GEOMETRIC LATENT STATES FOR SPATIALLY AWARE ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting physical scene dynamics under interaction is fundamental to reliable robotic manipulation. World Action Models (WAMs) have emerged as a promising paradigm by framing scene evolution as future video prediction. However, existing WAMs predominantly operate in appearance-oriented video latent spaces, lacking the 3D spatial understanding required for precise manipulation. While per-frame depth prediction constrains distances to visible surfaces without capturing spatial correspondences across views and time, emerging 3D foundation models offer rich spatial representations whose high dimensionality and redundancy make direct prediction computationally demanding. We present GeoAct, a world action model that grounds robot manipulation in latent 3D geometric prediction. GeoAct jointly models complementary future states: RGB latents to capture fine-grained visual dynamics and geometric latents to model evolving 3D scene structure. To obtain a compact spatial state, we design GeoAE to compress multi-layer VGGT-Omega features into a low-dimensional spatiotemporal latent space through joint feature reconstruction and explicit 3D geometric decoding. During training, GeoAct co-denoises future RGB, geometry, and actions, while retaining RGB-to-action inference without online geometry encoding at deployment. GeoAct achieves mean success rates of 84.6% on LIBERO-Plus and 92.6% on RoboTwin 2.0, with strong performance under camera viewpoint perturbations. Real-world experiments further demonstrate GeoAct's improved performance on manipulation tasks requiring precise spatial coordination.

1 INTRODUCTION

Reliable, precise manipulation in real-world environments requires a physical understanding of objects, spatial relationships, and interaction dynamics (Bai et al., 2026; Finn & Levine, 2017; Zhu et al., 2025; Ke et al., 2025). World action models (WAMs) have emerged as a promising paradigm for robot manipulation by jointly modeling robot actions and future visual states (Bi et al., 2026; Yuan et al., 2026; Wu et al., 2024). However, most existing WAMs operate predominantly in appearance-oriented 2D latent spaces, predicting visual transitions while lacking the 3D spatial perception and structural understanding required for precise action generation.

Existing approaches incorporate spatial supervision into world action models primarily through explicit geometric prediction or static feature alignment (Zhen et al., 2024; Zhang et al., 2026). Explicit per-frame depth prediction (Guo et al., 2026; Lopetegui Gonzalez et al., 2026) constrains physical distances to visible surfaces, without capturing spatial correspondences across views and time (Figure 2(a)). Alternatively, feature alignment methods (Li et al., 2026a) treat 3D foundation models as external targets to enrich current observation representations with static spatial cues (Figure 2(b)). While these foundation representations (Wang et al., 2026a) capture rich multi-view geometry, extending them to dynamic state prediction is hindered by their high dimensionality and structural redundancy (Wang et al., 2024; Zhuo et al., 2026).

We propose **GeoAct**, a spatially aware world action model that jointly predicts future RGB, geometric latent states, and robot actions within a unified generative process (Figure 1(a)). GeoAct decomposes future physical evolution into three complementary streams (Figure 2(c)): RGB latents characterize fine-grained appearance dynamics, geometric latents capture cross-view and temporal spatial structures, and actions describe robot interactions aligned with future scene transitions. For

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.

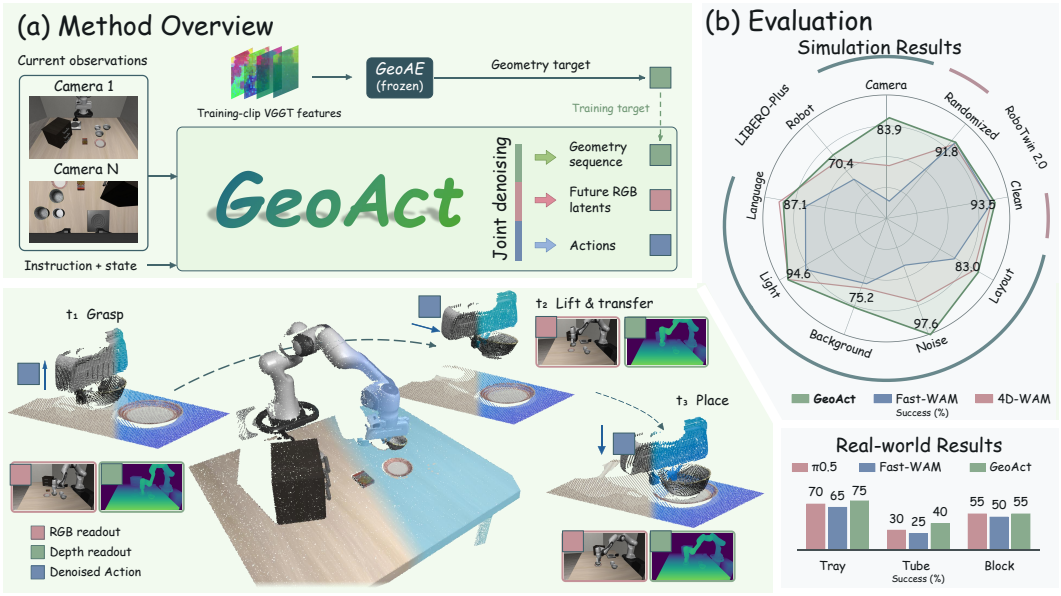

Figure 1: **Overview of GeoAct. (a)** GeoAct jointly predicts future RGB, geometric latent states, and robot actions to learn spatially aware manipulation policies, while requiring only current observations for action inference. **(b)** Results on LIBERO-Plus, RoboTwin 2.0, and real-world bimanual tasks demonstrate improved spatial awareness and manipulation performance.

the geometric stream, we design **GeoAE** (Geometric Autoencoder) to compress multi-layer VGGT-Omega features into a compact spatiotemporal latent space, preserving spatial structure through feature, depth, and camera reconstruction. During joint training, the complete geometric sequence spanning both current and future states is co-denoised alongside future RGB and actions. This allows the policy to execute actions directly from current RGB observations at deployment with minimal latency, requiring neither online 3D extraction nor video rollouts.

We evaluate GeoAct on LIBERO-Plus, RoboTwin 2.0, and real-world bimanual tasks (Figure 1(b)). On LIBERO-Plus, GeoAct achieves a state-of-the-art 84.6% mean success rate, reaching 83.9% under camera viewpoint perturbations and outperforming Fast-WAM by 67.5 percentage points. On RoboTwin 2.0, GeoAct attains a 92.6% mean success rate across 50 diverse bimanual tasks. Real-world evaluations on bimanual tray placement, test-tube insertion, and block handover further demonstrate improved performance on tasks requiring precise spatial coordination.

Our contributions can be summarized as follows:

- We present **GeoAct**, a spatially aware world action model that explicitly models 3D geometric latent states alongside future RGB and actions, unifying visual appearance, spatial evolution, and robot interactions within a joint generative process.
- We design **GeoAE**, a geometric autoencoder that maps high-dimensional 3D foundation representations into a compact spatiotemporal latent space, enabling dynamic geometric prediction while preserving essential 3D scene geometry.
- We demonstrate across extensive simulation benchmarks and real-world bimanual tasks that GeoAct significantly improves spatial awareness and manipulation precision, while retaining efficient RGB-to-action inference.

2 Related Work

2.1 World Action Models

World Action Models (WAMs) jointly learn future scene changes and robot actions, using predictive rollouts to provide rich physical supervision for policy learning (Kim et al., 2026; Hansen et al.,

2024; Li et al., 2026b; Wang et al., 2026c). Motus (Bi et al., 2026) unifies visual understanding, video generation, and action prediction through a multi-expert architecture. Fast-WAM (Yuan et al., 2026) further examines the decoupled roles of future prediction during training and inference, demonstrating that joint video modeling during training strengthens policy representations, whereas inference can bypass explicit future video rollouts.

To enrich future supervision beyond raw RGB video, recent efforts extend to jointly modeling latent dynamics or high-level semantics. For instance, LaWAM (Chen et al., 2026a) models future dynamics and actions in a compact latent space, while ST-WAM (Wang et al., 2026b) jointly predicts VAE latents and DINO features to capture both appearance dynamics and task context. Although these methods enhance visual representation learning, they predict visual representations without explicit supervision of 3D scene structure. We therefore incorporate geometric states into world prediction to describe both appearance dynamics and spatial evolution.

2.2 Geometry-Aware Policies and World Modeling

To enhance the spatial awareness of robot manipulation policies, several recent approaches incorporate explicit geometric priors into future state prediction (Guo et al., 2026; Lopetegui Gonzalez et al., 2026; Liu et al., 2026). For instance, X-WAM employs a dedicated depth branch to predict multi-view RGB-D videos, while SA-WAM encodes depth into latents for joint prediction with RGB and actions. However, per-frame depth reconstruction primarily constrains physical distances to visible surfaces; reconstruction errors alone do not explicitly enforce 3D spatial correspondences across views and time.

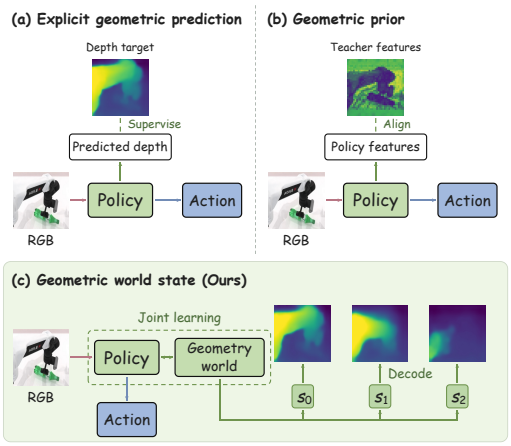

Figure 2: **Geometric modeling for policy learning.** (a) Explicit depth prediction provides geometric supervision. (b) Policy features are aligned with pretrained geometric representations. (c) GeoAct jointly predicts geometric state sequences and robot actions to model evolving spatial structure.

Beyond explicit geometric prediction, another line of research leverages pretrained 3D foundation models to extract multi-view representations of spatial structures and relationships (Wang et al., 2025a; Tian et al., 2026; Wang et al., 2025b). For example, Spatial Forcing aligns intermediate policy representations with these features, providing spatial supervision for action learning. However, precise manipulation requires modeling how spatial relations dynamically evolve under robot actions, yet directly predicting high-dimensional, redundant 3D foundation features remains computationally demanding. We therefore construct a compact, spatially aware geometric latent space and jointly model its states with RGB latents and actions to unify appearance dynamics, 3D spatial evolution, and robot interactions.

3 Method

3.1 Problem Formulation

At control time t, the policy π_θ generates an action chunk of length H from synchronized multi-view RGB observations $\mathbf{o}_t = \{I_t^v\}_{v=1}^V$, proprioception $q_t \in \mathbb{R}^{d_q}$, and a language instruction ℓ:

$$A_t = [a_t, \dots, a_{t+H-1}] \in \mathbb{R}^{H \times d_a}, \qquad \widehat{A}_t \sim \pi_\theta(\cdot \mid \mathbf{o}_t, q_t, \ell). \tag{1}$$

Here, V is the number of views, d_a is the dimension of each action, and A_t and $\widehat{A}_t$ denote demonstrated and predicted action chunks, respectively.

World action models supervise action learning by predicting scene changes (Tian et al., 2025). For a demonstrated action chunk A_t, let $\mathbf{O}_t$ denote the temporally aligned multi-view video clip. To

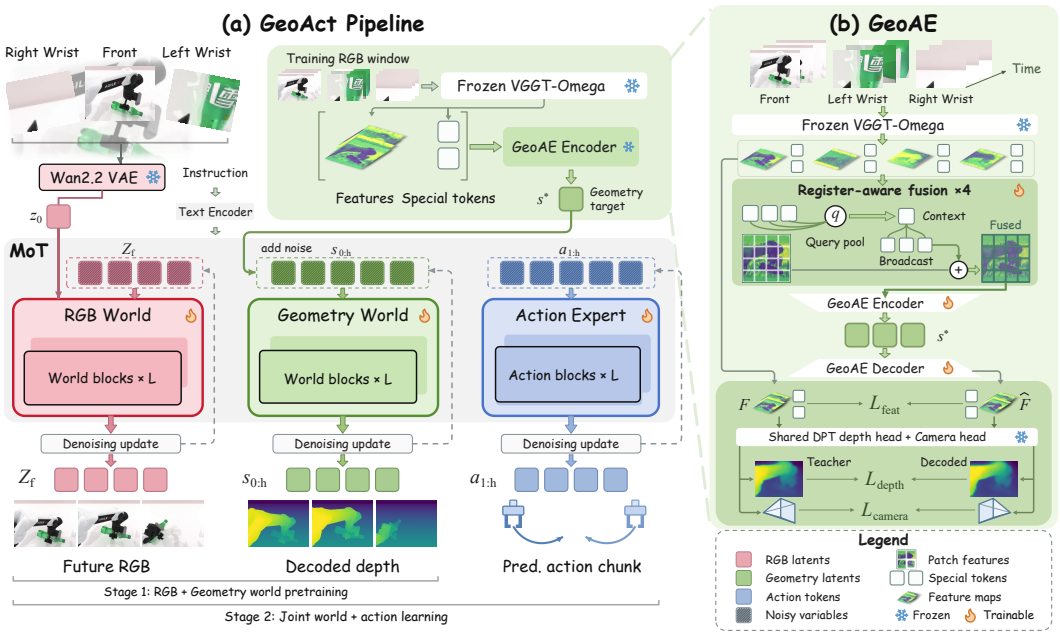

Figure 3: **Overview of GeoAct.** (a) Conditioned on current multi-view RGB, the model jointly denoises future RGB latents, the complete geometric sequence including s_0, and an action chunk. (b) GeoAE compresses multi-layer VGGT-Omega features into geometric latents using feature, depth, and camera reconstruction objectives.

describe appearance changes and spatial structure, we construct two complementary states: the frozen video encoder E_{rgb} extracts RGB latents, while GeoAE, introduced below, encodes the clip's multi-view geometric features Ω_t into compact geometric states:

$$[z_0; Z_f] = E_{\text{rgb}}(\mathbf{O}_t), \qquad S = E_{\text{GeoAE}}(\Omega_t). \tag{2}$$

Here, z_0 and Z_f denote current and future RGB latents, respectively, and S is the complete geometric state sequence. The model jointly predicts Z_f, S, and the action chunk A_t.

3.2 Geometric Autoencoder

We aim to construct a compact latent space that represents scene spatial structure. To this end, we design GeoAE, which encodes multi-view VGGT-Omega geometric features into compact latents and learns this space through joint feature and geometry reconstruction. Frozen VGGT-Omega extracts four layers of geometric features $\{\Omega^{(l)}\}_{l\in\mathcal{L}}$ from a synchronized multi-view clip.

