogously from the frozen posterior modes, pooling over window, camera, time, height, and width for each of 96 channels. Neither calibration is fitted on evaluation observations. They standardize marginal channel scales, not cross-channel covariance.

The posterior is $q_\phi(z \mid \widetilde{F}) = \mathcal{N}(\mu_\phi, \mathrm{diag}(\exp \eta_\phi))$, where $\widetilde{F} = (F - \mu^F)/s^F$. Using a reparameterized sample, VAE pretraining minimizes

$$\begin{aligned} \mathcal{L}_{\mathrm{VAE}} = {} & \mathrm{SmoothL1}_{0.15}(\widehat{\widetilde{F}}, \widetilde{F}) + \Big\langle 1 - \cos(\widehat{F}, F) \Big\rangle \\ & + \beta_{\mathrm{KL}}\, \overline{D}_{\mathrm{KL}}\big(q_\phi \| \mathcal{N}(0, I)\big), \quad \widehat{F} = \widehat{\widetilde{F}} \odot s^F + \mu^F. \end{aligned} \tag{12}$$

Cosine is computed over raw 2048-dimensional feature vectors. The posterior log variance is clamped to $[-30, 20]$. The scalar-averaged prior penalty is

$$\overline{D}_{\mathrm{KL}} = \frac{1}{2} \left\langle \mu_\phi^2 + \exp \eta_\phi - 1 - \eta_\phi \right\rangle, \qquad \beta_{\mathrm{KL}}(k) = 10^{-4} \min\left(1, \frac{k+1}{5000}\right), \tag{13}$$

where k is the zero-based VAE optimizer step. Feature Smooth-L1 and raw-feature cosine both have coefficient 1. The geometry VAE is trained independently before policy training; no video-posterior alignment is used in this configuration.

A.2 PATCH POSITIONS AND EXPERT INITIALIZATION

Table 4 summarizes the packed inputs. Video and geometry use Conv3d patch embeddings with kernel and stride $(1, 2, 2)$. All experts use 30 blocks and project to 24 attention heads of dimension

Table 4: Expert inputs and tokenization for LIBERO, omitting the batch dimension.

Expert	Input shape	Hidden width	Tokens
Video	$48 \times 3 \times 14 \times 28$	3072	294
Geometry	$96 \times 9 \times 14 \times 28$	512	882
Action	32×7	1024	32

128. Video latents retain the frozen Wan VAE's built-in normalization; geometry uses fixed channel statistics of its posterior modes.

Let geometry and video patch grids be (T_g, H_g, W_g) and (T_v, H_v, W_v), with per-camera widths $w_g = W_g/C$ and $w_v = W_v/C$. A geometry token at camera c (zero-based), local column w, row h, and time t uses rotary coordinates

$$(t,\ \lfloor hH_v/H_g \rfloor\ ,\ c\,w_v + \lfloor w\,w_v/w_g \rfloor)\,. \tag{14}$$

Both spatial grids are 7×14 after patch embedding, so the spatial mapping is the identity in our configuration. The second camera retains its packed-width offset; its width coordinate does not restart at zero. Geometry has nine time indices and video has three; these indices are not rescaled to claim identical temporal support. The 128 real coordinates of each attention head are allocated as $(44, 42, 42)$ to time, height, and width, with rotary base 10000. Geometry also has a zero-initialized learned camera embedding of shape $[2, 512]$. Actions use one-dimensional rotary positions for their 32 sequence indices.

Video, action, and geometry feed-forward widths are 14336, 4096, and 2048, respectively. Compatible action and geometry backbone tensors are initialized by linear interpolation of Wan tensors along resized dimensions. When enabled, alpha scaling multiplies tensors whose last dimension changes by the square root of the source-to-target last-dimension ratio. Input patch embeddings, prediction heads, and geometry camera embeddings are not loaded from these interpolated backbone tensors. Patch width two divides each camera's latent width of 14, so no patch crosses a camera boundary. In particular, a geometry patch contains $96 \cdot 1 \cdot 2 \cdot 2 = 384$ scalar inputs and maps to 512 hidden coordinates. This dimension count does not establish invertibility of the learned embedding.

