WAM4D: FAST 4D WORLD ACTION MODEL VIA SPATIAL REGISTER TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

World-action models couple future visual prediction with robot action generation, yet visually plausible futures do not necessarily capture the spatial changes required for reliable manipulation. We introduce WAM4D, a geometry-grounded world-action model that connects future geometric prediction to robot control while retaining RGB-only sensing. Guided by a pretrained geometric decoder and future-depth supervision, spatial register tokens learn temporally structured features from RGB observation history within a shared video-action backbone. Asymmetric causal attention makes these geometry-supervised features directly available to action queries alongside RGB features, integrating geometric prediction and action generation into single-stage training without future-target leakage. At inference, history and register features are computed once per decision step and cached for flow-based action denoising, requiring neither future-video generation nor dense depth decoding. In controlled experiments on four RoboTwin 2.0 tasks, WAM4D achieves higher average success than an RGB-only WAM and a 4D-WAM reimplementation across their common evaluation conditions, with the largest gains under texture and dynamic lighting changes. Frozen-feature probes further show that its registers retain action-relevant information and are less sensitive to appearance perturbations than RGB-only WAM features. Extensive evaluations show competitive full-task RoboTwin performance, improved geometric prediction on RT-1 and Bridge-Data, and higher average sub-action success in real-world manipulation.

1 INTRODUCTION

VLA policies have become a dominant route for robot control by directly mapping visual observations and language instructions to actions (Brohan et al., 2022; Black et al., 2024; Physical Intelligence et al., 2025; Cao et al., 2026; Intelligence et al., 2026). However, manipulation is not a simple observation to action mapping. It requires reasoning about scene geometry, contacts, object dynamics, and future scene evolution. Current VLAs learn these factors mainly from action supervision, which does not explicitly require predicting how the physical world evolves under interaction and can limit generalization (Zhang et al., 2026b). Recent spatially grounded VLAs address this issue by injecting 3D inputs or geometric foundation priors into policy learning (Li et al., 2026a; Sun et al., 2025; Li et al., 2025; Wang et al., 2026a). These methods improve the spatial representation of direct policies, but scene evolution and robot dynamics remain implicit in the action prediction objective.

Meanwhile, world models have been viewed as a foundation for embodied intelligence by predicting how the physical environment evolves under interaction (Wan et al., 2025; Yang et al., 2025; Ding et al., 2025). With the rapid progress of video generative models, this predictive capability has begun to transfer from general video modeling to embodied domains (Chi et al., 2025b; Li et al., 2026d;c; Zeng et al., 2026; Chen et al., 2026). As robotic simulators of future scene evolution, world models support data generation and policy evaluation (Fan et al., 2026; Zhou et al., 2024; Yu et al., 2025; Wu et al., 2026a). The generated videos are further converted into executable actions using inverse dynamics models. However, future prediction and action generation remain separate in this pipeline. World action models (WAMs) bridge this gap by extending world models from robotic video generation to joint prediction of future observations and executable robot actions (Li et al., 2026b; Yuan et al., 2026b; Team et al., 2026; Zhang et al., 2026c). By using the priors of video generative foundation models, this end-to-end formulation couples imagination and control in one

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.

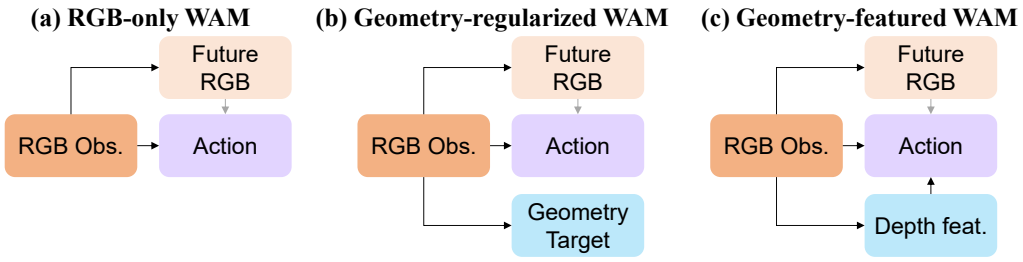


Figure 1: Information flow in world-action models. (a) RGB-only WAMs represent geometry implicitly; (b) geometry-regularized WAMs use geometric supervision without exposing geometry-specific features to action prediction; (c) geometry-featured WAMs provide such features directly to the action predictor. Gray arrows indicate optional dependence on predicted future RGB. WAM4D follows (c) using depth-supervised spatial registers derived from RGB history.

pipeline. It learns an implicit transition model that predicts how the scene and robot may evolve from the initial observation and task instruction.

Yet existing WAMs still predominantly operate in 2D RGB or latent video spaces. Visually plausible futures may conceal errors in manipulation-critical geometry, such as relative object poses, contacts, and occluded structure (Ai et al., 2025). As illustrated in Fig. 1, WAMs differ in how geometric information reaches action prediction. RGB-only WAMs encode geometry implicitly in visual features (Li et al., 2026b; Yuan et al., 2026b; Ma et al., 2026; Kim et al., 2026; Pai et al., 2025); future RGB generation need not be part of action inference (Yuan et al., 2026b). Geometry-regularized WAMs add geometric training targets while keeping geometry-specific features outside the action pathway (Guo et al., 2026; Yuan et al., 2026a; Zhang et al., 2026a; Yang et al., 2026a). Geometry-featured WAMs instead expose such features directly to the action predictor, as in SA-WAM (Lopetegui Gonzalez et al., 2026) and related multimodal models (Hung et al., 2026; Yan et al., 2026; Zhao et al., 2026).

We propose WAM4D as a geometry-featured WAM that retains RGB-only sensing. Future-depth supervision shapes spatial registers derived from RGB history through shared backbone attention. Action queries read these registers directly, while dense depth decoding is used only during training.

We summarize our main contributions as:

- We use future depth as training-time geometric supervision for world-action learning, enabling geometry-grounded action prediction without requiring depth inputs or depth prediction at inference.
- We integrate spatial registers through asymmetric causal routing in a single stage, transferring depth-supervised geometry into action prediction while preserving the RGB-action pathway.
- We show that geometry supervision improves manipulation robustness under controlled distribution shifts and yields action-relevant representations less sensitive to nuisance appearance variations.

2 Related Work

Robotic Manipulation Policies. Robot manipulation policies commonly learn direct mappings from observations and language instructions to actions. Early methods model action sequences from demonstrations (Zhao et al., 2023; Chi et al., 2025a), with related approaches also studying generative and inference-based trajectory optimization (Yin et al., 2024; 2025; Cai et al., 2025). RT-series models formulate robot control as sequence prediction over visual, language, and action tokens (Brohan et al., 2022; Zitkovich et al., 2023). Open generalist policies scale this paradigm with large robot datasets (Kim et al., 2024; Team et al., 2024; Liu et al., 2026b), while recent diffusion and flow policies improve cross-embodiment generalization through larger training mixtures and stronger action modeling (Liu et al., 2025; Black et al., 2024; Physical Intelligence et al., 2025; Intelligence et al., 2026). Spatially grounded VLAs further inject 3D representations or geometric foundation priors into action prediction (Qu et al., 2025; Li et al., 2026a; Sun et al., 2025; Li et al., 2025; Wang

et al., 2026a). These policies improve spatial grounding, but future scene evolution and interaction dynamics remain implicit in the action prediction objective, motivating predictive world models.