Spatiotemporal encoding. In VGGT-Omega, the camera token encodes camera information, while register tokens (Darcet et al., 2024) aggregate global scene context across views. To preserve these geometric cues while compressing local patch features, we project the special tokens to $R^{(l)} \in \mathbb{R}^{17\times d}$ and pool them into a context vector using a learnable query $u \in \mathbb{R}^d$:

$$c^{(l)} = (R^{(l)})^\top \operatorname{softmax}\left(R^{(l)}u/\sqrt{d}\right). \tag{3}$$

We broadcast $c^{(l)}$ onto the patch grid, concatenate the fused layers along channels, and encode them with a deterministic Wan2.2 (Team Wan et al., 2025) causal spatiotemporal residual encoder. Temporal downsampling and channel compression produce $S = [s_0, s_1, \ldots, s_{T_g-1}]$ while preserving the spatial grid. Views share encoder parameters.

Feature and geometry reconstruction. The decoder restores the original temporal length. Pointwise projection reconstructs patch tokens, while per-frame, per-view global average pooling of the decoder's upsampled intermediate feature map produces $\bar{h}$ for special-token reconstruction with learnable embeddings $e_n^{(l)}$:

$$\hat{r}_n^{(l)} = f_s^{(l)}\left(W_s^{(l)}\bar{h} + e_n^{(l)}\right), \tag{4}$$

where $W_s^{(l)}$ and $f_s^{(l)}$ are the mapping matrix and projection head, respectively, and $e_n^{(l)}$ distinguishes different special tokens reconstructed from the same context. Concatenating the two outputs along the token dimension restores the complete features, which are passed to frozen VGGT-Omega geometric decoding heads to predict depth and camera parameters. GeoAE jointly optimizes feature reconstruction and explicit 3D geometric decoding objectives:

$$\mathcal{L}_{\text{GeoAE}} = \mathcal{L}_{\text{feat}} + \lambda_d \mathcal{L}_{\text{depth}} + \lambda_c \mathcal{L}_{\text{camera}}, \tag{5}$$

where $\mathcal{L}_{\text{feat}}$ combines the L_1 distance and cosine distance of normalized features to supervise reconstruction of all tokens; $\mathcal{L}_{\text{depth}}$ uses teacher-confidence-weighted log-depth errors; and $\mathcal{L}_{\text{camera}}$ constrains camera parameters to match the teacher, with loss weights $\lambda_d = 0.5$ and $\lambda_c = 0.2$. With this multi-task objective, GeoAE compresses redundant foundation features into a compact latent sequence S while preserving 3D spatial structure.

3.3 Geometry-Enhanced World–Action Modeling

Three-stream joint modeling and structured attention. As shown in Figure 3(a), given current observations $\mathbf{o}_t$, proprioception q_t, and an instruction ℓ, we represent future RGB latents Z_f, the compact geometric sequence $S = [s_0, \ldots, s_{T_g-1}]$, and an action chunk A through the conditional joint distribution

$$p_\theta(Z_f, S, A \mid \mathbf{o}_t, q_t, \ell)\,. \tag{6}$$

The RGB, Geometry, and Action experts follow Fast-WAM's Mixture-of-Transformers (MoT) architecture (Yuan et al., 2026). Future RGB and the complete geometric sequence S interact bidirectionally, whereas Action attends only to current RGB $z_0 = E_{\text{rgb}}(\mathbf{o}_t)$ and its own tokens (Figure 4, left), avoiding dependence on future geometry during action generation. Gradients from geometry prediction jointly optimize the RGB expert, providing geometric supervision for current observation representations.

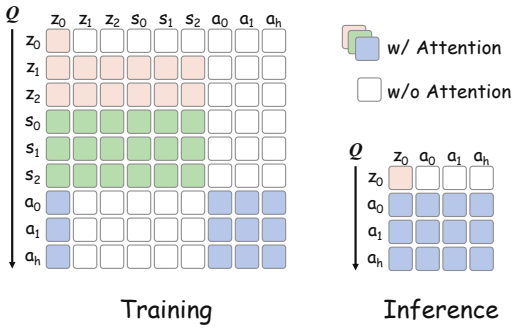

Figure 4: **Structured attention for training and inference.** Rows and columns denote queries and keys. During training, future RGB and the full geometric sequence interact bidirectionally, while action tokens attend only to current RGB and action tokens. Only current RGB and action tokens are processed at inference.

Geometric state prediction and action inference. Extracting 3D geometry online incurs non-negligible computational latency during real-time control. During training, GeoAct uses clean current RGB latents z_0 as its only visual conditioning input and jointly denoises the complete geometric sequence S (including s_0), learning to infer current 3D configurations directly from 2D observations. Consequently, test-time deployment can omit the geometry branch and online 3D extraction, generating actions directly from cached RGB representations with low latency (Figure 4, right). Starting from Gaussian noise $A_{t_0} \sim \mathcal{N}(0, I)$, the Action expert directly generates actions through Euler integration over times $0 = t_0 < \cdots < t_N = 1$, with $k = 0, \ldots, N-1$:

$$A_{t_{k+1}} = A_{t_k} + (t_{k+1} - t_k)\, \widehat{v}_\theta^a(A_{t_k}, t_k \mid z_0, q_t, \ell)\,. \tag{7}$$

3.4 Joint Flow-Matching Training

We train the model using conditional flow matching (Lipman et al., 2023). Let $x^v = Z_f$, $x^g = S$, and $x^a = A$ denote the training targets of the RGB, Geometry, and Action branches, respectively. For branch $r \in \{v, g, a\}$, we sample Gaussian noise $\epsilon^r \sim \mathcal{N}(0, I)$ and interpolate from noise ($t_r = 0$) to data ($t_r = 1$):

$$x_{t_r}^r = (1 - t_r)\epsilon^r + t_r x^r. \tag{8}$$

RGB and Geometry share a flow time $t_v = t_g = t_w$, while Action independently samples t_a. Gaussian noise is independent across branches.

The model predicts the flow velocity $\widehat{v}_\theta^r$ of each branch, with the flow-matching loss

$$\mathcal{L}_r = \mathbb{E}\Big[w_r(t_r)\,\|\widehat{v}_\theta^r - (x^r - \epsilon^r)\|_2^2\Big], \qquad r \in \{v, g, a\}. \tag{9}$$

where $w_r(t_r)$ is the time-dependent weight for that branch. The joint training objective is

$$\mathcal{L}_{\text{WAM}} = \lambda_v \mathcal{L}_v + \lambda_g \mathcal{L}_g + \lambda_a \mathcal{L}_a. \tag{10}$$

We set $\lambda_v = 1$, $\lambda_g = 0.02$, and $\lambda_a = 1$. The video encoder, VGGT-Omega, and GeoAE remain frozen during world action model training.

Training proceeds through GeoAE pretraining for feature and geometry reconstruction (Section 3.2), RGB and geometric world prediction on multi-domain videos (Stage 1), and joint world and action learning from robot demonstrations (Stage 2).

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate GeoAct on LIBERO-Plus (Fei et al., 2026) and RoboTwin 2.0 (Chen et al., 2026b). LIBERO-Plus assesses robustness to changes in visual observations, robot and object configurations, and language instructions. Camera viewpoint perturbations change the visual projection of a scene, whereas changes in robot initialization and object layout alter the relative positions of the robot and manipulation targets. Policies are trained on the Spatial, Object, Goal, and LIBERO-10 suites of LIBERO (Liu et al., 2023) and evaluated on LIBERO-Plus without additional fine-tuning. RoboTwin 2.0 extends the evaluation to 50 bimanual tasks involving coordination of end-effector and object poses within a shared workspace. We train on 27,500 demonstrations and report mean success rates under both Clean and Randomized settings.

Real-robot setup. Our real-world platform consists of two Piper arms and three D435 cameras, with one external camera observing the workspace and one camera mounted on each wrist (Figure 5(a)). Bimanual tray placement assesses tray orientation control and shelf alignment during transport; test-tube insertion assesses relative pose control between the tube and target hole; block handover and stacking assesses coordination of the two end effectors during transfer and placement.

Implementation details. The RGB expert is parameterized by the 5B Wan2.2-TI2V-5B DiT, while the Geometry and Action experts each use an approximately 1B-parameter DiT (Peebles & Xie, 2023). For the main experiments, the RGB and Geometry experts are initialized from Wan2.2 for Stage 1. Stage 2 inherits their pretrained weights and introduces an Action expert initialized from adapted Wan2.2 weights. GeoAE ($\sim$0.33B parameters) remains frozen after pretraining. With inputs resized to 224×224, LIBERO training pairs 32-step action chunks with nine-frame clips sampled at 4-step intervals; inference uses 10 integration steps. For ablation studies, we bypass Stage 1 pretraining and train policies directly on original, unaugmented demonstrations.

Baselines. VLA baselines are π_0 (Black et al., 2025b), $\pi_{0.5}$ (Black et al., 2025a), and StarVLA (StarVLA Community, 2026). Visual future-prediction baselines are LingBot-VA (Li et al., 2026b), Fast-WAM (Yuan et al., 2026), and ST-WAM (Wang et al., 2026b). Geometry-related baselines are X-WAM, 4D-WAM (Yang et al., 2026), and Spatial Forcing.

4.2 Simulation Results

As shown in Table 1, GeoAct achieves a mean success rate of 84.6% on LIBERO-Plus. Spatial Forcing and X-WAM, which incorporate geometric supervision, outperform Fast-WAM under camera viewpoint changes. GeoAct achieves the highest success rate under these perturbations (83.9%). Compared with aligning current visual features to geometry or reconstructing future visible-surface depth, GeoAct organizes cross-view and temporal relationships into compact predictive states, supervising the 3D evolution of robot–object interactions.

On RoboTwin 2.0, GeoAct achieves 93.5% on Clean and 91.8% on Randomized, averaging 92.6% and exceeding LingBot-VA and Fast-WAM by 0.4 and 0.8 percentage points, respectively, demonstrating consistent performance across diverse bimanual tasks.

Table 1: **Success rates (%, ↑) on LIBERO-Plus and RoboTwin 2.0.** Bold denotes the best mean success rate for each benchmark; green highlights our method.

	LIBERO-Plus								RoboTwin 2.0-Full		
Method	**Camera**	**Robot**	**Lang.**	**Light**	**Bkgd.**	**Noise**	**Layout**	**Mean**	**Clean**	**Rand.**	**Mean**
VLA											
π_0	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6	65.9	58.4	62.2
$\pi_{0.5}$	78.4	73.6	80.8	96.2	94.1	89.0	84.5	84.4	82.7	76.8	79.8
StarVLA	52.5	49.8	88.5	95.7	95.7	73.0	76.9	74.1	88.2	88.3	88.3
Spatial Forcing	64.3	38.8	83.5	93.1	91.1	71.7	78.1	72.7	86.7	87.4	87.1
WAM											
LingBot-VA	40.9	83.0	86.4	82.3	53.1	64.4	76.2	69.5	92.9	91.5	92.2
Fast-WAM	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5	91.9	91.8	91.8
4D-WAM	45.2	64.3	90.6	94.3	57.7	69.1	79.2	71.0	91.9	90.8	91.3
X-WAM	66.2	75.2	82.7	91.4	89.8	48.0	85.5	75.6	89.8	90.7	90.2
GeoAct (ours)	83.9	70.4	87.1	94.6	75.2	97.6	83.0	**84.6**	93.5	91.8	**92.6**

4.3 Real-World Bimanual Manipulation

We compare GeoAct, $\pi_{0.5}$ (Black et al., 2025a), and Fast-WAM (Yuan et al., 2026) on bimanual tray placement, test-tube insertion, and block handover and stacking, with 20 trials per policy per task. GeoAct and Fast-WAM train for 10 epochs on the same real-robot demonstrations.

GeoAct achieves a mean success rate of 56.7%, exceeding $\pi_{0.5}$ and Fast-WAM by 5.0 and 10.0 percentage points, respectively (Table 2). Gains are concentrated in bimanual tray placement and test-tube insertion, where predicting future geometric states provides explicit 3D supervision for controlling relative object–target poses during bimanual coordination. On block handover and stacking, GeoAct reaches 55.0%, matching $\pi_{0.5}$.

Table 2: **Real-world bimanual task success rates (%, ↑).** Bold denotes the best mean success rate.

Method	**Bimanual tray placement**	**Test-tube insertion**	**Block handover and stacking**	**Mean**
$\pi_{0.5}$	70.0	30.0	55.0	51.7
Fast-WAM	65.0	25.0	50.0	46.7
GeoAct (ours)	75.0	40.0	55.0	**56.7**

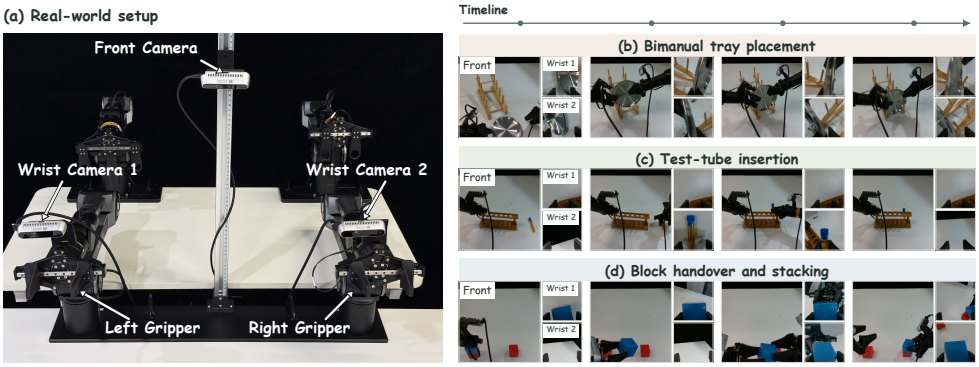

Figure 5: **Real-world setup and multiview task demonstrations.** (a) The dual-arm Piper platform and camera setup. (b–d) Recorded demonstrations of bimanual tray placement, test-tube insertion, and block handover and stacking. Each row shows four key moments from one demonstration along the timeline, with the Front view on the left and the Wrist 1 and Wrist 2 views stacked on the right. The timeline indicates temporal order only.

4.4 Ablation Studies

4.4.1 Effect of Joint RGB–Geometry Prediction

Table 3 compares three configurations that augment action prediction with RGB prediction, geometry prediction, or both as world supervision.

Table 3: **Combinations of world prediction targets.** All variants predict actions alongside RGB, geometry, or both. We report LIBERO-Plus success rates (%, ↑); bold denotes the best mean success rate.

World prediction target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean
RGB only	53.41	62.88	50.07	79.60	69.59	83.31	70.76	66.49
Geometry only	49.84	49.16	35.91	89.32	87.73	89.88	71.74	65.88
RGB + Geometry	68.17	64.52	83.80	92.03	67.19	94.38	77.84	**78.27**

Joint prediction outperforms both single-stream variants in six of seven perturbation categories, with gains spanning changes in viewpoint, object layout, and language. Geometric supervision constrains spatial relationships between the robot and objects, while RGB supervision preserves object appearance; together, they support spatial localization and instruction-conditioned target identification. Background perturbations are the sole exception, with Geometry-only performing better. RGB reconstruction may preserve task-irrelevant background textures, leaving the joint model sensitive to changes in background appearance.

4.4.2 Choice of World Prediction Targets

Table 4 compares geometric prediction targets on LIBERO-Plus. GeoAE latents outperform direct depth and VGGT feature prediction on all four suites, improving mean success rates by 7.34 and 5.11 percentage points, respectively.