B Training and Inference Details

B.1 Noise Sampling and Loss Reduction

The continuous scheduler uses $\rho(r) = 5r/(1 + 4r)$ and passes 1000σ as the expert timestep. For $y(\sigma) = \exp[-2(\sigma - \frac{1}{2})^2]$, its loss weight is

$$\begin{aligned} \bar{\sigma}_i &= \rho(1 - i/1000), \quad i = 0, \ldots, 999, \\ y_{\min} &= \min_i y(\bar{\sigma}_i), \qquad Z = \frac{1}{1000} \sum_i \big[y(\bar{\sigma}_i) - y_{\min}\big], \\ w(\sigma) &= \frac{y(\sigma) - y_{\min}}{Z + 10^{-10}}. \end{aligned} \tag{15}$$

Weights are applied to per-sample losses before batch averaging. Equation 10 combines the three velocity-regression terms with a channel-wise cosine term $\mathcal{L}_{\cos}$ between predicted and target geometry velocity. For the reported geometry-supervised experiments, we use $\lambda_v = \lambda_a = \lambda_g = 1$ and $\lambda_{\cos} = 0.2$; the matched video–action baseline has no geometry objective. The video loss omits the clean initial slice and applies the video padding reduction; actions exclude padded tokens. The implemented geometry loss averages all positions without an additional padding mask. Consequently, repeated endpoint frames supplied by dataset padding can contribute to its target.

B.2 Action Deployment

Deployment uses $K = 10$ Euler steps on the shifted grid $\sigma_k = \rho(1 - k/10)$, $k = 0, \ldots, 10$. In the primary no-rollout model, action queries read only current-observation video keys and values, which are cached after a single pass over the video expert, and no future-video latent is denoised.

In the Joint extension, the same ten steps integrate the video and action flow fields together, the clean current-observation slice is restored after each update, and action queries additionally read the evolving future-video keys and values. Both settings execute ten actions before replanning.

C Evaluation Protocols

C.1 LIBERO and LIBERO-Plus

LIBERO evaluation uses ten tasks in each of four suites and 50 trials per task. The suite success rate is the fraction of successful rollouts among its 500 trials; the reported average weights the four suites equally. LIBERO-Plus evaluation uses the LIBERO-trained policy without fine-tuning and runs one trial for each of its 10,030 variants, using the variant's supplied initial state and language instruction.

Table 5: LIBERO-Plus evaluation coverage. Each variant contributes one rollout per evaluation seed.

Perturbation category	Variants
Camera viewpoints	1,599
Robot initial states	1,550
Language instructions	1,537
Lighting	1,142
Background textures	1,076
Sensor noise	1,601
Object layouts	1,525
Total	10,030

For category j, let S_j and N_j be its successful and evaluated rollout counts. The LIBERO-Plus average is

$$\text{Avg.} = \frac{100}{7} \sum_{j=1}^{7} \frac{S_j}{N_j}. \tag{16}$$

This category macro-average differs from the pooled rate $100 \sum_j S_j / \sum_j N_j$, since category sizes are unequal. We compute the average before rounding the displayed category scores. Complete evaluation requires the evaluated count in every category to match Table 5; infrastructure errors are not valid completed rollouts.

C.2 Preprocessing and Rollout Settings

Both benchmarks use the action and proprioception normalization statistics saved with the LIBERO training run, rather than statistics estimated from evaluation data. Language perturbations are passed to the text encoder as the actual variant instruction. Each camera is encoded separately by the frozen video VAE before latent-width packing, matching policy training. The policy predicts 32 actions using ten denoising steps and executes ten actions before replanning, without action ensembling. The evaluation code allows up to 400 control steps for Spatial, Object, and Goal, and 700 for Long, excluding initialization settling steps. LIBERO-Plus uses 30 settling steps before policy execution. The primary GeoWAM policy uses current-observation video conditioning only; GeoWAM-Joint additionally denoises future-video latents. Future RGB decoding is not required for joint action generation. Neither setting executes the geometry branch to produce actions.