Embodied World Models and World-Action Models. Video generative models have made strong progress in modeling temporal visual dynamics (Ho et al., 2022; Blattmann et al., 2023; Wan et al., 2025; Yang et al., 2025; Seedance et al., 2026). This progress has motivated embodied world models that use video generation as learned simulators for interactive environments (Chi et al., 2025b; Li et al., 2026d;c; Zeng et al., 2026; Chen et al., 2026). Several works learn interactive world models from mixed embodied data and Internet videos (Yang et al., 2023; Yin et al., 2026). Other works predict robotic videos and convert them into actions with inverse dynamics models (Du et al., 2023; Zhou et al., 2024; Wang et al., 2025a). These works show that video priors can model future scene evolution, but future prediction and action generation are separated. Recent world action models (WAMs) jointly model future observations and executable robot actions (Wang et al., 2026d; Bi et al., 2025; Team et al., 2026; Li et al., 2026b). Other video-action models couple future prediction and control through joint generation or intermediate video features (Ye et al., 2026b; Kim et al., 2026; Ma et al., 2026; Pai et al., 2025). Fast-WAM and GigaWorld-Policy make future-video generation optional at inference (Yuan et al., 2026b; Ye et al., 2026a), while Next Forcing adds multi-chunk future supervision to causal world modeling (Xu et al., 2026a). Predictive embedding methods instead model future representations: V-JEPA 2-AC uses action-conditioned prediction for planning (Assran et al., 2025), StageWAM adds stage-level prediction to a WAM (Liu et al., 2026a), and LeWAM predicts future embeddings for action generation (Fang et al., 2026). Despite these advances, most WAMs still represent future scene evolution primarily in RGB or latent video spaces.

Geometry-Grounded World Models and WAMs. Feed-forward geometric foundation models recover depth, point maps, camera poses, and dense correspondence from images, providing scalable geometric supervision (Wang et al., 2024; 2025c; Lin et al., 2025). Building on such priors, geometry-grounded world models extend RGB future prediction with depth, surface normals, point maps, optical flow, or explicit 4D structure (Zhen et al., 2025; Xu et al., 2026b; Wang et al., 2026c; Zhao et al., 2026). GEM-4D distills 4D correspondence into a video backbone but obtains actions through inverse dynamics (Zhou et al., 2026). Geometry-regularized WAMs add geometric training targets while keeping geometry-specific features outside the action pathway (Guo et al., 2026; Yuan et al., 2026a; Zhang et al., 2026a; Yang et al., 2026a). Geometry-featured WAMs instead expose such features to action prediction (Lopetegui Gonzalez et al., 2026; Hung et al., 2026; Yan et al., 2026). WAM4D follows the latter approach by supervising future depth through spatial registers whose cached features remain available to action queries even when dense depth decoding is skipped.

3 METHOD

3.1 WAM4D BACKBONE

We use a causal video-action backbone to jointly model future visual observations and robot actions. At decision step t, the model conditions on a language instruction l, multi-view RGB history, and a queue of historical robot states. Let O_t^{hist} denote the history RGB mosaic sequence, O_t^{fut} the future RGB targets used during training, and $s_{i:j}$ and $a_{i:j}$ the robot state and action sequences over $[i, j)$, respectively. With state history length L_s and action prediction horizon H_a, the causal context is

$$\mathcal{C}_t = \{l,\ O_t^{\text{hist}},\ s_{t-L_s:t}\}, \tag{1}$$

where future RGB frames and future actions are prediction targets rather than additional causal inputs. The noised future video and action tokens are still included in the transformer sequence only as flow-matching states, so that the model can predict their corresponding flow targets.

For the video stream, a video VAE encodes the RGB sequence and splits the resulting latents into history tokens and clean future targets:

$$[\mathbf{Z}_t^{\text{hist}},\ \mathbf{Z}_t^{\text{fut}}] = E_{\text{vae}}\left([O_t^{\text{hist}},\ O_t^{\text{fut}}]\right). \tag{2}$$

During flow-matching training, the future video tokens fed to the backbone are noised states of these clean future targets, denoted as $\tilde{\mathbf{Z}}_t^{\text{fut}}$.

The state history and future action chunk are embedded as:

$$\mathbf{S}_t^{\text{hist}} = E_a(s_{t-L_s:t}), \qquad \tilde{\mathbf{A}}_t^{\text{fut}} = E_a(\tilde{a}_{t:t+H_a}), \tag{3}$$

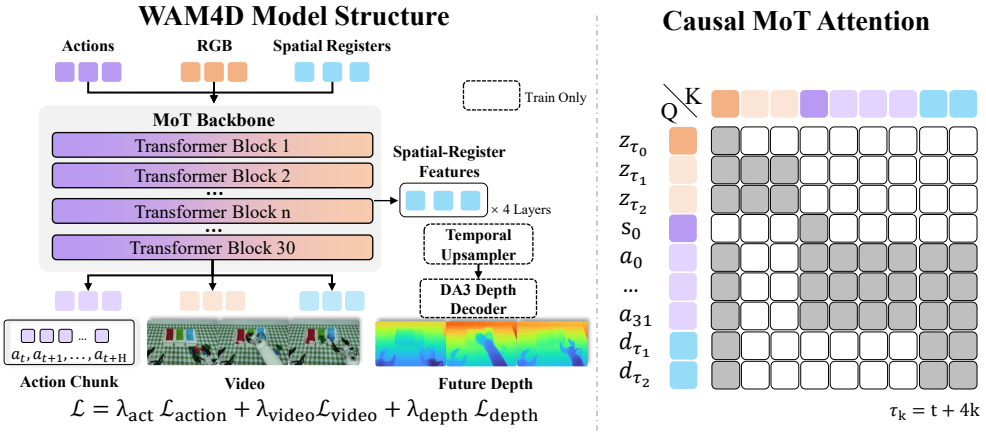


Figure 2: WAM4D architecture and causal MoT attention. Spatial registers share the backbone transformer blocks with RGB and action tokens. Intermediate spatial-register features feed the temporal upsampler and DA3 depth decoder for future-depth supervision; action queries can read register keys and values under the causal mask.

where E_a is the lightweight embedding layer shared by state and action tokens, and $\tilde{a}_{t:t+H_a}$ denotes the flow-matching state of the future action chunk.

WAM4D follows causal WAMs (Li et al., 2026b) in jointly modeling future video latents and future actions, while adding spatial registers to the shared token sequence:

$$\mathbf{X}_t^{(0)} = [\mathbf{Z}_t^{\text{hist}},\ \tilde{\mathbf{Z}}_t^{\text{fut}},\ \tilde{\mathbf{A}}_t^{\text{fut}},\ \mathbf{S}_t^{\text{hist}},\ \mathbf{R}_t^0]. \tag{4}$$

Here $\mathbf{R}_t^0$ denotes the input spatial registers, defined below. The complete sequence is processed by a video-action Mixture-of-Transformers (MoT) backbone:

$$\mathbf{X}_t^{(\ell+1)} = \text{VABlock}_\ell\left(\mathbf{X}_t^{(\ell)}; \mathcal{M}\right), \tag{5}$$

where $\mathcal{M}$ is the asymmetric causal visibility mask, specified in Sec. 3.3. The video and action heads predict flow targets for the noised future video and action tokens, respectively. The clean future video latents $\mathbf{Z}_t^{\text{fut}}$ and the action chunk $a_{t:t+H_a}$ are used to construct the corresponding training targets.

The spatial registers remain in the shared sequence throughout the backbone; their geometric supervision and causal interaction with action tokens are described below.

3.2 Depth-Supervised Spatial Registers

Spatial registers are geometry-supervised latent tokens within the same shared transformer. They traverse all backbone layers alongside video and action tokens, using the same attention projections and feed-forward networks. Their spatial-temporal initialization and causal visibility distinguish them from the other tokens. The registers are aligned with the multi-view RGB mosaic and instantiated at coarse future time slots. Since multi-camera observations are tiled into a single RGB canvas before VAE encoding, each register corresponds to a spatial region of the mosaic and a coarse future time.

Let $\mathbf{R}_\star$ denote the learnable register grid, and let $\mathcal{T}_t^{\text{coarse}}$ denote the coarse future query times. The input registers are constructed as

$$\mathbf{R}_t^0 = \text{Repeat}_{\tau \in \mathcal{T}_t^{\text{coarse}}}\left(\mathbf{R}_\star\right). \tag{6}$$

The register states are read from the complete sequence after each backbone layer:

$$\mathbf{R}_t^{\ell+1} = \text{Select}_R\left(\mathbf{X}_t^{(\ell+1)}\right). \tag{7}$$

A standard RoPE encoding uses the target timestep and mosaic coordinates of the register and video tokens. Under the causal mask, registers attend to themselves and history-video tokens.