Table 4: **Choice of world prediction targets.** LIBERO-Plus success rates (%, ↑). "+" adds the indicated target to joint RGB–action prediction. Bold denotes the best mean success rate.

Prediction target	Spatial	Object	Goal	Long	Mean
RGB only	72.58	75.35	65.72	52.63	66.49
+ DINO (ST-WAM)	75.95	80.70	70.52	66.60	73.40
+ Depth	79.48	80.62	69.09	54.98	70.93
+ VGGT features	80.97	81.73	70.98	59.35	73.15
+ GeoAE latent	86.68	83.48	79.35	63.91	**78.27**

As visualized in Figure 6, direct depth prediction exhibits progressive spatial collapse: arm and cup boundaries become blurred, and local depth variations are smoothed out. Decoding directly predicted VGGT features yields distorted object contours, whereas GeoAE preserves sharper boundaries and their changing spatial relationships during the motion.

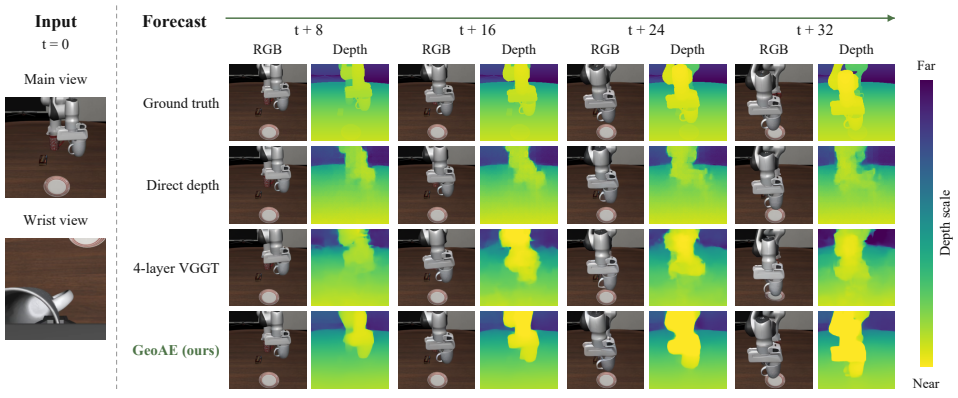

Figure 6: **Future RGB and depth predictions.** We compare three geometric prediction targets on a LIBERO-10 training clip of white-cup placement. The top row shows recorded RGB and simulator depth. Depth predictions use sequence-level scale-and-shift alignment. Direct depth prediction progressively loses arm and cup contours, which are better preserved by GeoAE.

4.4.3 Geometry Preservation under Compression

We evaluate compressed representations through geometric reconstruction and downstream manipulation. Feature preservation is measured using LN-L1, cosine similarity, and linear CKA (Kornblith et al., 2019); depth reconstruction is evaluated using AbsRel and δ_1 (Eigen et al., 2014). Table 5 compares PCA-T3, Feature-only AE trained with feature reconstruction alone, and GeoAE, all sharing the same latent channel and temporal dimensions $(C, T) = (48, 3)$. Depth reconstruction is eval-

uated against outputs obtained by passing uncompressed teacher features through the same frozen depth head.

Table 5: **Comparison of geometric compression methods.** Feature and depth reconstruction across five domains, with LIBERO-Plus success rates. Depth is evaluated against teacher predictions; bold denotes the best mean success rate.

Compressor	(C, T)	**Feature preservation**			**Depth reconstruction**		**LIBERO-Plus**
		LN-L1 ↓	Cosine ↑	CKA ↑	AbsRel ↓	δ_1 (%) ↑	Mean (%) ↑
PCA-T3	$(48, 3)$	0.4715	0.7376	0.6101	0.18452	64.55	73.76
Feature-only AE	$(48, 3)$	0.3645	0.8213	0.7451	0.12048	78.97	75.05
GeoAE	$(48, 3)$	0.3719	0.8258	0.7550	0.04216	96.51	**78.40**

GeoAE has comparable feature reconstruction accuracy to Feature-only AE, while its depth AbsRel is 65.0% lower and δ_1 is 17.54 percentage points higher under the same frozen depth head. This demonstrates that high feature similarity does not guarantee geometric recoverability and highlights the benefit of explicit geometric reconstruction losses.

As the geometric prediction target, GeoAE achieves 78.40% on LIBERO-Plus, exceeding Feature-only AE and PCA-T3 by 3.35 and 4.64 percentage points, respectively.

4.5 Inference Efficiency

Closed-loop control requires accurate manipulation with low latency. Figure 7 compares inference configurations: future-video generation increases Fast-WAM-Joint's latency to 510.3 ms, while the s_0-conditioned variant of GeoAct has a total inference latency of 281.8 ms when current geometry is encoded online through VGGT-Omega and GeoAE.

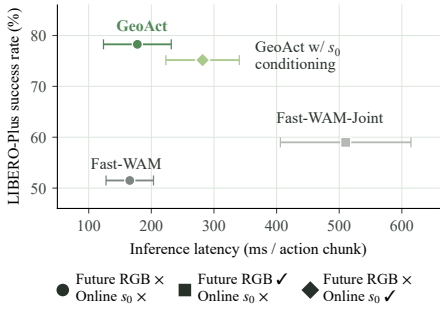

Figure 7: **Inference configurations and efficiency on LIBERO-Plus.** Marker shapes indicate whether future RGB is generated alongside actions and whether s_0 is encoded online with VGGT-Omega and GeoAE. Latency is measured on one A800 GPU for a 32-step action chunk with 10 integration steps; horizontal error bars show the sample standard deviation.

GeoAct omits both computations at deployment, generating action chunks in 177.7 ms, close to Fast-WAM's 165.6 ms, while attaining 78.27% success versus 51.50% for Fast-WAM (Yuan et al., 2026) and 75.16% for the s_0-conditioned variant. Unlike frame-wise DINO features, VGGT-Omega features depend on the jointly encoded frames. The s_0-conditioned variant separately encodes current observations to match training and inference inputs, while its future targets are obtained from full clips. The model must therefore account for both scene evolution and changes in encoding context, which may complicate geometric prediction; GeoAct instead jointly predicts a complete sequence encoded from the same clip. GeoAct thus eliminates test-time geometric overhead, enabling high manipulation precision through direct, low-latency execution.

5 Conclusion

GeoAct learns spatially aware manipulation by jointly predicting geometric latent states, future RGB, and robot actions. GeoAE enables this prediction by compressing 3D foundation features while preserving spatial structure. Experiments and ablations show that geometric prediction complements appearance supervision, improving performance under viewpoint changes and in tasks requiring precise coordination. Together, these results establish compact geometric states as effective world prediction targets for policy learning, with GeoAct retaining efficient RGB-to-action inference without online geometry encoding or future video generation.

Reproducibility statement

Section 3 and Appendix B describe GeoAE and GeoAct, their training objectives, data preprocessing, initialization, and training schedules. LIBERO-Plus evaluations use matched initial states and environment seeds; RoboTwin 2.0 averages the 50 tasks equally within Clean and Randomized settings; and real-robot policies are evaluated from matched initial scenes in 20 trials per task. Appendices C–E specify success criteria, per-task results, and ablation settings.

Ethics statement

All authors have read and agree to abide by the ICLR Code of Ethics. This work and all related conference activities, including submission, reviewing, and discussion, are conducted in accordance with its principles and requirements.

AI use statement

ChatGPT and Codex were used to assist with language polishing during the preparation of this manuscript. Their use included correcting grammatical errors, improving sentence structure and readability, refining wording to follow academic writing conventions, and ensuring consistent use of technical terminology. The authors reviewed and revised all AI-assisted text to ensure that it accurately conveyed the intended scientific meaning and remained consistent with the methods and experimental results. The authors take full responsibility for the final content of the paper.

References

Shuanghao Bai, Wenxuan Song, Jiayi Chen, et al. Embodied Robot Manipulation in the Era of Foundation Models: Planning and Learning Perspectives. *IEEE Transactions on Robotics*, pp. 1–21, 2026.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, et al. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35101–35113. Computer Vision Foundation, 2026.

Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems (RSS)*, pp. 1–17, 2025b. doi: 10.15607/RSS.2025.XXI.010.

Jialei Chen, Kai Wang, Kang Chen, et al. LaWAM: Latent World Action Models for Efficient Dynamics-Aware Robot Policies. arXiv preprint arXiv:2606.15768, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, et al. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1–24, 2026b.

Hyung Won Chung, Xavier Garcia, Adam Roberts, et al. UniMax: Fairer and More Effective Language Sampling for Large-Scale Multilingual Pretraining. In *International Conference on Learning Representations (ICLR)*, pp. 1–19, 2023.

Timothée Darcet, Maxime Oquab, Julien Mairal, et al. Vision Transformers Need Registers. In *International Conference on Learning Representations (ICLR)*, pp. 2632–2652, 2024.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pp. 2366–2374. Curran Associates, Inc., 2014.

Senyu Fei, Siyin Wang, Junhao Shi, et al. LIBERO-Plus: A Progressive Robustness Benchmark for Visual-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, 2026.

Chelsea Finn and Sergey Levine. Deep Visual Foresight for Planning Robot Motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793, 2017. doi: 10.1109/ICRA.2017.7989324.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5842–5850, 2017.

Jun Guo, Qiwei Li, Peiyan Li, et al. Unified 4D World Action Modeling from Video Priors with Asynchronous Denoising. arXiv preprint arXiv:2604.26694, 2026.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *International Conference on Learning Representations (ICLR)*, pp. 3534–3563, 2024.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy Diffusion with 3D Scene Representations. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1949–1974. PMLR, 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, et al. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning. In *International Conference on Learning Representations (ICLR)*, pp. 71531–71552, 2026.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, et al. Similarity of Neural Network Representations Revisited. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

Fuhao Li, Wenxuan Song, Han Zhao, et al. Spatial Forcing: Implicit Spatial Representation Alignment for Vision-Language-Action Model. In *International Conference on Learning Representations (ICLR)*, pp. 132324–132345, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, et al. Causal World Modeling for Robot Control. In *Proceedings of Robotics: Science and Systems (RSS)*, pp. 1–23, Sydney, Australia, July 2026b. doi: 10.15607/RSS.2026.XXII.016.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, et al. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*, pp. 1–28, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, et al. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939.

Zeyi Liu, Shuang Li, Eric Cousineau, et al. Geometry-Aware 4D Video Generation for Robot Manipulation. In *International Conference on Learning Representations (ICLR)*, pp. 140751–140773, 2026.

Javier Alejandro Lopetegui Gonzalez, Paul Pacaud, and Cordelia Schmid. Spatially Aware World Action Model via Geometric Latent Diffusion. arXiv preprint arXiv:2609.02531, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, pp. 1–19, 2019.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

StarVLA Community. StarVLA: A Lego-Like Codebase for Vision-Language-Action Model Developing. arXiv preprint arXiv:2604.05014, 2026.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, et al. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063.

Team Wan, Ang Wang, Baole Ai, et al. Wan: Open and Advanced Large-Scale Video Generative Models. arXiv preprint arXiv:2503.20314, 2025.

Yang Tian, Sizhe Yang, Jia Zeng, et al. Predictive Inverse Dynamics Models Are Scalable Learners for Robotic Manipulation. In *International Conference on Learning Representations (ICLR)*, pp. 92033–92052, 2025.

Yinchen Tian, Huan Li, Muyao Peng, et al. MV-Actor: Aligning Multi-View Semantics and Spatial Awareness for Bimanual Manipulation. arXiv preprint arXiv:2606.10899, 2026.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, et al. BridgeData V2: A Dataset for Robot Learning at Scale. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, et al. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, et al. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21486–21499, 2026a.

Mingxin Wang, Bin Hu, Bin Qian, et al. ST-WAM: Semantic-Temporal World Action Model for Robust Manipulation under Visual Distribution Shifts. arXiv preprint arXiv:2607.28993, 2026b.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, et al. Continuous 3D Perception Model with Persistent State. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, et al. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2024.

Yuqi Wang, Xinghang Li, Wenxuan Wang, et al. Unified Vision-Language-Action Model. In *International Conference on Learning Representations (ICLR)*, pp. 80929–80944, 2026c.

WorldArena. WorldArena 2.0: Track 3 Real-Robot Manipulation. Dataset and benchmark documentation, 2026.

Hongtao Wu, Ya Jing, Chilam Cheang, et al. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. In *International Conference on Learning Representations (ICLR)*, pp. 10641–10662, 2024.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and Improving Layer Normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 4381–4391. Curran Associates, Inc., 2019.

Lishan Yang, Wenxuan Song, Xi Wang, et al. 4D-WAM: Infusing Spatiotemporal Awareness into World Action Models through Trajectory Fields. arXiv preprint arXiv:2608.08023, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, et al. Fast-WAM: Do World Action Models Need Test-Time Future Imagination? arXiv preprint arXiv:2603.16666, 2026.

Zhengshen Zhang, Hao Li, Yalun Dai, et al. From Spatial to Actions: Grounding Vision-Language-Action Model in Spatial Foundation Priors. In *International Conference on Learning Representations (ICLR)*, pp. 35936–35962, 2026.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, et al. 3D-VLA: A 3D Vision-Language-Action Generative World Model. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61229–61245. PMLR, 2024.

Haoyi Zhu, Honghui Yang, Yating Wang, et al. SPA: 3D Spatial-Awareness Enables Effective Embodied Representation. In *International Conference on Learning Representations (ICLR)*, pp. 26361–26391, 2025.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, et al. Streaming Visual Geometry Transformer. In *International Conference on Learning Representations (ICLR)*, pp. 88055–88072, 2026.

SUPPLEMENTARY MATERIAL

Section A presents the theoretical analysis, and Section B details the data, architecture, and three-stage training. The experimental supplements follow the order of the main paper: simulation benchmarks (Section C), real-robot experiments (Section D), and ablation studies (Section E), including combinations and choices of world prediction targets, geometric compression, and inference configurations and efficiency.

A THEORETICAL ANALYSIS

We provide a theoretical justification for GeoAE's joint reconstruction objective: lower feature reconstruction error does not imply lower geometric distortion, and explicit depth supervision tightens the worst-case geometric distortion bound.

Let $F \in \mathbb{R}^d$ collect the teacher features and $e = \widehat{F} - F$ be the reconstruction residual. Let $\mathcal{N}$ stack the per-token LayerNorm features with the feature-loss averaging factors absorbed, g be the frozen depth decoder with positive outputs, and W be the diagonal matrix of fixed confidence weights and depth-loss averaging factors. Define $A = J_{\mathcal{N}}(F)$ and $B = W J_{\log \circ g}(F)$, with $a(e) = \|Ae\|_1$ and $b(e) = \|Be\|_1$, where J denotes the Jacobian. Assume nonzero normalized tokens and twice continuously differentiable mappings in a neighborhood of F. The implemented feature and depth losses satisfy $\ell_f(e) = a(e) + O(\|e\|_2^2)$ and $\ell_d(e) = b(e) + O(\|e\|_2^2)$; the cosine term in ℓ_f contributes only at second order around exact reconstruction.