C.3 Comparison Scope

The published baselines in the main tables are distinct from our locally trained Matched FastWAM control. In particular, the latter matches GeoWAM's per-camera video encoding, whereas the original Fast-WAM packs camera images before video-VAE encoding. Cross-paper comparisons also differ in auxiliary supervision and potentially other training choices; they should not be interpreted as controlled interventions on geometry supervision alone. Comparisons between GeoWAM and

GeoWAM-Joint change both training-time attention connectivity and test-time future-video denoising. They assess the complete variants rather than isolate the effect of rollout. Small differences in reported success rates should not be read as statistically established improvements without repeated-run uncertainty estimates.

D Ablation Definitions

The controls in Table 3 use the primary no-rollout deployment setting. We detail their training interventions below to distinguish changes to supervision from changes to conditioning.

Matched FastWAM. This control retains the video–action policy and GeoWAM's per-camera video-VAE encoding and latent packing, but removes the geometry expert and its objectives. It tests the addition of geometry supervision without conflating it with the camera-encoding convention.

GeoWAM-Detach. The operator R_α in Eq. 8 is applied to the video/action keys and values consumed by geometry. Setting $\alpha = 0$ preserves forward values while blocking this cross-expert gradient path; the full method uses $\alpha = 1$. The control retains the geometry objective and does not detach shared conditioning embeddings. It therefore isolates attention-mediated transfer, not every possible path through shared trainable components.

Static-Geometry Transfer. This control repeats the first geometry latent along the temporal axis before noise assignment, preserving target shape and token count while removing variation across target frames. The repeated latent comes from the full-window teacher and geometry encoder, so it can still contain information from later observations. This is a control for temporally varying targets, not a strictly current-observation-only geometry teacher.

Decayed-Anchor GeoWAM. The probability of keeping the initial geometry slice clean decreases linearly from one to zero over the first half of training and remains zero thereafter. Clean slices are excluded from geometry velocity supervision. The anchor query reads current video and itself, but not future geometry or action; later geometry queries can read the anchor, other geometry tokens, and action states. This restricted anchor connectivity remains part of the variant even after the clean-anchor probability reaches zero. Consequently, the ablation changes both the noise curriculum and geometry attention structure rather than isolating noise assignment alone.

E Geometry Diagnostics and Visualization

E.1 Depth Decoding

For diagnosis, the geometry expert can also denoise a Gaussian-initialized latent sequence while reading video and action states. This is an additional diagnostic computation, not the standard action-inference path. Predicted geometry is unpacked by camera and decoded through

$$\widehat{z} = \widehat{g} \odot s^G + \mu^G, \qquad \widehat{\widetilde{F}} = D_\psi(\widehat{z}), \qquad \widehat{F} = \widehat{\widetilde{F}} \odot s^F + \mu^F. \tag{17}$$

The reconstructed four feature levels can then be passed to the frozen VGGT-Ω dense head to visualize depth. We compare raw-teacher depth, VAE-reconstructed depth, and predicted-geometry depth to distinguish representation reconstruction from policy-side geometry prediction. These are teacher-based diagnostics rather than ground-truth depth measurements. Neither this decoder path nor a depth loss is required for action deployment. The current evaluation implementation can still instantiate geometry modules when loading a checkpoint; omission of their execution does not by itself imply that they have been removed from memory.

E.2 Interpreting Qualitative Comparisons

For a given observation window, teacher depth and VAE-reconstructed depth use the same observed images; their difference reflects representation reconstruction error. Predicted depth instead comes from generated geometry latents and can additionally differ because of future-prediction errors or

ambiguity. In closed-loop control, a predicted trajectory need not match the subsequently realized trajectory exactly, since the policy replans before executing the entire predicted action chunk. Depth visualizations therefore complement, but do not replace, task-success evaluation. They do not establish metric depth accuracy or isolate the causal contribution of gradient transfer. Likewise, attention visualizations alone would show where representations attend, not whether the corresponding gradients improve control; the Detach intervention provides a more direct test of that pathway.