To supervise geometry in these shared latent states, we read register features from a set of intermediate backbone layers $\mathcal{L}_r$. A lightweight shared temporal upsampler expands each readout to match the

future-depth horizon. Layer-specific learned adapters then map the readouts to the feature spaces of the pretrained DA3 depth decoder:

$$\mathbf{F}_{t,\ell}^{\mathrm{geo}} = \mathcal{P}_\ell\big(\mathcal{U}_t(\mathbf{R}_t^{\ell+1})\big), \quad \ell \in \mathcal{L}_r, \tag{8}$$

$$\hat{\mathbf{D}}_t^{\mathrm{fut}} = \mathcal{D}_\phi\Big(\{\mathbf{F}_{t,\ell}^{\mathrm{geo}}\}_{\ell\in\mathcal{L}_r}\Big). \tag{9}$$

Here $\mathcal{U}_t$ is the shared temporal upsampler, $\mathcal{P}_\ell$ is the adapter for readout layer ℓ, and $\mathcal{D}_\phi$ is the depth decoder from Depth Anything 3 (Lin et al., 2025). $\hat{\mathbf{D}}_t^{\mathrm{fut}}$ is the predicted future depth sequence.

The future-depth loss supervises these register readouts and backpropagates through the shared video-action backbone. The pretrained decoder provides a structured geometric readout that encourages the register states and shared backbone representations to encode depth-compatible information. The resulting geometry-supervised registers remain part of the shared backbone and inform action prediction through the asymmetric visibility mask $\mathcal{M}$, described next.

3.3 Asymmetric Causal Visibility

To route geometric information into action prediction while preserving causality, we apply the asymmetric visibility mask $\mathcal{M}$ illustrated in Fig. 2. To specify the permitted attention connections, let $\mathcal{V}(i)$ be the valid key/value tokens visible to query i within the same sample and define the additive attention mask as

$$\mathcal{M}_{ij} = \begin{cases} 0, & j \in \mathcal{V}(i), \\ -\infty, & \text{otherwise.} \end{cases} \tag{10}$$

To prevent future-target leakage, history RGB tokens attend only to valid history RGB, while future-video tokens attend to history RGB and noised future video and remain invisible to action queries. To encourage robust geometric representations from observed visual context, spatial registers attend only to valid history RGB and one another, excluding future video and action tokens. To make these geometry-supervised representations available for control, future action queries additionally read the registers alongside history RGB, state history, and noised future-action tokens, establishing a one-way geometry-to-action pathway. To condition predictions on language, the backbone retains its text cross-attention pathway outside $\mathcal{M}$, with register queries excluded.

To reuse geometric features efficiently at inference, we process history RGB and spatial registers in a single forward pass, cache their layer-wise keys and values, and let action denoising read this cache without running the temporal upsampler, adapters, or dense depth decoder.

3.4 Training Objective

The main video-action stream is trained with the conditional flow matching objective of the base causal WAM (Li et al., 2026b). We denote the future-video and future-action losses as $\mathcal{L}_{\mathrm{video}}$ and $\mathcal{L}_{\mathrm{action}}$, respectively. To supervise future geometry while giving each valid time step equal weight, we average the SmoothL1 depth error over valid pixels within each frame and then over valid sampled future timestamps. Let $\mathcal{T}_t$ contain the sampled future depth timestamps with at least one valid pixel, and let Ω_τ collect valid pixels across camera views at time τ. With $\hat{D}_{\tau,p}$ and $D_{\tau,p}$ denoting predicted and target depth, the per-sample loss is

$$\mathcal{L}_{\mathrm{depth}} = \frac{1}{|\mathcal{T}_t|}\sum_{\tau\in\mathcal{T}_t}\frac{1}{|\Omega_\tau|}\sum_{p\in\Omega_\tau}\mathrm{SmoothL1}_\beta\left(\hat{D}_{\tau,p}, D_{\tau,p}\right). \tag{11}$$

Here β is the SmoothL1 transition parameter; the batch loss averages over samples with valid depth supervision. The base training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{video}} + \lambda_{\mathrm{act}}\mathcal{L}_{\mathrm{action}} + \lambda_{\mathrm{depth}}\mathcal{L}_{\mathrm{depth}}. \tag{12}$$

To learn geometry-aware representations, the depth loss backpropagates through the spatial registers and shared backbone, while jointly optimizing the temporal upsampler, adapters and depth decoder.

Table 1: RoboTwin 2.0 full-task success rate and inference cost.

Method	Clean	Rand.	Avg.	Latency (ms)	VRAM (GiB)
π_0	65.9	58.4	62.2	64.16 ± 0.06	**8.45**
$\pi_{0.5}$	82.7	76.8	79.8	72.03 ± 0.06	**8.45**
Motus	88.7	87.0	87.9	1516.30 ± 10.64	11.55
LingBot-VA	92.9	91.6	**92.3**	843.57 ± 11.55	12.97
Fast-WAM	91.9	**91.8**	91.8	425.53 ± 6.01	11.55
WAM4D	**94.2**	90.1	92.2	592.34 ± 7.12	10.09

Table 2: Real-robot sub-action success over 10 rollouts per task.

Method	Plate	Bottle	Blocks			Pen		Avg.
	S1	S1	S1	S2	S3	S1	S2	
$\pi_{0.5}$	1.0	0.8	0.7	0.6	0.5	0.8	0.8	0.74
LingBot-VA	1.0	1.0	1.0	0.7	0.4	0.9	0.9	0.84
Fast-WAM	0.9	1.0	0.8	0.7	0.5	0.9	0.8	0.80
WAM4D	0.9	0.9	1.0	0.9	0.8	0.9	0.9	**0.90**

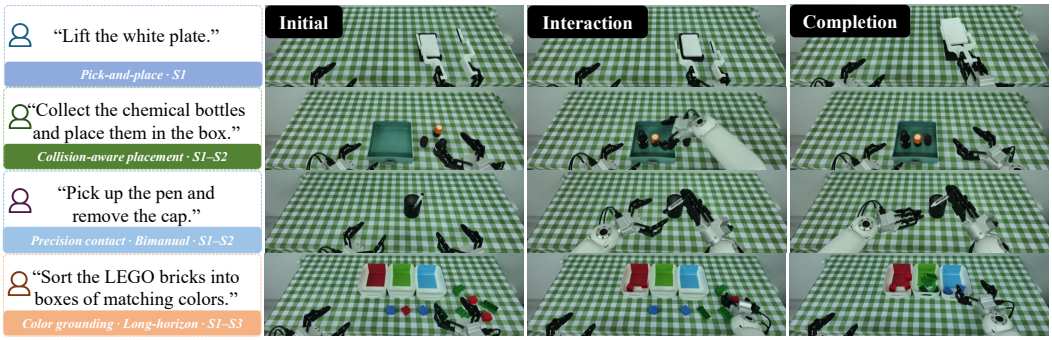


Figure 3: Real-world tasks on the AstriBot S1 platform.

4 Experiments

4.1 Experimental Setup

Datasets and Tasks. We evaluate WAM4D on simulation control, video-generation benchmarks, and real-world manipulation. For policy evaluation, we train a unified policy on the full RoboTwin 2.0 task suite under clean and randomized settings (Chen et al., 2025). For model-analysis ablations, we deliberately select a fixed ten-task hard diagnostic subset based on prior WAM baseline difficulty, while retaining diverse manipulation requirements. The task list was fixed before comparing the geometry-ablation variants. These ablations use 10k training steps, versus 50k for the full-task experiments; their success-rate differences characterize this selected, lower-budget setting rather than an average effect over all 50 tasks (Appendix H). For video and geometry evaluation, we train and test on RT-1 and Bridge-Data (Brohan et al., 2022; Walke et al., 2023; Ebert et al., 2021). For real-world evaluation, we use the AstriBot S1 robot on four tasks covering pick-and-place, precision alignment, and long-horizon sorting. RoboTwin uses simulator ground-truth depth as an auxiliary training target; RT-1, Bridge-Data, and the real-world demonstrations use offline Depth Anything 3 (DA3) pseudo-depth generated from RGB images. Action inference requires neither depth observations nor dense depth decoding (Appendix D).