Theorem A.1 (Feature–geometry ordering mismatch)**.** *Suppose there exist directions u and v with $a(u) = a(v) = 1$ and $b(u) > b(v)$. Then there exist arbitrarily small reconstruction residuals e_A and e_B with lower feature reconstruction error, $\ell_f(e_A) < \ell_f(e_B)$, but higher depth reconstruction error, $\ell_d(e_A) > \ell_d(e_B)$.*

Proof. Choose any c satisfying $b(v)/b(u) < c < 1$ and set $e_A = \eta c u$ and $e_B = \eta v$, with $\eta > 0$. Positive homogeneity of the two seminorms gives

$$a(e_A) = \eta c < \eta = a(e_B), \qquad b(e_A) = \eta c b(u) > \eta b(v) = b(e_B). \tag{A1}$$

For sufficiently small η, both residuals are arbitrarily close to zero and the strict first-order margins dominate the $O(\eta^2)$ remainders in ℓ_f and ℓ_d, proving the same ordering for the implemented losses. □

We now compare the worst-case depth distortion at a fixed reconstruction-loss budget in the homogeneous first-order model. Define $\kappa = \sup_{a(e)\le 1} b(e) \in [0, \infty]$. For $0 < \kappa < \infty$, κ is the maximal ratio of depth to feature distortion.

Theorem A.2 (Depth supervision tightens the geometric distortion bound)**.** *Let $\lambda_d > 0$ and $\varepsilon > 0$. If $0 < \kappa < \infty$, then*

$$\sup_{a(e)+\lambda_d b(e)\le\varepsilon} b(e) = \frac{\kappa\varepsilon}{1+\lambda_d\kappa} < \kappa\varepsilon = \sup_{a(e)\le\varepsilon} b(e). \tag{A2}$$

Thus, depth supervision strictly reduces the worst-case geometric distortion bound by a factor of $1/(1+\lambda_d\kappa)$ at the same loss budget. If $\kappa = \infty$, the feature-only supremum is infinite, while the joint supremum is ε/λ_d. If $\kappa = 0$, both are 0.

Proof. Positive homogeneity gives the feature-only supremum $\kappa\varepsilon$. For $0 < \kappa < \infty$, $b(e) \le \kappa a(e)$ and the joint constraint imply $b(e) \le \kappa\varepsilon/(1+\lambda_d\kappa)$. To prove tightness, choose $a(u_n) > 0$ with $r_n = b(u_n)/a(u_n) \to \kappa$ and normalize $e_n = \varepsilon u_n/(a(u_n) + \lambda_d b(u_n))$. These residuals saturate the joint budget and satisfy

$$\frac{\varepsilon r_n}{1+\lambda_d r_n} = b(e_n) \le \sup_{a(e)+\lambda_d b(e)\le\varepsilon} b(e) \le \frac{\kappa\varepsilon}{1+\lambda_d\kappa}. \tag{A3}$$

Passing to the limit proves equality. When $\kappa = \infty$, take $a(u_n) \le 1$ with $b(u_n) \to \infty$; the same normalization yields the joint supremum ε/λ_d. When $\kappa = 0$, $b(e)$ is identically 0. □

Adding the nonnegative camera term only shrinks the feasible set, so the depth bound remains valid for the full first-order objective. These results justify explicit depth supervision in GeoAE: feature reconstruction accuracy does not determine geometric fidelity, whereas the depth objective directly constrains geometric distortion.

B Implementation Details and Three-Stage Training

GeoAct is trained through GeoAE pretraining, RGB–geometry world pretraining (Stage 1), and world–action adaptation (Stage 2). Table A1 summarizes the training settings.

B.1 Training Data and Preprocessing

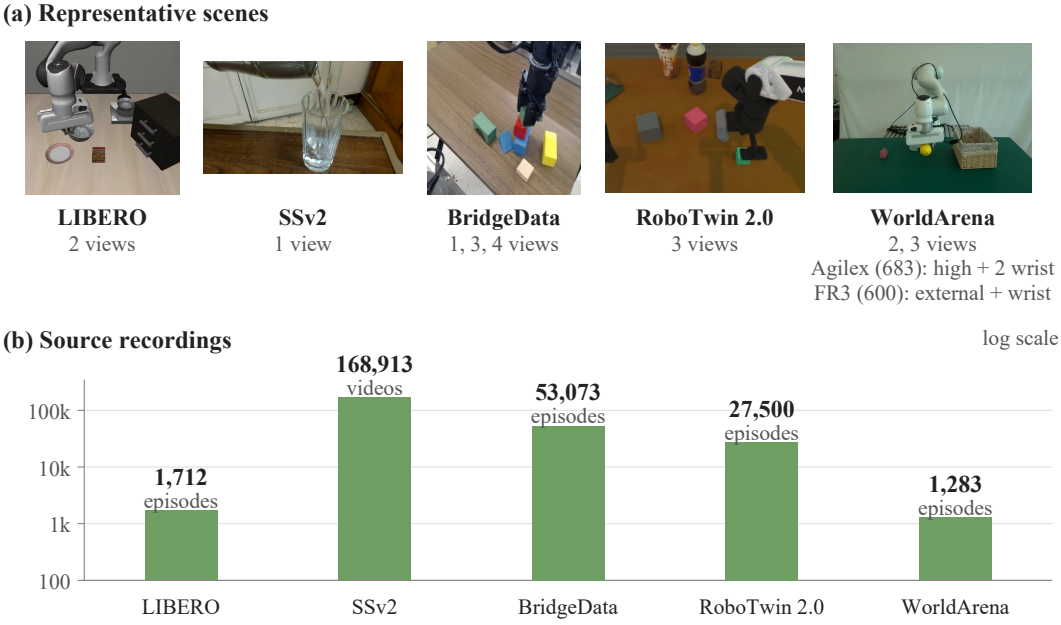

Figure A1: **Training datasets.** Representative scenes, available view counts, and source dataset sizes across five domains. Data sizes count robot episodes or SSv2 videos before the validation split.

GeoAE and Stage 1 use LIBERO (Liu et al., 2023), Something-Something V2 (Goyal et al., 2017), BridgeData V2 (Walke et al., 2023), RoboTwin 2.0 (Chen et al., 2026b), and WorldArena 2.0 (WorldArena, 2026), with equal sampling across domains. Figure A1 summarizes their sizes and available views.

We sample nine-frame clips at 5 FPS, retaining up to four camera views at corresponding times. Each image is resized to 224×224. Approximately 1% of recordings are held out for validation, with all views of a recording assigned to the same split. The teacher jointly encodes synchronized views and video frames to extract cross-view geometric features.

B.2 Model Architectures

Table A2 compares the architectures of Wan2.2-VAE and GeoAE. GeoAE preserves the 14×14 spatial grid while temporally compressing nine frames into a $48 \times 3 \times 14 \times 14$ latent per view.

GeoAE fuses patch and special-token information before compression. A learned query pools the special tokens into a context vector s, which is fused with patch features P as $(P + \mathbf{1}s^\top)/\sqrt{n}$ to stabilize activation scale, where n is the number of fused feature components. The projected maps from four teacher layers are concatenated and encoded. Separate heads reconstruct patch and special tokens; the complete features are then passed to the frozen depth and camera heads.

Table A1: **Three-stage training configurations.** GeoAE pretraining, RGB–geometry world pretraining (Stage 1), and world–action adaptation (Stage 2).

Configuration	GeoAE pretraining	Stage 1	Stage 2
Training data	Five-domain mixture	Five-domain mixture	Four LIBERO suites
Image resolution	224×224	224×224	224×224
Video frames per window	9	9	9
Action chunk length	—	—	32
Training duration	50,000 iterations	20,000 iterations	10 epochs
Global batch size	16–24	32	128
Microbatch per GPU	1	1	1
Gradient accumulation	1	2	4 (32 GPUs) / 8 (16 GPUs)
Number of GPUs	16–24	16	32 or 16
Distributed optimizer	—	—	DeepSpeed ZeRO-1
Image augmentation	Disabled	Disabled	Enabled
Time-shift factor ρ (world / action)	—	5 / —	5 / 5
Window stride	Random video window	Random video window	1
Optimizer	AdamW (Loshchilov & Hutter, 2019)		
Base learning rate / weight decay	10^{-4} / 10^{-2}		
LR schedule / precision	Cosine / BF16		
GeoAE warm-up / final LR	500 / 10^{-5}	—	—
Gradient clipping	Global norm 1.0		
Feature / depth / camera	1.0 / 0.5 / 0.2	—	—
RGB / geometry / action	—	1.0 / 0.02 / 0	1.0 / 0.02 / 1.0

World and action experts. The RGB expert uses 30 Transformer layers with width 3072; the geometry and action experts each use 30 layers with width 1024. All experts project queries, keys, and values to a shared attention space of 24 heads with 128 dimensions per head, while retaining separate input and output projections. World tokens use 3D RoPE over time, height, and width, together with learned view embeddings for up to four cameras. Action tokens use 1D RoPE over their positions in the action chunk (Su et al., 2024). The frozen Wan T5 (Chung et al., 2023) encoder supplies 4096-dimensional language features with a maximum context length of 128 tokens.

Multiview positional encoding. RGB-only shares the RGB–Action MoT backbone and the joint prediction objective for future RGB and actions with Fast-WAM, but the two models represent multiview observations differently. Fast-WAM concatenates the two camera images horizontally in pixel space, encodes them with the VAE, and applies 3D RoPE over the resulting latent grid. RGB-only instead uses the multiview representation of GeoAct. Each view is encoded separately by the shared VAE, with spatiotemporal positions defined within each view and learned view embeddings identifying the cameras.

Let $v \in \{1, 2\}$ denote the camera index, n the latent temporal position, (y, x) the spatial token index within a view, and W_p the width of its token grid. The positions used for RoPE are

$$p^v_{\text{Fast-WAM}} = (n, y, x + (v-1)W_p), \qquad p^v_{\text{ours}} = (n, y, x). \tag{A4}$$

Each LIBERO view has 7×7 spatial tokens, so $W_p = 7$. Figure A2 compares the position indices for the same pair of camera observations.

RoPE determines relative rotations from differences in position. Image concatenation adds $(v' - v)W_p$ to the relative width index between views. This offset depends on image placement rather than the spatial relationship between the cameras. Defining indices within each view removes this offset, while the view embedding e_v represents camera identity. Each observation thus retains its own image coordinate system. Even with the same expert backbone and prediction targets, the multiview representation changes the positional prior used to relate observations from different cameras.

Table A3 summarizes the input organization, positional encoding, and expert specifications. Both models use 294 tokens for the complete RGB sequence and 98 tokens for the current observations. RGB-only encodes the views separately with shared VAE and RGB expert parameters, while retaining the same token counts and expert widths.

Table A2: **Architecture of Wan2.2-VAE and GeoAE.** Per-view grid and token shapes follow (C, T, H, W) and (T, N, D), respectively. RB denotes a residual block; resampling factors follow (T, H, W), and arrows indicate output shapes.

Component	Wan2.2-VAE (RGB)	GeoAE
Input and feature adaptation		
Input	RGB: $(3, 9, 224, 224)$	Four teacher layers, each $9 \times 213 \times 2048$; $213 = 196 + 17$ tokens.
Input adapter	2×2 spatial patchify: $(12, 9, 112, 112)$; causal 3^3 conv $12 \to 160$.	Per-layer LN and $2048 \to 256$ projection; special-token pooling and broadcast fusion; concatenate four maps; 1^3 conv $1024 \to 160$.
Encoder and latent bottleneck		
Encoder stage 1	2 RBs, $160 \to 160$; $(1, 2, 2)$ downsample; $\to (160, 9, 56, 56)$.	2 RBs, $160 \to 160$; no resampling; $\to (160, 9, 14, 14)$.
Encoder stage 2	2 RBs, $160 \to 320$; $(2, 2, 2)$ downsample; $\to (320, 5, 28, 28)$.	2 RBs, $160 \to 320$; $(2, 1, 1)$ stride; $\to (320, 5, 14, 14)$.
Encoder stage 3	2 RBs, $320 \to 640$; $(2, 2, 2)$ downsample; $\to (640, 3, 14, 14)$.	2 RBs, $320 \to 640$; $(2, 1, 1)$ stride; $\to (640, 3, 14, 14)$.
Encoder stage 4	2 RBs, $640 \to 640$; no resampling; $\to (640, 3, 14, 14)$.	2 RBs, $640 \to 640$; no resampling; $\to (640, 3, 14, 14)$.
Encoder middle	RB–single-head spatial attention–RB, 640 channels.	RB–single-head spatial attention–RB, 640 channels.
Latent head	RMSNorm–SiLU– 3^3 conv $640 \to 96$; 1^3 conv $96 \to 96$; split into μ and $\log \sigma^2$.	RMSNorm–SiLU– 3^3 conv $640 \to 48$.
Latent	48 channels; $(48, 3, 14, 14)$. Normalized posterior mean.	Deterministic code: $(48, 3, 14, 14)$; no posterior sampling or KL term.
Decoder and reconstruction heads		
Decoder stem	1^3 conv $48 \to 48$; 3^3 conv $48 \to 1024$.	1^3 conv $48 \to 640$.
Decoder middle	RB–single-head spatial attention–RB, 1024 channels.	RB–single-head spatial attention–RB, 640 channels.
Decoder stage 1	3 RBs, $1024 \to 1024$; $(2, 2, 2)$ upsample; $\to (1024, 5, 28, 28)$.	3 RBs, $640 \to 640$; temporal $3 \to 5$; $\to (640, 5, 14, 14)$.
Decoder stage 2	3 RBs, $1024 \to 1024$; $(2, 2, 2)$ upsample; $\to (1024, 9, 56, 56)$.	3 RBs, $640 \to 320$; temporal $5 \to 9$; $\to (320, 9, 14, 14)$.
Decoder stage 3	3 RBs, $1024 \to 512$; $(1, 2, 2)$ upsample; $\to (512, 9, 112, 112)$.	3 RBs, $320 \to 160$; no resampling; $\to (160, 9, 14, 14)$.
Decoder stage 4	3 RBs, $512 \to 256$; no resampling; $\to (256, 9, 112, 112)$.	3 RBs, $160 \to 160$; no resampling; $\to (160, 9, 14, 14)$.
Output head	RMSNorm–SiLU– 3^3 conv $256 \to 12$; 2×2 unpatchify $\to (3, 9, 224, 224)$.	RMSNorm–SiLU– 3^3 conv $160 \to 160$; four patch heads and four special-token heads.
Reconstruction	RGB pixels.	Reconstructed tokens: $9 \times (196 + 17) \times 2048$ per teacher layer. Frozen VGGT-Omega heads decode depth maps and predict camera parameters.