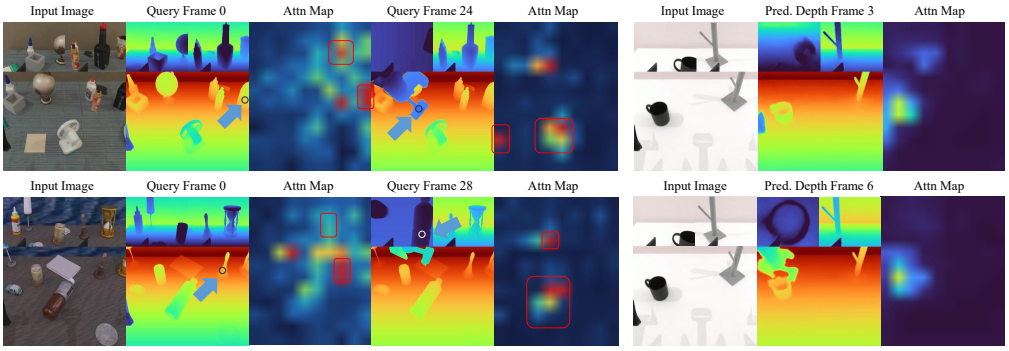
Metrics. We report success rate for simulation and real-world control, and sub-action success rates for real-world tasks. Following prior 4D generation evaluation (Zhen et al., 2025), the main tables report a compact set of video and geometry metrics: FVD, PSNR, LPIPS, AbsRel, δ_1, CD_1, F-score, and F-score-T. Full metric tables including SSIM, δ_2, and CD_2, are provided in Appendix.

4.2 Simulation Results on RoboTwin 2.0

We evaluate WAM4D on the full RoboTwin 2.0 task suite and report the average success rate over 50 tasks under both clean and randomized settings. The full per-task success rates are provided in Appendix. We compare WAM4D with strong VLA baselines and recent video-action models. The overall success rate is reported in Tab. 1. WAM4D achieves competitive performance while maintaining low inference cost. Latency uses 10 action denoising steps and 5 video denoising steps where applicable. For WAM4D, we use nine observed RGB frames and 32 state-history steps to generate a 32-action chunk. Timing includes history and spatial-register prefill plus action denoising from preencoded RGB; RGB/text encoding and network transport are excluded. VRAM is the peak allocated memory of the video-action backbone during inference. Appendix M details the measurement protocol.

Table 3: Selected video and geometry generation metrics on RT-1 and Bridge-Data.

Dataset	Method	FVD ↓	PSNR ↑	LPIPS ↓	AbsRel ↓	δ_1 ↑	CD_1 ↓	F-score ↑	F-score-T ↑
RT-1	LingBot-VA	300.3	20.04	0.160			–		
	X-WAM reimpl.	311.1	19.99	0.164	0.206	0.837	0.043	0.224	0.480
	WAM4D	**289.3**	**20.37**	**0.157**	**0.175**	**0.881**	**0.032**	**0.295**	**0.580**
Bridge	LingBot-VA	**294.6**	18.67	**0.178**			–		
	X-WAM reimpl.	301.5	18.51	0.186	0.148	0.842	0.044	0.250	0.487
	WAM4D	296.5	**18.76**	0.182	**0.106**	**0.907**	**0.031**	**0.334**	**0.582**



(a) Spatial-register attention over RGB features **(b) Action-query attention over spatial registers**

Figure 4: Attention visualization on RoboTwin. (a) Spatial-register queries attend to history RGB features in randomized scenes. (b) Predicted future scene depth and action-query attention over depth-aligned spatial registers in a hanging-mug example. Brighter regions indicate higher attention.

4.3 Real-World Results

We evaluate WAM4D on the AstriBot S1 platform using four different manipulation scenarios. Fig. 3 illustrates the evaluation setup, and the stage definitions are provided in Appendix. The quantitative results are reported in Tab. 2. WAM4D achieves the best overall performance across all four real-robot evaluations, demonstrating its robustness in contact-rich, geometry-sensitive, and long-horizon manipulation. LingBot-VA shows lower success rates in these settings. In its default code implementation, action predictions are anchored to the initial state history, which can cause global trajectory drift when the initial configuration changes. Its long history window may further introduce stale state information that conflicts with the current observation, reducing reliability during precise contact and long-horizon execution. In our tests, $\pi_{0.5}$ and Fast-WAM show a narrower effective workspace, failing more often when object positions move farther from the nominal region.

4.4 Video and Geometry Quality

To compare depth-branch designs under the same evaluation protocol, we reimplement X-WAM (Guo et al., 2026) within our framework using a 4-layer depth branch. Tab. 3 shows that WAM4D achieves the best RGB video metrics on RT-1 among the three models, while its RGB metrics on Bridge-Data are close to LingBot-VA. X-WAM reimpl. achieves slightly lower video quality than WAM4D, and its depth metrics are clearly worse. This gap suggests that the pretrained-initialized DA3 depth decoder in WAM4D provides a more precise continuous depth readout, while directly discretizing depth into RGB-style channels introduces quantization error and reduces depth accuracy.

4.5 Model Analysis and Ablations

Unless stated otherwise, ablations use matched data, backbone size, history length, action horizon, optimizer, and training budget. The environmental-perturbation comparison tests the complete depth-supervised spatial pathway against an RGB-only counterpart, rather than isolating its individual components. We therefore evaluate both generation quality and control performance, since a lower

Table 4: RoboTwin policy success (%) under clean and perturbed conditions. Each completed condition pools 50 paired episodes per task.

Condition	RGB-only WAM	4D-WAM-reimpl	WAM4D
Clean	32.5	30.5	**35.0**
Table height	21.5	**23.0**	**23.0**
Texture	18.5	16.0	**24.0**
Clutter	**19.5**	15.5	**19.5**
Dynamic lighting	30.5	29.5	**35.5**
Average	24.5	22.9	**27.4**

Table 5: Frozen-feature action readout on four RoboTwin tasks at 5k backbone steps. Probes train on clean features only. Entries are normalized MSE for a 32-step action chunk, averaged over three probe seeds. All probes share state history and instruction.

Probe configuration	Appearance Clean ↓	shift ↓	Randomized ↓
State + instruction	0.0236	0.0236	**0.0271**
+RGB feat. (RGB WAM)	0.0197	0.0297	0.0745
+Depth feat. (WAM4D)	**0.0154**	**0.0167**	0.0316

depth loss alone does not show that geometry improves manipulation. Additional settings and raw per-task counts for the ten-task ablations are provided in Appendix.

4.5.1 Robustness to Environmental Perturbations

We compare WAM4D with an RGB-only WAM and fair reimplementated 4D-WAM, denoted 4D-WAM-reimpl. The RGB-only WAM removes both depth supervision and the spatial-register pathway. We train all models on the same clean RoboTwin demonstrations, using matched appearance augmentation and training schedules for 10k steps. Four tasks are preselected covering articulated motion, placement, and bimanual interaction: open microwave, hanging mug, handover block, and turn switch. Evaluation comprises a clean reference and four environment perturbations: table height, texture, clutter, and dynamic lighting. each task–condition pair are evaluated 50 episodes.

As shown in Table 4, WAM4D achieves 27.4% average success across all five conditions, outperforming the RGB-only WAM (24.5%) and 4D-WAM-reimpl (22.9%) by 2.9 and 4.5 percentage points, respectively. Gains over both baselines are largest under texture and dynamic lighting, indicating improved policy performance particularly under appearance perturbations, although the benefits are not uniform across tasks and conditions. The full per-task breakdown, including evaluated denominators, is reported in Appendix Table 8.

4.5.2 Action-Relevant Information under Appearance Changes

To assess whether the shared-backbone spatial registers retain action-relevant information under appearance changes, we freeze matched 5k-step checkpoints of WAM4D and the RGB-only WAM trained without depth supervision described in Section 4.5.1. Equal-capacity probes read either RGB-history tokens from the RGB-only WAM (+RGB feat.) or spatial-register tokens from WAM4D (+Depth feat.), extracted at layers 12, 14, 16, and 18. Both probe variants additionally receive the same state history and episode instruction. All probes are trained exclusively on clean features; Table 5 reports action-prediction errors on held-out episodes, averaged over three probe seeds. For appearance-shift evaluation, we apply appearance augmentations to clean held-out images while keeping the corresponding state histories and instructions unchanged. We further evaluate on RoboTwin randomized demonstrations, which neither WAM encounters during training. Appendix H.3 details the data split and paired evaluation protocol.

On clean held-out episodes, both feature-based probes outperform the state-and-instruction-only baseline. Under both appearance shifts and randomized evaluation, +Depth feat. achieves lower action-prediction errors than +RGB feat. Together, these results suggest that WAM4D's spatial registers retain action-relevant information that is more reliably decoded under visual distribution shifts than that encoded in the RGB-history features of the RGB-only WAM.

4.5.3 Spatial Register Design Ablation

Appendix H reports the full spatial-register design ablations. Middle-layer registers give the best depth and point-cloud metrics among the unidirectional placements, while bidirectional visibility improves clean control at the expense of most geometry metrics.

Table 6: DA3 depth decoder initialization ablations on the RoboTwin ten-task hard diagnostic subset.

Variant	Policy Performance		Video Quality			Depth Quality				
	Clean SR	Rand. SR	FVD ↓	PSNR ↑	LPIPS ↓	AbsRel ↓	δ_1 ↑	CD_1 ↓	F-score ↑	F-score-T ↑
w/o depth	71.7	69.1	181.2	20.39	0.165			–		
Trainable random init	70.0	74.8	189.8	20.22	0.162	0.059	0.941	0.0129	0.589	0.737
Trainable pretrained	**80.1**	**75.4**	**164.5**	**21.13**	**0.143**	**0.049**	**0.948**	**0.0099**	**0.710**	**0.848**
Fixed pretrained	75.2	69.7	179.8	20.67	0.159	0.053	0.945	0.0108	0.685	0.825

4.5.4 DA3 Depth Decoder Initialization Ablation

We compare the no-depth baseline with DA3 depth decoder initialization and training choices under the same ten-task evaluation protocol. A randomly initialized DA3 depth decoder tests whether ordinary depth prediction is sufficient. A trainable, pretrained-initialized DA3 depth decoder tests whether adaptation is needed. A frozen, pretrained-initialized DA3 depth decoder tests whether geometric foundation knowledge is sufficient without adaptation.

Tab. 6 shows that the trainable pretrained DA3 decoder achieves the best policy, video, and geometry metrics on the ten-task, lower-budget diagnostic subset, improving clean and randomized success over the no-depth baseline by 8.4 and 6.3 percentage points, respectively. Its advantage over both random initialization and a frozen pretrained decoder supports combining pretrained geometric priors with adaptation during training. However, the frozen decoder produces better geometry than random initialization but lower randomized success, indicating that geometric accuracy alone does not guarantee better control. We therefore adopt the trainable pretrained decoder.

In a separate test on the same subset, replacing simulator ground-truth depth with offline DA3 pseudo-depth yields 77.2% clean success, compared with 80.1% using ground-truth depth and 71.7% without depth supervision. Thus, RGB-derived auxiliary targets retain a policy benefit in this setting without requiring measured depth.

4.5.5 Spatial Register Attention Analysis

Fig. 4(a) shows register-to-history attention on randomized RoboTwin samples. Spatial registers attend to corresponding objects across views, consistent static background regions, and observed gripper locations, suggesting that future-depth prediction draws on geometry-related cues from visual history. Fig. 4(b) shows predicted future depth alongside action-query attention over depth-aligned registers. Attention concentrates around target objects and anticipated gripper positions. This alignment shows that depth-supervised registers expose action-relevant spatial cues to the policy.

5 Conclusion and Limitation

We present WAM4D, a 4D world action model that transfers geometric foundation priors into causal video-action representations through spatial register tokens. Experiments show that WAM4D improves spatial consistency and action prediction across simulation, video-generation benchmarks, and real-world long-horizon tasks. In frozen-feature study, its register readout retains action-predictive information on clean tasks and is less sensitive to paired appearance changes than the RGB-only WAM readout, even under broader unseen randomized shifts. Depth supervision adds training cost and, when measured depth is unavailable, requires offline pseudo-depth generation. These targets inherit teacher errors and temporal inconsistency, and we have not validated training on Internet-scale RGB datasets. The model also lacks persistent object memory: after prolonged occlusion, autoregressive rollouts can change an object's identity (Appendix K). Action inference avoids dense depth decoding, but WAM inference remains slower than the VLA baselines considered here.

AI Use Statement

We used generative AI tools to assist with running experiments, monitoring training and evaluation progress, and improving the clarity and readability of the manuscript. The authors supervised the experimental workflow and reviewed the AI-assisted outputs. We take full responsibility for the experimental procedures, reported results, their interpretation, and the final content of this paper.

REFERENCES

Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Tobias Pfaff, Cheston Tan, Henrik I Christensen, Hao Su, Jiajun Wu, and Yunzhu Li. A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 10(106):eadt1497, 2025.

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

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

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

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Shizhe Cai, Zeya Yin, Jayadeep Jacob, and Fabio Ramos. Q-guided stein variational model predictive control via rl-informed policy prior. *arXiv preprint arXiv:2507.06625*, 2025.

Jiajun Cao, Qizhe Zhang, Peidong Jia, Xuhui Zhao, Bo Lan, Xiaoan Zhang, Xiaobao Wei, Sixiang Chen, Liyun Li, Xianming Liu, et al. Fastdrivevla: Efficient end-to-end driving via plug-and-play reconstruction-based token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2571–2579, 2026.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Yuzhi Chen, Ronghan Chen, Dongjie Huo, Yandan Yang, Dekang Qi, Haoyun Liu, Tong Lin, Shuang Zeng, Junjin Xiao, Xinyuan Chang, et al. Abot-physworld: Interactive world foundation model for robotic manipulation with physics alignment. *arXiv preprint arXiv:2603.23376*, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025a.

Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025b.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys*, 58(3):1–38, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-Anything: Automating real-world robots using vision-language models. *arXiv preprint arXiv:2406.18915*, 2024.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

Chun-Kai Fan, Xiaowei Chi, Xiaozhu Ju, Hao Li, Yong Bao, Yu-Kai Wang, Lizhang Chen, Zhiyuan Jiang, Kuangzhi Ge, Ying Li, et al. Wow, wo, val! a comprehensive embodied world model evaluation turing test. *arXiv preprint arXiv:2601.04137*, 2026.

Xueji Fang, Boqiang Duan, Hua Wu, Jingdong Wang, and Guo-Jun Qi. Latent evolving world action model. *arXiv preprint arXiv:2609.27455*, 2026.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D world action modeling from video priors with asynchronous denoising. *arXiv preprint arXiv:2604.26694*, 2026.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Yue Liao, Zhengkai Jiang, Yue Hu, Peng Gao, Hongsheng Li, Maoqing Yao, et al. Enerverse: Envisioning embodied future space for robotics manipulation. *Advances in Neural Information Processing Systems*, 38:37693–37720, 2026.

Adam Hung, Bardienus P. Duisterhof, Deva Ramanan, and Jeffrey Ichnowski. Modality-autoregressive world-action models. *arXiv preprint arXiv:2609.17524*, 2026.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. Pi0.7: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

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

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

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11(3):2506–2513, 2026a.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

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

Ying Li, Xiaobao Wei, Xiaowei Chi, Yuming Li, Zhongyu Zhao, Hao Wang, Ningning Ma, Ming Lu, and Sirui Han. Manipdreamer3d: Synthesizing plausible robotic manipulation video with occupancy-aware 3d trajectory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6644–6652, 2026c.