B.3 GeoAE Pretraining and Reconstruction

GeoAE learns to reconstruct teacher features and their geometric readouts:

$$\mathcal{L}_{\text{GeoAE}} = \mathcal{L}_{\text{feature}} + \lambda_d \mathcal{L}_{\text{depth}} + \lambda_c \mathcal{L}_{\text{camera}}. \tag{A5}$$

The feature loss combines L1 and cosine discrepancy after per-token LayerNorm. The depth loss is a confidence-weighted absolute log-depth error; the camera loss combines rotation-angle, translation, and normalized-intrinsic errors. Teacher features and geometric readouts supply the targets, while gradients through the frozen geometric heads train the autoencoder. The loss weights are given in Table A1.

B.3.1 Qualitative reconstruction

Figure A3 shows depth reconstruction across the five data domains, pairing RGB observations with teacher and reconstructed depth.

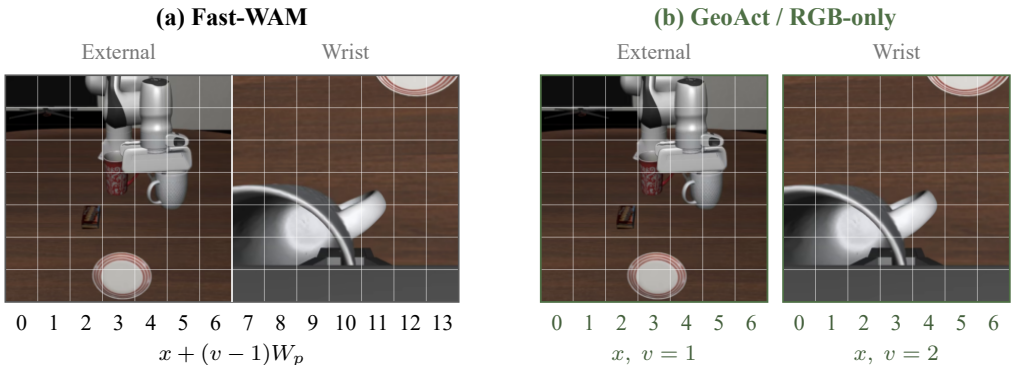

Figure A2: **Multiview positional encoding.** (a) Fast-WAM uses consecutive width indices across the concatenated images. (b) GeoAct and RGB-only index each view independently. Both panels show the same external and wrist RGB observations at one time step. Grids illustrate the spatial partition of the latents; the numbers below give the width indices used for RoPE.

Table A3: **Two-view input and fusion specifications on LIBERO.** Batch axes are omitted; video and latent shapes follow (C, T, H, W). The view embeddings in RGB-only support up to four cameras, of which LIBERO uses two.

Configuration	Fast-WAM	RGB-only
Input and positional encoding		
Video input	$(3, 9, 224, 448)$	$2 \times (3, 9, 224, 224)$
VAE encoding	Encode concatenated images	Encode each view; shared weights
RGB latents	$(48, 3, 14, 28)$	$2 \times (48, 3, 14, 14)$
Latent patch size	$(1, 2, 2)$	$(1, 2, 2)$
Current RGB tokens	$7 \times 14 = 98$	$2 \times 7 \times 7 = 98$
Complete RGB tokens	$3 \times 7 \times 14 = 294$	$2 \times 3 \times 7 \times 7 = 294$
3D RoPE indices	$(n, y, x + (v-1)W_p)$	(n, y, x)
Learned view embeddings	None	4×3072
Experts and cross-view fusion		
RGB / action width	3072 / 1024	3072 / 1024
Layers per expert	30	30
Heads / head dimension	24 / 128	24 / 128
Cross-view fusion	Joint computation over tokens of concatenated images	Joint computation over Q/K/V from all views
Action chunk length	32	32
Action K/V length	$98 + 32 = 130$	$2 \times 49 + 32 = 130$

B.3.2 Reconstruction evaluation

We evaluate 40 validation windows, eight per domain, covering 36 recordings and 783 valid camera observations. All methods receive the same nine-frame, multiview VGGT-Omega features from layers 4, 11, 17, and 23. We compare Feature-only AE with GeoAE using the same inputs and frozen geometric readout heads. The reconstruction results are reported in Table 5 of the main paper.

PCA-T3 uses rank-48 bases fitted on 80 training windows, 16 per domain. Temporal averaging reduces nine positions to three, and interpolation restores the original sequence length. PCA-T3 uses separate bases for patch and special tokens, requiring 30,672 latent scalars per view versus 28,224 for GeoAE. Feature-only AE uses the geometry encoder, decoder, and latent shape in Table A2, but omits depth and camera reconstruction losses.

Feature preservation. LN-L1 measures average absolute error after per-token LayerNorm (Xu et al., 2019); cosine similarity compares the original and reconstructed feature directions. Spatial linear CKA (Kornblith et al., 2019) measures preservation of inter-patch relationships. For channel-centered patch matrices X and Y from normalized reference and reconstructed features, let $K =$

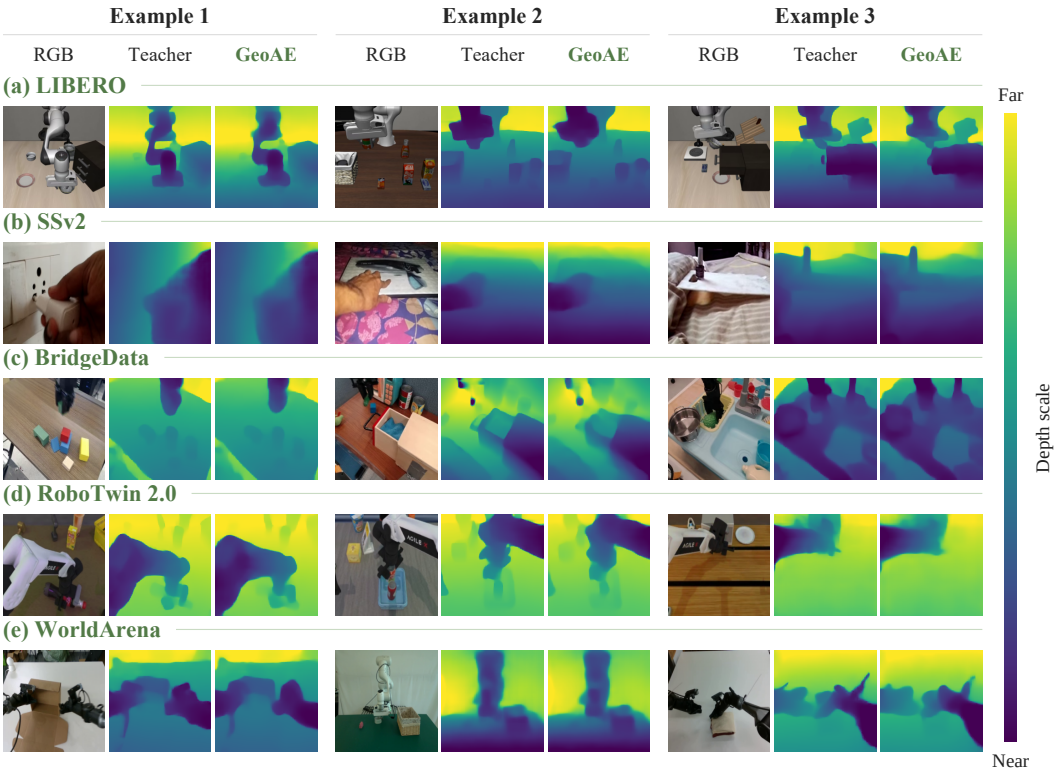

Figure A3: **Depth reconstruction across five domains.** GeoAE reconstructs depth for three held-out recordings per domain. RGB observations and VGGT-Omega teacher depth provide the references.

$XX^\top$ and $L = YY^\top$. We compute

$$\mathrm{CKA}(X, Y) = \frac{\langle K, L \rangle_F}{\|K\|_F \|L\|_F}. \tag{A6}$$

Feature metrics are averaged over valid frames and views, then equally over the four layers and 40 windows.

Depth reconstruction. The same frozen depth head decodes both reconstructed and uncompressed features, with the latter providing the reference. AbsRel and δ_1 (Eigen et al., 2014) measure depth error and accuracy, respectively: $\mathrm{AbsRel} = M^{-1} \sum_i |\widehat{d}_i - d_i|/d_i$, where d_i and $\widehat{d}_i$ are reference and reconstructed depth at the M valid pixels; δ_1 is the fraction satisfying $\max(\widehat{d}_i/d_i, d_i/\widehat{d}_i) < 1.25$. We refer to this protocol as *unaligned native-scale evaluation*, where depth predictions are assessed directly in the teacher model's coordinate frame without sequence-level normalization. For qualitative comparison with simulator depth, we use *simulator-aligned visualization*, fitting one scale and shift per method over the full predicted sequence and sharing the color scale across methods. The former measures reconstruction of the teacher's depth; the latter displays predicted scene structure in the simulator's depth coordinates.

B.4 Stage 1: RGB–Geometry World Pretraining

Stage 1 learns RGB and geometry dynamics before action supervision. The RGB expert is initialized from Wan2.2-TI2V-5B, and the geometry expert uses adapted Wan weights. GeoAE, the RGB autoencoder, and the language encoder remain frozen during world pretraining and policy adaptation. VGGT-Omega uses its 1B checkpoint and receives the synchronized 224×224 views. Geometry latents are standardized channel-wise.

B.4.1 MULTIVIEW GEOMETRIC REPRESENTATIONS

Figure A4 visualizes multiview geometric representations through GeoAE reconstruction, pairing teacher and reconstructed depth across cameras and time.

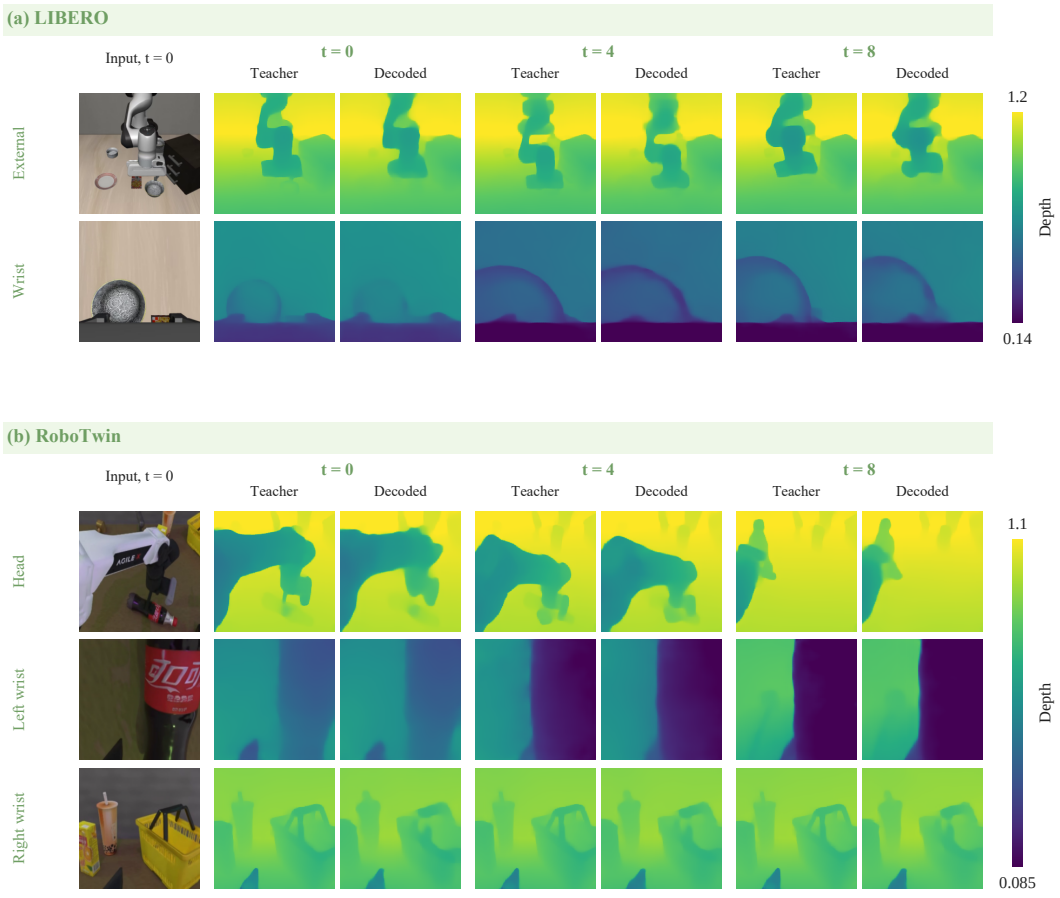

Figure A4: **Multiview depth reconstruction.** GeoAE reconstructs teacher features, which are decoded by the frozen depth head, with teacher depth as the reference. (a) LIBERO external and wrist views. (b) RoboTwin external and two wrist views.

Each clip yields RGB slots $z_{0:2}$ and geometry slots $s_{0:2}$. The current RGB slot z_0 and instruction provide the conditions; future RGB slots $z_{1:2}$ and the complete geometry sequence $s_{0:2}$ are corrupted and jointly predicted. Geometry targets are extracted from the full synchronized clip, including its initial temporal slot. Training combines the RGB and geometry losses with the weights in Table A1. Both world streams share a flow time $t_w = u/[\rho - (\rho - 1)u]$, where $u \sim \mathcal{U}(0, 1)$ and the time-shift factor is $\rho = 5$. Flow-matching losses use a bell-shaped temporal weight proportional to $\exp[-2(t_w - 1/2)^2] - \exp(-1/2)$, normalized to unit mean over the shifted training-time grid.

B.5 STAGE 2: WORLD–ACTION ADAPTATION

The LIBERO main experiments use the image augmentation settings in Table A4. Parameters are sampled independently for each camera and shared across its video window. RGB and geometric targets are extracted from the same augmented video. LIBERO ablations omit Stage 1 world-model pretraining and proceed directly to Stage 2 on the original demonstrations without image augmentation.

Stage 2 adapts the world model to LIBERO Spatial, Object, Goal, and LIBERO-10. Each demonstration window contains 33 observations and 32 actions, with nine evenly spaced video frames compressed into three latent slots. The policy observes front and wrist RGB, the instruction, and proprioception. The seven-dimensional actions and eight-dimensional proprioceptive states (3D

Table A4: **Image augmentation settings for LIBERO.**

Parameter	Setting	Parameter	Setting
Random crop area	90% of the original image	Brightness factor	$[0.7, 1.3]$
Crop aspect ratio	$1:1$	Contrast factor	$[0.6, 1.4]$
Output resolution	224×224	Saturation factor	$[0.5, 1.5]$
Rotation (front / wrist)	$[-5°, 5°]$ / $0°$	Hue shift	$[-0.05, 0.05]$
JPEG quality	95	Color jitter order	Random

end-effector position, 3D axis-angle orientation, and two gripper joint positions) are normalized using min–max scaling. Front and wrist views are resized independently to 224×224, and demonstration windows are sampled with stride one. A linear projection maps the current state to a 4096-dimensional token appended to the language context. Action and world flow times are sampled independently using the same time-shift factor $\rho = 5$ and temporal loss weighting. The same time-shift factor is used during inference.

In the main experiments, the RGB and geometry experts inherit Stage 1 weights, and the action expert is initialized from adapted Wan2.2 weights. All three experts are jointly optimized using the losses in Table A1. Current RGB z_0, state, and instruction remain clean; future RGB, all geometry slots, and the action chunk are denoising targets. Action queries read current RGB and actions without directly attending to geometry, enabling action-only inference without evaluating the geometry teacher, compressor, or geometry expert online.

C Simulation Experiments

C.1 LIBERO and LIBERO-Plus

C.1.1 Evaluation protocol

Policies are trained on LIBERO demonstrations and evaluated zero-shot on LIBERO-Plus (Fei et al., 2026). The evaluation includes 2,402 Spatial, 2,518 Object, 2,591 Goal, and 2,519 Long episodes, totaling 10,030. Each suite covers seven perturbation categories: camera viewpoints, robot initial states, language instructions, lighting, backgrounds, sensor noise, and object layouts. Episode success is determined by the benchmark's native success criterion. Policies use matched initial states and environment seeds.

Inputs are current front/wrist RGB, proprioception, and the task instruction. The action-only inference path is defined in Section B.5. Each query uses 10 flow steps to generate 32 actions and executes 8 actions before replanning, with inference batch size 1. Each episode is initialized with 30 no-op actions, followed by a maximum policy execution horizon of 400 steps.

Metrics. Success rate is the percentage of successful evaluation episodes. Mean $\pm$ SD summarizes perturbation-category success rates weighted by their episode counts; SD describes variation across categories, in percentage points.

Main results. GeoAct trained with image augmentation achieves a mean success rate of 84.6% on LIBERO-Plus, compared with 84.4% for $\pi_{0.5}$ (Black et al., 2025a). Its success rates under robot-initial-state and background perturbations are 70.4% and 75.2%, respectively.

C.1.2 RGB and depth forecasts

Figure A5 shows joint RGB and geometry forecasts from the 10-epoch policy on three LIBERO tasks. Each prediction is paired with the reference at the same future time step and camera view.

C.2 RoboTwin 2.0

RoboTwin 2.0 (Chen et al., 2026b) contains 50 bimanual manipulation tasks. We use the Full multi-task data protocol of Fast-WAM (Yuan et al., 2026), combining 2,500 Clean and 25,000 Randomized demonstrations. We evaluate policies separately under Clean and Randomized conditions, with environmental variation in the latter.

GeoAct is trained for 5 epochs with an effective global batch size of 1024. Optimization uses AdamW with weight decay 10^{-2}, a cosine learning-rate schedule, BF16 precision, and global

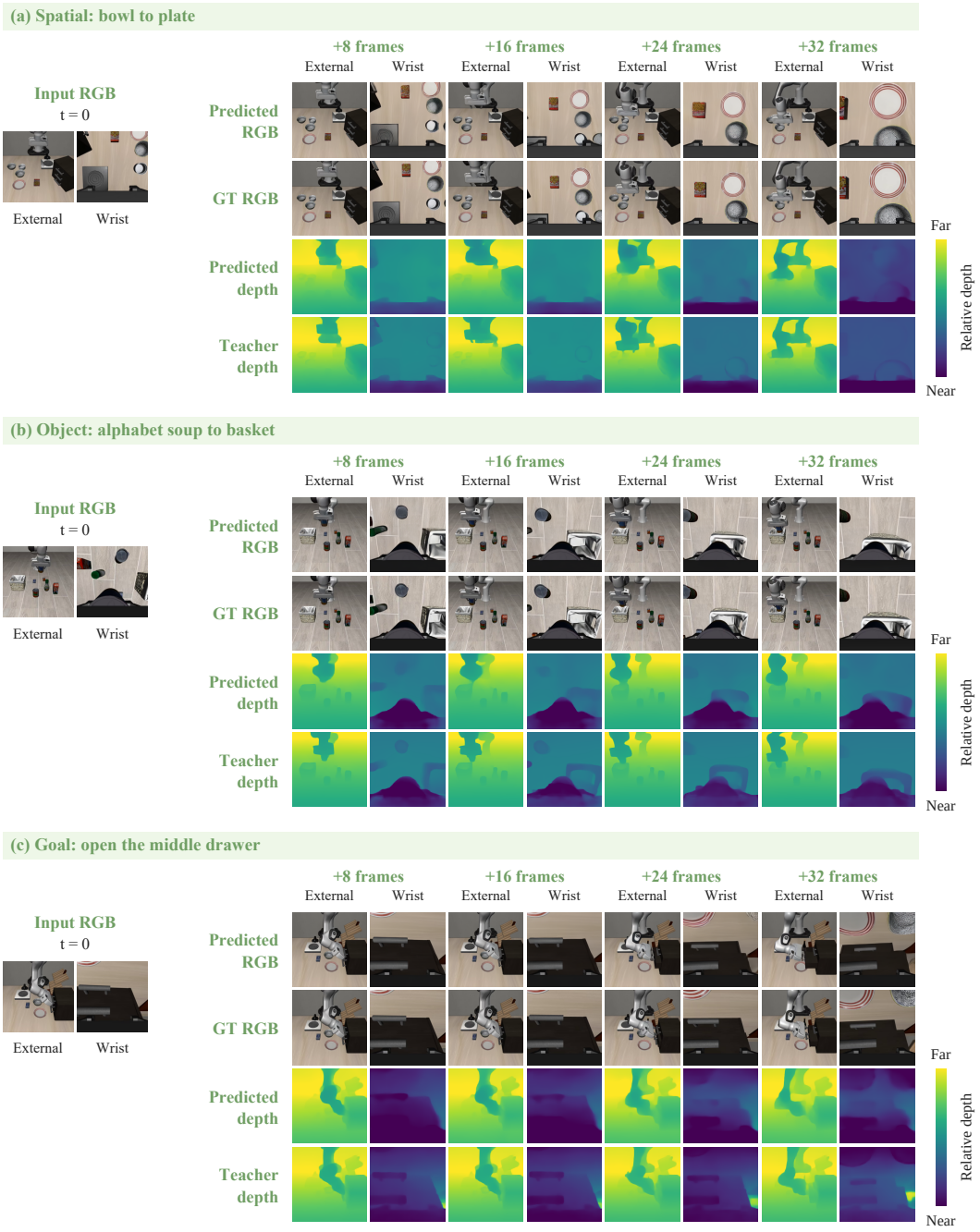

Figure A5: **Future RGB and depth predictions.** Conditioned on current external and wrist RGB, GeoAct predicts future RGB and geometric states for three LIBERO training clips. Predicted geometry is decoded through GeoAE and the frozen VGGT-Omega depth head; recorded RGB and teacher depth provide the references.

gradient-norm clipping at 1.0. Both world and action branches use the time-shift factor $\rho = 5$ during training and inference.

The policy receives external and two wrist RGB views, proprioception, and the instruction. Each view is resized independently to 224×224. A 32-step action chunk is paired with 9 video frames. We report unweighted task-averaged success rates across the 50 tasks within each condition, and average the Clean and Randomized scores to obtain the reported mean.

Table A5: **Per-task success rates on RoboTwin 2.0 (%, ↑).** Each entry gives Clean / Randomized; the final row averages all 50 tasks within each setting. Bold marks the best result for each task and setting; green highlights GeoAct.

#	Task	Motus	LingBot-VA	X-WAM	Fast-WAM	GeoAct
1	Adjust Bottle	89 / 93	90 / 94	**100** / 99	**100 / 100**	**100 / 100**
2	Beat Block Hammer	95 / 88	96 / **98**	98 / 96	**99** / 97	94 / 96
3	Blocks Ranking RGB	99 / 97	99 / 98	99 / 95	**100 / 100**	95 / 95
4	Blocks Ranking Size	75 / 63	94 / 96	76 / 82	94 / **98**	**98** / 97
5	Click Alarmclock	**100 / 100**	99 / **100**	98 / 99	**100 / 100**	**100 / 100**
6	Click Bell	**100 / 100**	**100 / 100**	**100 / 100**	**100 / 100**	**100 / 100**
7	Dump Bin Bigbin	95 / 91	89 / **96**	90 / **96**	**97 / 96**	93 / 95
8	Grab Roller	**100 / 100**	**100 / 100**	**100 / 100**	**100 / 100**	**100** / 99
9	Handover Block	86 / 73	**99** / 78	88 / 79	95 / 81	97 / **88**
10	Handover Mic	78 / 63	94 / 96	88 / 89	**99 / 100**	95 / 93
11	Hanging Mug	38 / 38	40 / 28	46 / 55	58 / **62**	**66 / 62**
12	Lift Pot	96 / 99	**100** / 99	**100** / 99	**100 / 100**	98 / **100**
13	Move Can Pot	34 / 74	**94 / 97**	80 / 84	90 / 88	93 / 96
14	Move Pillbottle Pad	93 / 96	99 / **99**	96 / 98	**100 / 99**	**100** / 98
15	Move Playingcard Away	**100** / 96	**100** / 99	**100** / 98	**100 / 100**	96 / 97
16	Move Stapler Pad	83 / **85**	**91** / 79	67 / 70	77 / 64	87 / 82
17	Open Laptop	95 / 91	92 / 94	96 / 97	**98 / 100**	93 / 83
18	Open Microwave	**95** / 91	82 / 86	89 / **92**	62 / 45	75 / 68
19	Pick Diverse Bottles	90 / 91	89 / 82	91 / **92**	80 / 85	**94** / 90
20	Pick Dual Bottles	96 / 90	**100** / 99	99 / **100**	**100** / 96	99 / 97
21	Place A2B Left	88 / 79	**97** / 93	90 / 87	95 / 93	96 / **98**
22	Place A2B Right	91 / 87	**97** / 95	92 / 89	93 / **99**	94 / 91
23	Place Bread Basket	91 / 94	**97 / 95**	90 / 91	91 / 93	91 / 94
24	Place Bread Skillet	86 / 83	**95** / 90	90 / **96**	90 / 93	91 / 88
25	Place Burger Fries	**98** / 98	97 / 95	97 / **99**	96 / **99**	94 / 90
26	Place Can Basket	81 / 76	81 / 84	84 / 82	71 / 69	**89 / 86**
27	Place Cans Plasticbox	98 / 94	**100 / 99**	99 / 98	99 / 96	99 / 97
28	Place Container Plate	98 / 99	**99** / 97	98 / **100**	96 / **100**	97 / 99
29	Place Dual Shoes	93 / 87	**94** / 89	83 / 81	**94** / 88	92 / **96**
30	Place Empty Cup	99 / 98	**100 / 100**	98 / 99	**100 / 100**	**100** / 99
31	Place Fan	91 / 87	**99** / 93	84 / 92	96 / **96**	96 / 91
32	Place Mouse Pad	66 / 68	**93 / 96**	84 / 86	83 / 89	92 / 89
33	Place Object Basket	81 / 87	**91** / 88	85 / 87	89 / 88	90 / **91**
34	Place Object Scale	88 / 85	96 / 95	93 / 89	90 / **97**	**97** / 94
35	Place Object Stand	98 / 97	**99** / 96	97 / 96	90 / 94	98 / **100**
36	Place Phone Stand	87 / 86	**97** / 97	75 / 80	**97 / 99**	95 / 97
37	Place Shoe	**99** / 97	98 / 98	97 / **99**	96 / **99**	98 / **99**
38	Press Stapler	93 / **98**	85 / 82	**94** / 90	90 / 97	89 / 89
39	Put Bottles Dustbin	81 / 79	87 / 91	85 / **95**	**95** / 90	90 / 85
40	Put Object Cabinet	88 / 71	85 / 87	66 / 76	**94 / 89**	90 / 82
41	Rotate QRcode	89 / 73	**96 / 91**	84 / 83	93 / 89	91 / 85
42	Scan Object	67 / 66	**96** / 91	86 / 79	89 / **92**	90 / 83
43	Shake Bottle	**100** / 97	**100** / 97	99 / 99	**100 / 100**	98 / 99
44	Shake Bottle Horizontally	**100** / 98	**100** / 99	**100** / 99	**100 / 100**	**100** / 99
45	Stack Blocks Three	91 / 95	**99 / 98**	97 / 95	95 / 97	97 / 93
46	Stack Blocks Two	**100** / 98	**100** / 98	**100 / 100**	**100 / 100**	**100 / 100**
47	Stack Bowls Three	79 / **87**	86 / 83	88 / 82	80 / 81	**89** / 86
48	Stack Bowls Two	**98 / 98**	94 / **98**	**98 / 98**	92 / **98**	96 / **98**
49	Stamp Seal	93 / 92	**96 / 97**	93 / 95	90 / 94	93 / 94
50	Turn Switch	**84 / 78**	44 / 45	61 / 72	61 / 59	68 / 63
50-task mean		88.66/87.02	92.90/91.50	89.76/90.68	91.88/91.78	**93.46/91.82**

Table A5 reports per-task results for Motus (Bi et al., 2026), LingBot-VA (Li et al., 2026b), X-WAM, Fast-WAM, and GeoAct under both Clean and Randomized conditions.

D Real-Robot Experiments

We evaluate three real-robot tasks: bimanual tray placement, test-tube insertion, and block handover and stacking. These tasks assess coordinated transport, insertion with bimanual support, and inter-arm object transfer, respectively.

D.1 Robot Platform and Tasks

The platform consists of two Piper arms and three D435 cameras, with one external view and one view on each wrist. Policies receive synchronized RGB, proprioception, and a task instruction. Each arm contributes six joint positions and one gripper opening, yielding 14-dimensional states and actions. Actions specify target joint positions and gripper openings. Each RGB image is resized independently to 224×224.