Ying Li, Xiaobao Wei, Xiaowei Chi, Yuming Li, Zhongyu Zhao, Hao Wang, Ningning Ma, Ming Lu, and Shanghang Zhang. Manipdreamer: Boosting robotic manipulation world model with action tree and visual guidance. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12027–12031. IEEE, 2026d.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Xiao Liu, Yuguang Yang, Xi Wang, Kai Jiang, Cheng Chi, Yong Xu, Wenchao Ding, Yilun Chen, and Yan Wang. StageWAM: Joint-embedding stage prediction for world-action models in robot manipulation. *arXiv preprint arXiv:2608.10780*, 2026a.

Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. Last _$\{0\}$: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026b.

Javier Alejandro Lopetegui Gonzalez, Paul Pacaud, and Cordelia Schmid. Spatially aware world action model via geometric latent diffusion. *arXiv preprint arXiv:2609.02531*, 2026.

Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. ManiGaussian: Dynamic gaussian splatting for multi-task robotic manipulation. *arXiv preprint arXiv:2403.08321*, 2024.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

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

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, et al. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

Ola Shorinwa, Johnathan Tucker, Aliyah Smith, Aiden Swann, Timothy Chen, Roya Firoozi, Monroe Kennedy III, and Mac Schwager. Splat-MOVER: Multi-stage, open-vocabulary robotic manipulation via editable gaussian splatting. *arXiv preprint arXiv:2405.04378*, 2024.

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025.

MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. *arXiv preprint arXiv:2604.27792*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, et al. BridgeData V2: A dataset for robot learning at scale. *Conference on Robot Learning*, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Boyang Wang, Nikhil Sridhar, Chao Feng, Mark Van der Merwe, Adam Fishman, Nima Fazeli, and Jeong Joon Park. This&that: Language-gesture controlled video generation for robot planning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12842–12849. IEEE, 2025a.

Hao Wang, Xiaobao Wei, Xiaoan Zhang, Jianing Li, Chengyu Bai, Ying Li, Ming Lu, Wenzhao Zheng, and Shanghang Zhang. Embodiedocc++: Boosting embodied 3d occupancy prediction with plane regularization and uncertainty sampler. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 925–934, 2025b.

Hao Wang, Xiaobao Wei, Jingyang He, Chengyu Bai, Chun-Kai Fan, Jiajun Cao, Jintao Chen, Ying Li, Shanyu Rong, Ming Lu, et al. Vega: Visual encoder grounding alignment for spatially-aware vision-language-action models. *arXiv preprint arXiv:2605.10485*, 2026a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025c.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-omega. *arXiv preprint arXiv:2605.15195*, 2026b.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, et al. Mvista-4d: View-consistent 4d world model with test-time action inference for robotic manipulation. *arXiv preprint arXiv:2602.09878*, 2026c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024.

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

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025d.

Xiaobao Wei, Peng Chen, Guangyu Li, Ming Lu, Hui Chen, and Feng Tian. Gazegaussian: High-fidelity gaze redirection with 3d gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13293–13303, 2025a.

Xiaobao Wei, Qingpo Wuwu, Zhongyu Zhao, Zhuangzhe Wu, Nan Huang, Ming Lu, Ningning Ma, and Shanghang Zhang. Emd: Explicit motion modeling for high-quality street gaussian splatting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 28462–28472, 2025b.

Xiaobao Wei, Zhangjie Ye, Yuxiang Gu, Zunjie Zhu, Yunfei Guo, Yingying Shen, Shan Zhao, Ming Lu, Haiyang Sun, Bing Wang, et al. Parkgaussian: Surround-view 3d gaussian splatting for autonomous parking. *arXiv preprint arXiv:2601.01386*, 2026.

Dongli Wu, Jingyu Hu, Ka-Hei Hui, Xiaobao Wei, Chengwen Luo, Jianqiang Li, and Zhengzhe Liu. Phymix: Towards physically consistent single-image 3d indoor scene generation with implicit–explicit optimization. *arXiv preprint arXiv:2604.10125*, 2026a.

Dongli Wu, Zhuoxiao Li, Tongyan Hua, Yinrui Ren, Xiaobao Wei, Rongjun Qin, and Wufan Zhao. Feed-forward gaussian splatting from sparse aerial views. *arXiv preprint arXiv:2605.19949*, 2026b.

Gangwei Xu, Qihang Zhang, Jiaming Zhou, Xing Zhu, Yujun Shen, Xin Yang, and Yinghao Xu. Next forcing: Causal world modeling with multi-chunk prediction. *arXiv preprint arXiv:2606.11187*, 2026a.

Mutian Xu, Tianbao Zhang, Tianqi Liu, Zhaoxi Chen, Xiaoguang Han, and Ziwei Liu. Kinema4d: Kinematic 4d world modeling for spatiotemporal embodied simulation. *arXiv preprint arXiv:2603.16669*, 2026b.

Ge Yan, Jinghao Liu, Yuzhi Fan, Lei Cai, Minwen Liao, Jesse Zhang, and Dieter Fox. Flex-π: A multi-stream world-action model with compute flexibility. *arXiv preprint arXiv:2608.10860*, 2026.

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

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026b.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. GigaWorld-Policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026a.

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

Chenghao Yin, Da Huang, Di Yang, Jichao Wang, Nanshu Zhao, Chen Xu, Wenjun Sun, Linjie Hou, Zhijun Li, Junhui Wu, et al. Genie sim 3.0: A high-fidelity comprehensive simulation platform for humanoid robot. *arXiv preprint arXiv:2601.02078*, 2026.

Zeya Yin, Tin Lai, Subhan Khan, Jayadeep Jacob, Yonghui Li, and Fabio Ramos. Stein movement primitives for adaptive multi-modal trajectory generation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11901–11908. IEEE, 2024.

Zeya Yin, Tin Lai, Lucas Barcelos, Jayadeep Jacob, Yonghui Li, and Fabio Ramos. Diverse motion planning with stein diffusion trajectory inference. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15610–15616. IEEE, 2025.

Tengbo Yu, Guanxing Lu, Zaijia Yang, Haoyuan Deng, Season Si Chen, Jiwen Lu, Wenbo Ding, Guoqiang Hu, Yansong Tang, and Ziwei Wang. Manigaussian++: General robotic bimanual manipulation with hierarchical gaussian world model. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12232–12239. IEEE, 2025.

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

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

Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. GNFactor: Multi-task real robot learning with generalizable neural feature fields. *Conference on Robot Learning*, 2023.

Kai Zeng, Zhanqian Wu, Kaixin Xiong, Xiaobao Wei, Xiangyu Guo, Zhenxin Zhu, Kalok Ho, Lijun Zhou, Bohan Zeng, Ming Lu, Haiyang Sun, BING WANG, Guang Chen, Hangjun Ye, and Wentao Zhang. Rethinking driving world model as synthetic data generator for perception tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=z3cFADf6zZ.

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

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026b.

Yi Zhang, Yinda Chen, Che Liu, Zeyuan Ding, Jin Xu, Shilong Zou, Junwei Liao, Jiayu Hu, Xiancong Ren, Xiaopeng Zhang, et al. Pelican-unified 1.0: A unified embodied intelligence model for understanding, reasoning, imagination and action. *arXiv preprint arXiv:2605.15153*, 2026c.