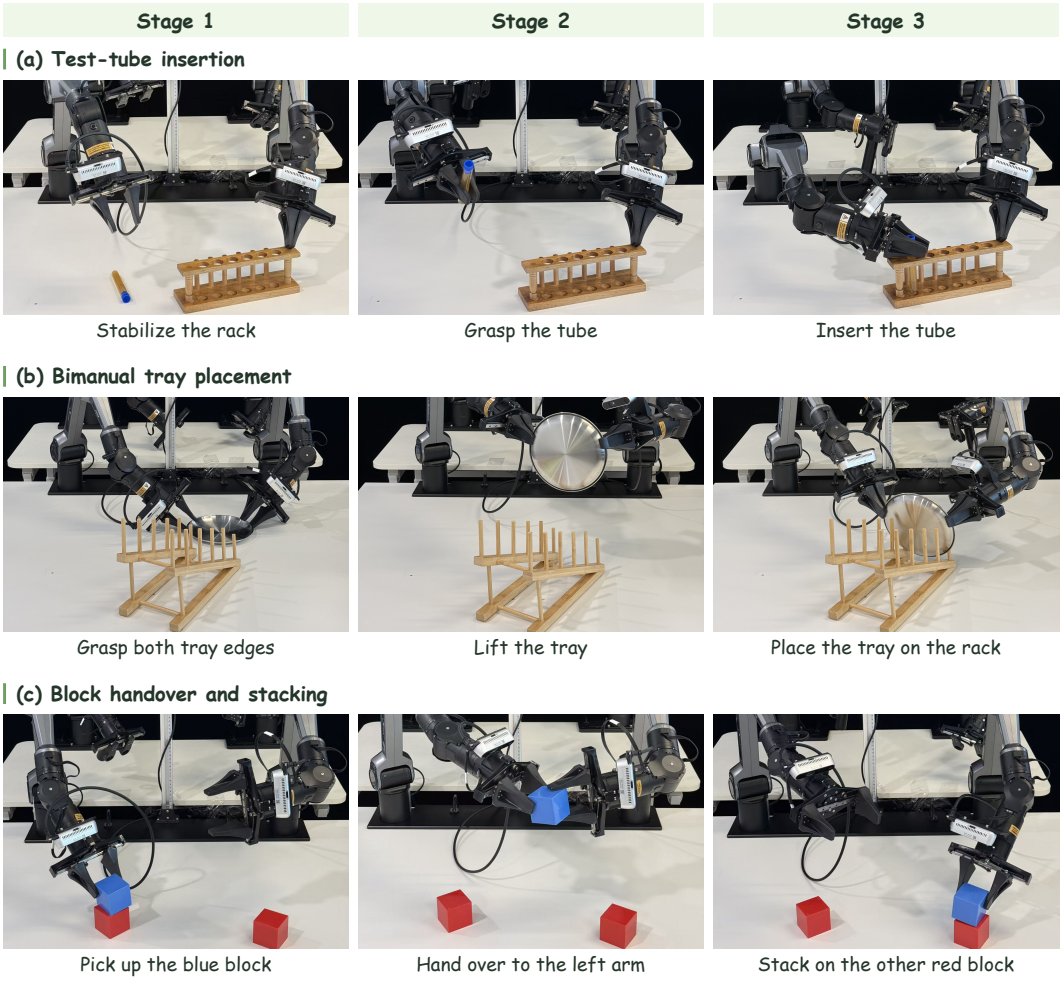

Figure A6: **Teleoperated demonstrations of the three real-robot tasks.** Each row shows three stages, ordered from left to right: (a) test-tube insertion, (b) bimanual tray placement, and (c) block handover and stacking.

Figure A6 illustrates the task procedures. **Test-tube insertion:** the right arm first stabilizes the rack; the left arm then grasps the tube and inserts it into a rack opening. **Bimanual tray placement:** both arms grasp opposite edges of the metal tray, lift it together, and place it on the rack. **Block handover and stacking:** the right arm picks up the blue block from its initial red support and transfers it to the left arm, which places it on the other red block. Table A6 lists the English training instruction for each task.

Table A6: **Real-world task instructions and success criteria.**

Task	Language instruction	Success condition
Test-tube insertion	Insert the test tube into the rack.	The released tube remains seated in a rack opening.
Bimanual tray placement	Place the metal tray on the rack.	Both grippers release the tray, which remains supported by the rack.
Block handover and stacking	Stack the blue block on the other red block.	After handover and release, the blue block remains stably stacked on the target red block.

D.2 Data, Training, and Deployment

We collect 100 teleoperated demonstrations per task, using 90 for training and 10 for offline evaluation.

GeoAct, Fast-WAM (Yuan et al., 2026), and $\pi_{0.5}$ are fine-tuned for 10 epochs on demonstrations from the three tasks. Training samples tasks uniformly and uses normalization statistics from the training split. GeoAct inherits its RGB and geometry experts from Stage 1 and initializes the action expert from adapted Wan2.2 weights. The RGB encoder and GeoAE are frozen while the three experts are optimized. Fast-WAM fine-tunes its pretrained video and action experts.

The $\pi_{0.5}$ baseline is initialized from its base checkpoint and uses an exponential moving average decay of 0.99.

Table A7 summarizes the optimization and action settings. GeoAct retains three separate views; Fast-WAM concatenates them horizontally to 224×672. The two world-action models use 9 supervised video frames spaced by 4 control steps. GeoAct uses RGB/geometry/action loss weights of $1.0/0.02/1.0$; Fast-WAM uses RGB/action weights of $1.0/1.0$. Actions execute at 20 Hz. Each policy replans after 8 actions, and each trial has a 60-second time limit.

Table A7: **Real-world adaptation and deployment configurations.**

Configuration	GeoAct	Fast-WAM	$\pi_{0.5}$
Trainable parameters	RGB, geometry, and action experts	Video and action experts	All model parameters
Training duration	10 epochs	10 epochs	10 epochs
Global batch size	32	32	64
Optimizer	AdamW	AdamW	AdamW
(β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.95)$	$(0.9, 0.95)$
Peak learning rate	10^{-5}	10^{-5}	2.5×10^{-5}
Cosine endpoint LR	10^{-7}	10^{-7}	2.5×10^{-6}
Cosine schedule length	10 epochs	10 epochs	30,000 updates
Weight decay	10^{-2}	10^{-2}	10^{-10}
Warmup / schedule	5% / cosine	5% / cosine	1,000 updates / cosine
Gradient clip / precision	1.0 / BF16	1.0 / BF16	1.0 / BF16
Action horizon	32	32	50
Executed actions per query	8	8	8
Flow steps	10	10	10

Table A8: **Real-world bimanual task success rates (%, ↑).** Entries show successful trials / 20 (success rate). Bold denotes the best Mean; green highlights GeoAct.

Method	Bimanual tray placement	Test-tube insertion	Block handover and stacking	Mean
$\pi_{0.5}$	14/20 (70%)	6/20 (30%)	11/20 (55%)	51.7
Fast-WAM	13/20 (65%)	5/20 (25%)	10/20 (50%)	46.7
GeoAct	15/20 (75%)	8/20 (40%)	11/20 (55%)	**56.7**

D.3 Success Criteria and Results

We compare GeoAct, $\pi_{0.5}$ (Black et al., 2025a), and Fast-WAM (Yuan et al., 2026), evaluating each policy in 20 independent robot trials per task from matched initial scenes. Success requires comple-

tion of the task-specific condition in Table A6, with the final state maintained for 2 seconds. Drops, timeouts, and human intervention count as failures. Table A8 reports the fraction of successful trials for each task and the equally weighted mean across the three tasks.

E Ablation Studies

We examine how geometric supervision supports action learning through complementary RGB and geometry prediction, the form of geometric targets, information retained under compression, and the deployment cost of different inference configurations.

E.1 Complementary RGB and Geometry Prediction

We examine whether RGB and geometry prediction provide complementary supervision for action learning. Table A9 compares RGB-only, geometry-only, and joint prediction across seven perturbations in each of the four LIBERO-Plus suites. RGB-only and GeoAct share the multiview input representation and positional encoding; Section B.2 describes their implementation differences from Fast-WAM. Geometry-only omits the RGB world stream and conditions actions on current geometry encoded online by frozen VGGT-Omega and GeoAE; GeoAct conditions on current RGB.

Table A9: **Combinations of world prediction targets.** LIBERO-Plus success rates (%, ↑). All variants predict actions; GeoAct jointly predicts RGB and geometry. Bold denotes the highest mean success rate.

(a) Spatial $N = 2{,}402$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
RGB only	56.01	66.09	50.77	88.46	80.93	88.46	84.62	72.58 ± 14.99
Geometry only	59.04	54.00	30.26	96.23	93.41	95.44	87.79	71.77 ± 24.44
GeoAct	77.13	70.57	86.15	98.29	87.98	98.01	91.17	**86.68** ± 9.52

(b) Object $N = 2{,}518$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
RGB only	66.82	62.86	77.80	80.91	83.10	89.36	70.35	75.35 ± 9.13
Geometry only	63.13	55.03	65.82	97.98	97.58	99.53	70.47	77.01 ± 17.60
GeoAct	81.82	56.03	93.79	99.66	77.42	97.63	80.15	**83.48** ± 14.29

(c) Goal $N = 2{,}591$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
RGB only	51.18	68.44	40.83	86.45	77.19	87.39	60.56	65.72 ± 16.31
Geometry only	55.88	52.08	25.12	93.55	90.75	95.78	62.59	65.19 ± 24.29
GeoAct	73.28	71.64	77.80	96.77	76.16	98.94	67.29	**79.35** ± 11.37

(d) Long (LIBERO-10) $N = 2{,}519$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
RGB only	40.60	54.27	33.60	61.75	40.48	70.16	68.08	52.63 ± 13.78
Geometry only	23.15	35.88	25.59	68.25	71.28	71.49	66.03	49.86 ± 21.07
GeoAct	42.24	60.31	78.59	72.26	31.14	84.63	72.76	**63.91** ± 18.16

Joint prediction achieves the highest mean success rate in all four suites and outperforms both single-stream configurations under camera, language, and layout perturbations in every suite. These changes affect scene projection, instruction wording, and relative object positions, requiring policies to combine target identification with spatial localization. RGB prediction preserves object appearance and motion cues, while geometry prediction constrains spatial relationships between the robot and objects. The two objectives jointly supervise the current visual representation used by actions, providing complementary information for instruction-conditioned manipulation.

Background perturbations show a different pattern: geometry-only prediction outperforms joint prediction in all four suites, reaching 71.28% versus 31.14% in Long. RGB reconstruction must preserve background textures, whose task-irrelevant variation may make the joint representation more sensitive to appearance changes. The complementarity therefore centers on target identification and spatial relationships, while background variation remains a weakness of joint prediction.

To examine this complementarity during closed-loop manipulation, Figure A7 shows paired executions under camera, robot-initial-state, and object-layout perturbations. In the selected episodes, GeoAct succeeds and both single-stream policies fail; matched initial states and action steps allow their executions to be compared directly.

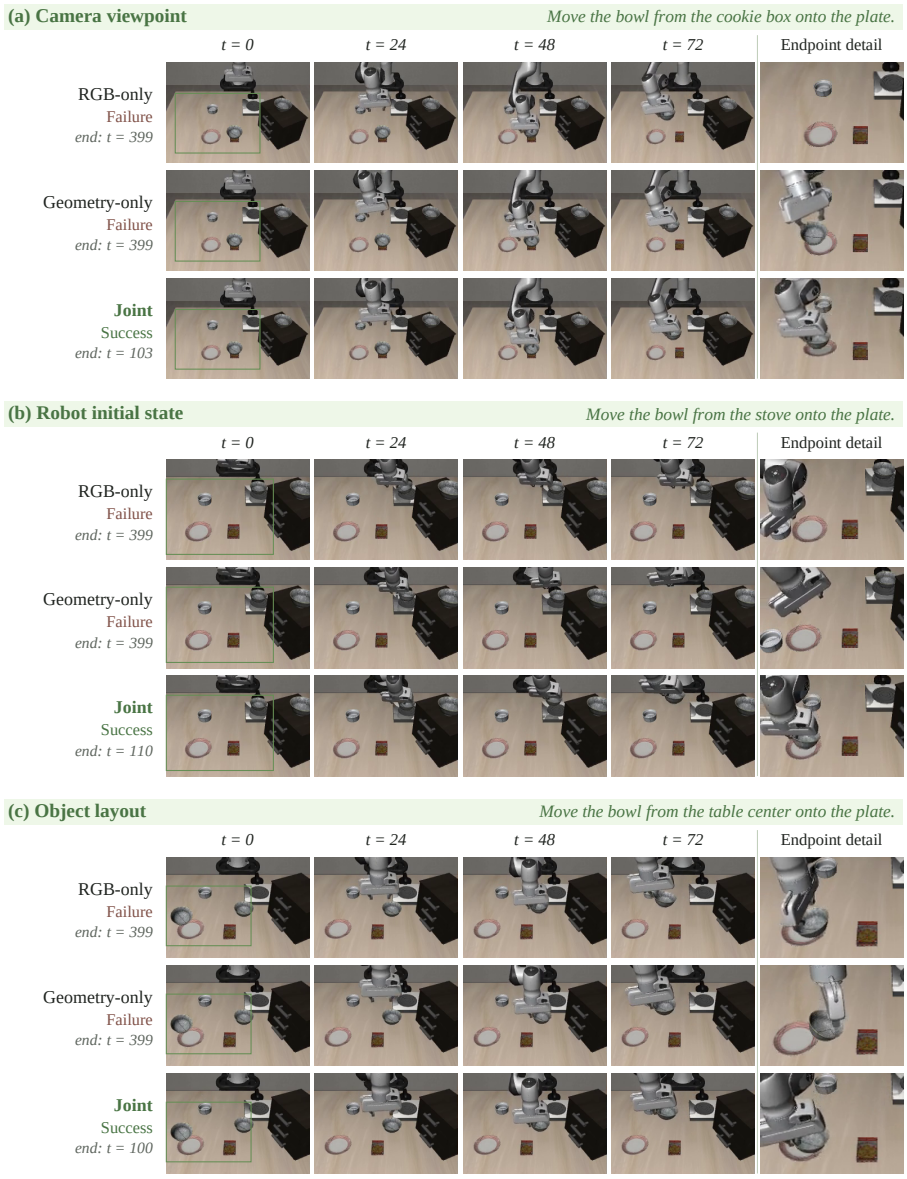

Figure A7: **Policy executions under spatial perturbations.** GeoAct (Joint) succeeds in the selected episodes where RGB-only and geometry-only policies fail. Policies share the initial state; columns show matched action steps and the final state. (a) Camera viewpoint. (b) Robot initial state. (c) Object layout.

E.2 Geometry-Aligned RGB Prediction

Geometry prediction supervises the RGB representations used for action generation during training. We therefore compare GeoAct geometry-sequence prediction with an RGB-only baseline that receives geometric supervision through feature alignment within MoT.

Alignment. The baseline aligns RGB expert hidden features with frozen VGGT-Omega features from the same observations (Figure A8), alongside RGB and action flow supervision. Table A10 specifies the alignment implementation. The teacher and projection head are used only during training.

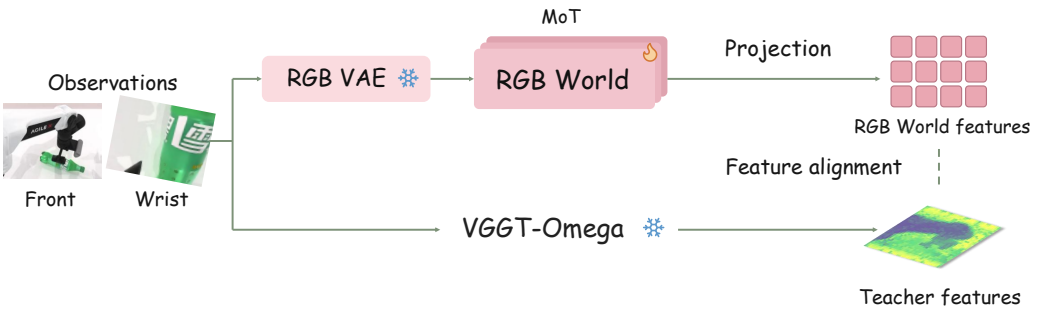

Figure A8: **RGB expert feature alignment in MoT.** Projected RGB expert features are trained to align with frozen VGGT-Omega targets. Feature imagery is schematic.