Haoyu Zhao, Xingyue Zhao, Siteng Huang, Xin Li, Deli Zhao, and Zhongyu Li. RynnWorld-4D: 4D embodied world models for robotic manipulation. *arXiv preprint arXiv:2607.06559*, 2026.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. TesserAct: Learning 4D embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

Kaichen Zhou, Yuzhen Chen, Fangneng Zhan, Hang Hua, Grace Chen, Xinhai Chang, Ao Qu, Yilun Du, Zhuang Liu, Paul Pu Liang, and Mengyu Wang. GEM-4D: Geometry-enhanced video world models for robot manipulation. *arXiv preprint arXiv:2605.22882*, 2026.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Additional Related Work

Geometric Foundation Models. Explicit geometry provides important priors for embodied perception and control. One line of work represents scenes with optimized 3D structures, such as neural radiance fields and 3D Gaussian primitives, which support view synthesis, planning, data generation, and manipulation (Mildenhall et al., 2020; Kerbl et al., 2023; Ze et al., 2023; Lu et al., 2024; Shorinwa et al., 2024; Duan et al., 2024; Huang et al., 2024; Wei et al., 2025b; 2026). Another line develops feed-forward geometric foundation models that recover depth, point maps, camera poses, and dense correspondence directly from images (Wang et al., 2024; 2025c; Lin et al., 2025; Wang et al., 2025d; Wei et al., 2025a; Wang et al., 2025b; Wu et al., 2026b; Wang et al., 2026b). These models enable scalable geometric supervision from large-scale robot videos. Recent embodied

world models further inject geometry into future prediction by generating RGBD videos, normals, point maps, 3D trajectories, or 4D robot world interactions (Zhen et al., 2025; Shen et al., 2026; Huang et al., 2026; Yang et al., 2026b; Xu et al., 2026b). Collectively, these methods indicate that explicit geometry improves physical consistency and spatial reasoning. However, they often require geometry to appear as an input, output, or reconstructed scene state during generation. WAM4D instead uses future depth as a training signal through spatial registers.

B Datasets and Evaluation Protocols

RoboTwin 2.0. A unified policy is trained on the complete RoboTwin 2.0 task suite. The clean setting comprises relatively uncluttered scenes, whereas the randomized setting introduces variation in clutter, illumination, background, tabletop height, object placement, and language instruction. Geometry-supervised training uses re-collected RoboTwin demonstrations with depth annotations. Each task contains 50 clean trajectories and 500 randomized trajectories.

RT-1 and Bridge-Data. Video and geometry prediction quality are evaluated on RT-1 and Bridge-Data (Brohan et al., 2022; Walke et al., 2023; Ebert et al., 2021). For each benchmark, 100 trajectories are held out for testing and the remaining trajectories are used for training. As these datasets do not provide aligned metric depth, pseudo-depth supervision is generated with Depth Anything 3 following Sec. D.

Real-world tasks. Real-world evaluation is conducted on the AstriBot S1 robot across four manipulation tasks: plate lifting, bottle placement, pen cap removal, and LEGO sorting. The dataset contains 100 demonstrations per task, yielding 400 demonstrations in total. Each method is evaluated with 10 physical rollouts per task. Depth supervision for real-world demonstrations is likewise obtained with Depth Anything 3 following the same offline pseudo-depth pipeline.

C Implementation Details

Frame sampling strategy. The video-action backbone uses the Wan2.2 video VAE. We sample at most 17 frames from each sequence. A start frame id is randomly sampled over the sequence; when enough preceding frames are available, the history context contains 1, 5, or 9 frames. After the start frame, 8 video-prediction target frames are sampled every 4 collected steps to match the VAE temporal stride. If the remaining sequence is shorter than 8 target frames, the targets are padded to length 8. For the latent video loss, only slots that can be encoded as valid VAE latents contribute to the loss, and latent slots containing padding frames are masked. Depth targets use the same timestamps as the sampled video targets; unlike the latent-level video loss, depth supervision is frame-level, so every valid depth frame contributes to the depth loss. The default action chunk size is 32. For RoboTwin and AstriBot S1 experiments, the input is represented as a three-view mosaic consisting of one head camera and two wrist cameras. The main view is resized to 256×320, and each wrist view is resized to 128×160.

Action Prediction. The policy predicts only the absolute end-effector poses of the left and right arms. Each arm is represented by a 3D position, a quaternion, and one gripper open/close value, giving a 16-dimensional action vector in total. Position channels are normalized with percentile-based min–max normalization, i.e., quantile min–max normalization, using dataset-level q_{01} and q_{99} statistics. Quaternion and gripper-opening channels use fixed lower and upper bounds of -1 and 1. All normalized action channels are represented in the range $[-1, 1]$.

View layout strategy. Spatial registers are aligned with the pixel layout of the mosaic. Given the VAE spatial stride of 16 and the WAM transformer's 2×2 latent grouping, each register corresponds to a 32×32 cell in the input image. The main view contributes an 8×10 register grid, each wrist view contributes a 4×5 register grid, and the tiled three-view mosaic forms a 12×10 grid per coarse future slot. With two coarse future slots, the current three-view model uses $2 \times 120 = 240$ spatial registers; the eight-slot, 960-register design is a separate ablation. A shared temporal upsampler converts the coarse register features through $2 \to 4 \to 8$ slots before the DA3 depth decoder. Spatial registers traverse and update in all backbone layers using the existing self-attention and feed-forward parameters. Layer outputs 12, 14, 16, and 18 are readout taps for depth decoding and geometry consistency loss, not the only update layers. Register queries attend only to valid history video tokens

and registers themselves, not to action tokens or future-video tokens. No separate register cross-attention projections are introduced. For the RT-1 and Bridge-Data video/depth quality experiments, only the main camera view is used during training and testing.

Depth readout. A shared temporal upsampler converts the four spatial-register feature maps from two coarse slots to eight depth-frame slots. Four learned linear adapters then map them to the input dimension of the `DA3-GIANT-1.1` any-view DualDPT depth decoder, which is initialized from pretrained weights.

Attention mask. The main transformer token groups are history-video tokens, future-video noise tokens, state-history tokens, future-action noise tokens, and spatial registers. Their principal self-attention paths within each sample are:

Query token group	Principal key/value token groups
Future action noise	History video, state history, future action noise, register
Spatial register	Spatial register, history video
Future video noise	History video, future video noise
History video	History video
History state	History state

Inference procedure. At deployment, history prefill still processes spatial registers in all backbone layers and caches their layer-wise features for action denoising. The temporal upsampler, adapters, and DA3 depth decoder can be skipped. The policy maintains an observation queue and a robot state history. In RoboTwin, states and actions share the same coordinate representation and align with a one-step temporal offset; we use state history to denote the historical condition and actions to denote future control targets. The inference loop is:

Algorithm 1: Deployment inference loop

$Q_{\text{obs}} \leftarrow \emptyset$, $S_{\text{hist}} \leftarrow \emptyset$
while policy is running **do**
  if $Q_{\text{obs}} = \emptyset$ **then** capture the current observation and enqueue it into Q_{obs}
  Encode Q_{obs} with the VAE to obtain video-context latents
  Prefill history video and spatial registers; refresh their layer-wise KV caches
  if $S_{\text{hist}} \neq \emptyset$ **then** encode it and cache the state keys and values
  Denoise future action tokens using history-video, state, and register KV to obtain an action chunk
  Execute the action chunk
  During execution, capture one observation every 4 actions and enqueue it into Q_{obs}
  $S_{\text{hist}} \leftarrow$ recent measured robot states collected during execution
end while

Training Hyperparameters. Unless stated otherwise, the loss weights are set to $\lambda_{\text{act}} = 1$ and $\lambda_{\text{depth}} = 1$. The shared-backbone WAM4D and RGB-only baseline use AdamW with explicit parameter-group learning rates: 2×10^{-5} for the shared video backbone, 10^{-4} for the action DiT/actor, and 2×10^{-5} for the action output head. WAM4D additionally uses 10^{-4} for the spatial-register/depth parameters and DA3 depth decoder; the RGB-only baseline has no depth parameter group. These runs use 10 optimizer warmup steps, gradient clipping at 2.0, and bf16 parameters. The reported full-task experiments use a 50k-step budget, and the ten-task ablations use a 10k-step budget.

D Pseudo-depth Label Generation

RoboTwin depth targets use simulator ground-truth depth. For RT-1, Bridge-Data, and real-world demonstrations, the generated depth maps are used only as auxiliary training targets. They are never provided as observations to the policy; action inference uses spatial-register features computed from RGB history without ground-truth depth. Thus, pseudo-depth supervision shapes the shared visual representation and spatial-register features during training. At inference, action queries read the depth-supervised register keys and values computed from RGB history, without using pseudo-depth as an additional sensing channel.