Table A10: **RGB expert alignment configurations.**

Configuration	RGB-only + feature alignment
Alignment target	VGGT-Omega features of the current observation
Student features	layer 24 of the 30-layer RGB expert
Projection head	LN(3072) →Linear(2048) →GELU→Linear(2048)
Patch-grid matching	Bilinear resize, $7 \times 7 \rightarrow 14 \times 14$
Alignment objective	Patchwise cosine distance
RGB / action / alignment	1.0 / 1.0 / 0.5
Training duration	20,000 iterations
Global batch size	128

LIBERO-Plus comparison. Both policies use the same 10,030 episodes, initial states, and environment seeds, with 10 flow steps, a 32-action horizon, and replanning every eight actions. Table A11 reports episode-weighted success by suite and overall.

Table A11: **RGB expert alignment versus geometry-sequence prediction.** LIBERO-Plus success (%, ↑); Overall pools 10,030 episodes.

Policy	Spatial	Object	Goal	Long	Overall
RGB-only + alignment	79.98	**84.03**	71.67	59.27	73.65
GeoAct (geometry prediction)	**86.68**	83.48	**79.35**	**63.91**	**78.27**

Analysis. GeoAct outperforms explicit feature alignment by 4.62 percentage points overall, with gains of 6.70, 7.68, and 4.64 percentage points on Spatial, Goal, and Long, respectively. While observation-level feature alignment (analogous to Spatial Forcing) introduces spatial priors, it aligns current policy features with geometric targets without directly supervising future geometric states. In contrast, GeoAct explicitly couples policy representations with temporal-geometric transitions, improving performance on the Spatial, Goal, and Long suites.

E.3 Comparison of Geometric Prediction Targets

We examine how the representation of a geometric prediction target affects policy learning. Table A12 compares depth, DINO features (ST-WAM (Wang et al., 2026b)), raw VGGT-Omega features, and GeoAE latents as auxiliary targets. Every model predicts RGB and actions; GeoAct uses GeoAE latents.

GeoAE latents outperform direct depth and raw VGGT feature prediction under camera and object-layout perturbations. Direct depth reduces geometric representations to visible-surface distances, while the world model must predict depth changes at high spatial resolution. Direct VGGT prediction retains multiple feature layers but requires the model to predict high-dimensional features that lie on the latent manifold expected by the frozen geometric decoder. Through joint feature and geometry reconstruction, GeoAE organizes these layers into compact spatiotemporal states. Its pretrained decoder handles feature recovery, allowing the world model to learn geometric evolution with fewer prediction variables.

Table A12: **Choice of world prediction targets.** LIBERO-Plus success rates (%, ↑). "+" adds the indicated target to joint RGB–action prediction. Bold denotes the highest mean success rate.

Prediction target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean
RGB only	53.41	62.88	50.07	79.60	69.59	83.31	70.76	66.49
+ DINO (ST-WAM)	65.46	70.25	71.33	83.77	67.42	83.82	72.47	73.40
+ Depth	59.16	61.61	60.05	89.32	62.64	90.44	75.28	70.93
+ VGGT features	52.47	69.35	65.06	91.86	63.75	95.75	75.74	73.15
+ GeoAE latent	68.17	64.52	83.80	92.03	67.19	94.38	77.84	**78.27**

Table A13 further shows that GeoAE achieves the highest mean success rate among the three geometric targets in all four suites. Raw VGGT features retain an advantage in some settings, including robot-initial-state perturbations in Object.

Table A13: **Geometric prediction targets across task suites.** LIBERO-Plus success rates (%, ↑). Each target is added to RGB–action prediction. Bold denotes the highest mean success rate.

(a) Spatial $N = 2{,}402$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
+ Depth	67.29	63.43	65.38	97.60	82.17	95.44	90.13	79.48 ± 13.75
+ VGGT features	71.28	68.86	64.62	97.26	82.56	95.44	91.43	80.97 ± 12.66
+ GeoAE latent	77.13	70.57	86.15	98.29	87.98	98.01	91.17	**86.68** ± 9.52

(b) Object $N = 2{,}518$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
+ Depth	76.26	62.81	88.14	95.62	74.19	91.94	76.92	80.62 ± 10.74
+ VGGT features	61.62	74.12	94.92	94.28	74.19	98.58	75.19	81.73 ± 13.26
+ GeoAE latent	81.82	56.03	93.79	99.66	77.42	97.63	80.15	**83.48** ± 14.29

(c) Goal $N = 2{,}591$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
+ Depth	55.64	67.97	44.88	96.77	65.48	97.36	65.41	69.09 ± 18.01
+ VGGT features	50.00	73.11	50.98	98.57	72.24	98.42	64.94	70.98 ± 18.21
+ GeoAE latent	73.28	71.64	77.80	96.77	76.16	98.94	67.29	**79.35** ± 11.37

(d) Long (LIBERO-10) $N = 2{,}519$

World target	Camera	Robot	Lang.	Light	Bkgd.	Noise	Layout	Mean ± SD
+ Depth	39.14	52.16	44.91	66.06	32.53	79.29	68.27	54.98 ± 16.02
+ VGGT features	29.36	61.07	53.00	76.64	29.76	91.09	71.79	59.35 ± 22.09
+ GeoAE latent	42.24	60.31	78.59	72.26	31.14	84.63	72.76	**63.91** ± 18.16

The main-paper depth-prediction figure compares the three targets through recovered scene structure. Direct depth is restored according to its training target definition; predicted VGGT features pass through the corresponding frozen depth head; GeoAE latents first recover features through the decoder and then use the frozen head to read out depth. All methods are shown at corresponding times within the same LIBERO training window, using the simulator-aligned visualization defined in Section B.3.2 to compare retained object contours and relative depth.

E.4 Comparison of Geometric Compressors

We examine whether feature reconstruction accuracy sufficiently reflects the geometric quality of a compressed representation, and compare compressors as world prediction targets. Table 5 of the main paper reports feature preservation and depth readouts from a frozen geometric head; Table A14 gives suite-level policy results for PCA-T3, Feature-only AE, and GeoAE.

GeoAE and Feature-only AE achieve comparable feature reconstruction accuracy, yet GeoAE recovers more accurate depth through the same frozen head (Figure A9). A geometric head responds differently to different feature changes, so a lower average feature error need not preserve the information that determines geometric readouts. GeoAE backpropagates depth and camera recon-

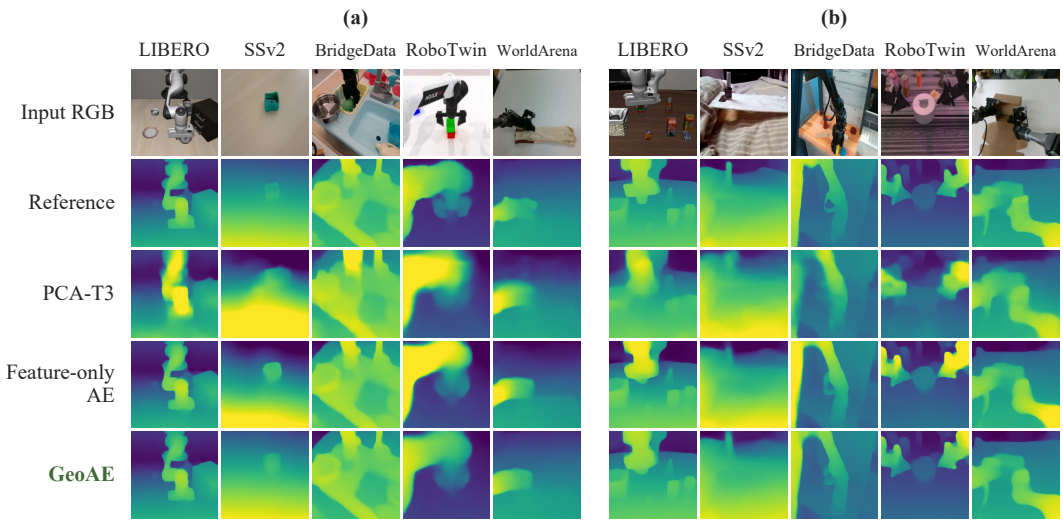

Figure A9: **Depth reconstruction after geometric compression.** We compare PCA-T3, Feature-only AE, and GeoAE on two scenes from each of five data domains. Reconstructed features are decoded by the same frozen depth head; Reference uses uncompressed VGGT-Omega features.

Table A14: **Comparison of geometric compression methods.** LIBERO-Plus success rates (%, ↑). Bold denotes the highest mean success rate.

Compressor	Spatial	Object	Goal	Long	Mean
PCA-T3	80.60	85.31	72.29	57.21	73.76
Feature-only AE	85.80	79.79	74.64	60.50	75.05
GeoAE	86.51	83.20	79.97	64.27	**78.40**

struction errors through frozen geometric heads to the encoder and decoder, directly constraining compression by the error in recovered scene structure.
GeoAE exceeds Feature-only AE in all four suites and achieves the highest mean success rate, while PCA-T3 retains an advantage on Object.

E.5 Inference Configurations and Efficiency

Table A15 reports LIBERO-Plus success rates and action-chunk inference latency for the four configurations. Figure A10 shows their training and inference attention patterns.

DINO features are extracted independently for each frame, whereas VGGT-Omega aggregates information across the input images. Consequently, the representation of a given frame can differ depending on whether it is encoded with only the current multiview observations or with the complete nine-frame clip. To avoid a training–inference mismatch in geometric conditioning, the s_0-conditioned variant uses current observations alone to construct its condition during both policy training and deployment.
During training, we separately encode the current multiview observations through VGGT-Omega and GeoAE to obtain s_0^{current}, and encode the complete nine-frame clip to obtain $[s_0^{\text{clip}}, s_1^{\text{clip}}, s_2^{\text{clip}}]$. We construct the hybrid condition by substituting the clip-level initial token with the single-frame latent:

$$S_{\text{cond}} = [s_0^{\text{current}},\ s_1^{\text{clip}},\ s_2^{\text{clip}}]. \tag{A7}$$

The initial latent remains clean, while the future latents are denoising targets. The condition and targets thus use different temporal contexts, requiring the model to account for context-dependent feature variation in addition to physical scene evolution. This additional variation may limit the benefit of geometric conditioning. GeoAct retains all three latents from the same clip encoding and jointly denoises the complete sequence.

Inference implementation. All configurations cache language features for the fixed task instruction and use CUDA Graph replay to reduce launch overhead for fixed-shape computations. Fast-

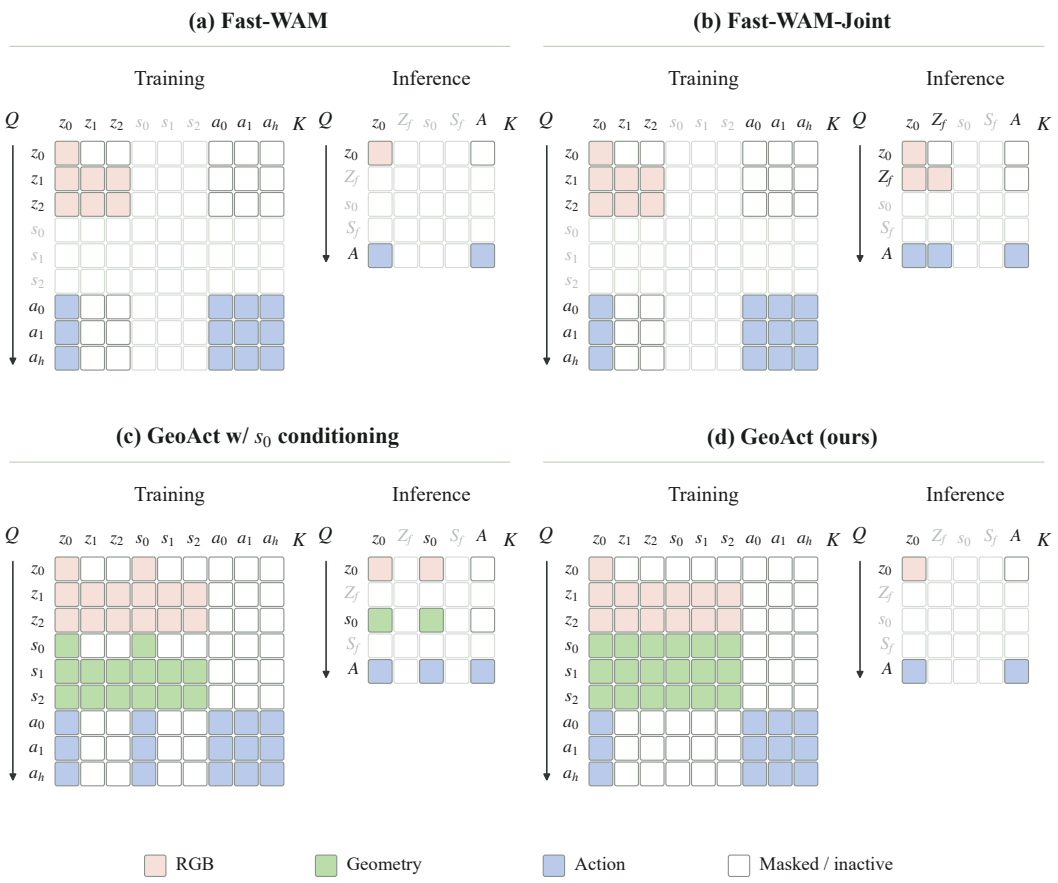

Figure A10: **Attention under four inference configurations.** Fast-WAM-Joint generates future RGB with actions; GeoAct w/ s_0 conditioning encodes current geometry online. GeoAct predicts the complete geometry sequence during training and generates only actions at inference. Rows and columns denote queries and keys.

Table A15: **Success rates and inference latency.** LIBERO-Plus success rates (%, ↑) and latency over 20 calls on one A800, using 10 integration steps per 32-step action chunk. SD denotes sample standard deviation. Bold denotes the best mean success rate.

	Latency (ms) ↓		LIBERO-Plus
Configuration	**Mean ± SD**	**Median**	**Mean (%) ↑**
Fast-WAM (Yuan et al., 2026)	165.6 ± 37.9	156.2	51.50
Fast-WAM-Joint	510.3 ± 104.3	463.5	59.00
GeoAct w/ s_0 conditioning	281.8 ± 58.6	263.8	75.16
GeoAct	177.7 ± 54.1	160.5	**78.27**

WAM, GeoAct, and the s_0-conditioned variant reuse observation-prefix keys and values because these conditions remain fixed across denoising steps. Fast-WAM-Joint recomputes its changing video states during joint RGB–action denoising.

Timing protocol. All configurations run on one NVIDIA A800-SXM4-80GB with BF16, batch size 1, and 10 flow steps to produce 32 actions. Inputs are two 224×224 RGB views; Fast-WAM uses its native 224×448 concatenated layout. Latency includes RGB encoding, conditioning, denoising of all active streams, and returning the predicted action chunk; the s_0-conditioned variant includes online geometry encoding. After five warm-up runs, we report the mean, sample SD, and median over 20 timed inference runs. Timing excludes input preprocessing, language encoding, and initial graph capture.