We process all frames offline using the frozen Depth Anything 3 (DA3) teacher. The predicted depth maps are resized back to the training resolution, converted to the same depth convention as simulator depth, and stored in uint16-mm format. Invalid or out-of-range pixels are excluded from the SmoothL1 depth loss. The same pseudo-depth generation pipeline is used for all methods that require depth supervision, ensuring that comparisons are not affected by different teacher labels.

On RT-1 and Bridge-Data, depth and point-cloud metrics are computed with respect to the same frozen DA3 pseudo-depth targets, because aligned metric depth is unavailable. These metrics therefore measure consistency with a strong geometric teacher rather than absolute metric-depth accuracy.

E Real-World Sub-Action Definitions

Appendix Table 7 defines the sub-actions used for real-robot evaluation on the AstriBot S1 platform. During each rollout, a sub-action is counted as successful when the robot completes the corresponding intermediate goal. Because the tasks are sequential, if any sub-action fails, subsequent sub-actions are not attempted and are recorded as 0.

Table 7: Sub-action definitions for the AstriBot S1 real-world tasks.

Task	Step	Success criterion
Plate	S1	Grasp and lift the white plate.
Bottles	S1	Grasp a chemical bottle.
	S2	Place the grasped bottle into the tray.
Lego	S1	Place the first colored Lego block into its matching box.
	S2	Place the second colored Lego block into its matching box.
	S3	Place the third colored Lego block into its matching box.
Pen	S1	Grasp the pen.
	S2	Remove the pen cap from the pen.

F Baseline Protocols

The RoboTwin evaluation compares WAM4D with VLA baselines (π_0, $\pi_{0.5}$) and recent video-action models, including Motus, LingBot-VA, and Fast-WAM. Whenever possible, reproduced baselines use the same training and evaluation splits. They also follow the same camera configuration and training budget. For latency comparisons, all models use 10 action-denoising steps, and video-action models use 5 video-denoising steps when applicable. WAM4D retains spatial registers for action prediction and skips dense depth decoding.

G Metric Definitions

Policy performance is reported as the fraction of successful rollouts. For real-world tasks, sub-action success records whether each task-specific intermediate goal is completed.

Video quality is evaluated with FVD, PSNR, SSIM, and LPIPS. FVD measures the distribution gap between generated and ground-truth videos in a learned video-feature space; lower values indicate more realistic temporal generation. PSNR measures pixel-level reconstruction fidelity, SSIM measures structural image similarity, and LPIPS measures perceptual distance in deep visual features.

Depth quality is evaluated with AbsRel and threshold accuracy δ_1 and δ_2 with respect to the available reference depth. AbsRel is the mean relative absolute depth error, so lower is better. δ_1 and δ_2 report the fraction of pixels whose predicted depth is within fixed multiplicative error thresholds of the target depth, so higher is better.

For point-cloud metrics, predicted and target depth maps are back-projected using a fixed dataset-level camera intrinsic matrix. This protocol yields comparable point clouds for all methods. CD_1 and CD_2 denote L1 and L2 Chamfer Distance variants between predicted and target point clouds. F-score measures thresholded point-cloud overlap with the target geometry, while F-score-T measures the temporal consistency of this overlap across predicted future frames. On RT-1 and Bridge-Data, depth

and point-cloud metrics are computed with respect to the same frozen DA3 pseudo-depth targets, because aligned metric depth is unavailable. These metrics therefore measure consistency with a strong geometric teacher rather than absolute metric-depth accuracy.

H Experiment Results

H.1 Environmental-Perturbation Evaluation Protocol

We evaluate open microwave, hanging mug, handover block, and turn switch under five separate conditions: clean, table height, texture, clutter, and dynamic lighting. Each task–condition cell targets 50 completed rollouts from the fixed development scene manifests. Clean is the reference condition. The four stress conditions each enable only their named factor; all other factor settings, including the fixed head and wrist cameras, remain at their clean values. The frozen RoboTwin task configurations are otherwise identical, and its native coupled domain-randomization switches are disabled.

Perturbations. Table height lowers the tabletop by 1, 2, or 3 cm relative to clean, cycling through these offsets by base-scene index; the cameras do not move with the table. Texture replaces only the tabletop material with one selected from the unseen texture pool (IDs 0–999, selected by the condition's domain seed); the wall and task-object textures are unchanged. Clutter requests ten additional objects sampled from RoboTwin's available clutter-object pool using a separate clutter seed. Objects are placed on the tabletop within the nominal $x \in [-0.59, 0.59]$ m and $y \in [-0.34, 0.34]$ m sampling region, subject to table-boundary and inter-object clearance checks and exclusion zones around task objects enlarged by 2 cm. Placement allows up to 50 object-selection attempts, with up to 100 pose samples per attempt; a scene that cannot realize all ten objects is invalid. Dynamic lighting retains the clean light positions and directions but samples the initial RGB intensities of the directional and point lights independently in $[0.25, 1.0]$ and ambient RGB in $[0.30, 0.70]$. At every render update, each directional and point light receives a new RGB color in $[0, 1]$, while each ambient channel takes a random increment in $[-0.1, 0.1]$, clipped to $[0, 1]$. Lighting samples use a separate domain RNG and do not advance the task RNG.

Pairing and invalid scenes. For a given task and base-scene index, clean and all four perturbations share the same base-scene ID, task-initialization seed, and policy seed. Their domain seeds differ by condition, so only the specified perturbation is independently sampled. The common task seed pairs the nominal initial task and object sampling; actual post-settling poses can differ when a perturbation changes the scene physics. Each rollout is conditioned on its natural-language episode instruction generated from the realized task parameters. The manifest pairs those parameters through the task seed but does not store or enforce identical instruction wording across conditions. A policy failure after a valid setup counts as an evaluated failure. A scene-setup or instruction error is excluded from the evaluated denominator and recorded as a skipped seed; errors in the fixed-manifest runs stop the cell rather than triggering an unconstrained seed search. For the clutter results below, the separately identified development-refill manifest retains the original 50 rows and replaces only setup-invalid slots with deterministic reserve rows for the same task and slot. A replaced clutter scene is therefore no longer paired with the original clean scene at that index; it has a corresponding clean reserve row in the refill manifest. The reported denominator remains 50 evaluated rollouts per cell.

H.2 Environmental Perturbation Results

Appendix Table 8 gives the full task-level results underlying Table 4. Each entry is successes/evaluated episodes, with a target of 50 episodes per task and condition. A pooled condition percentage is shown only when all four tasks reached 50 evaluated episodes. All policies use the natural-language instruction of each episode as their text condition. Averaging all five conditions gives 24.5% for RGB-only WAM, 22.9% for 4D-WAM-reimpl, and 27.4% for WAM4D (245, 229, and 274 successes out of 1,000 evaluated episodes, respectively).

H.3 Frozen-Feature Action Readout Protocol

The feature readout in Table 5 uses four RoboTwin tasks: Open Microwave, Hanging Mug, Handover Block, and Turn Switch. For each task, all 50 clean demonstration episodes provide two action anchors. Source episodes, rather than anchors, are divided into 30 training, 10 validation, and 10

Table 8: Per-task RoboTwin success counts under five separate evaluation conditions (snapshot: 26 September 2026, 11:46 UTC). The clutter condition uses the separately identified development refill protocol: its first 50 scene rows match the fixed development manifest, and reserve rows replace scene-setup failures. All task–condition pairs are complete with 50 evaluated episodes.

Model	Condition	Open microwave	Hanging mug	Handover block	Turn switch	Average S.R.
RGB-only WAM	Clean	15/50	9/50	31/50	10/50	32.5%
	Table height	15/50	0/50	22/50	6/50	21.5%
	Texture	10/50	7/50	16/50	4/50	18.5%
	Clutter	14/50	6/50	12/50	7/50	19.5%
	Dynamic lighting	15/50	8/50	29/50	9/50	30.5%
4D-WAM-reimpl	Clean	18/50	8/50	26/50	9/50	30.5%
	Table height	18/50	2/50	22/50	4/50	23.0%
	Texture	10/50	3/50	15/50	4/50	16.0%
	Clutter	12/50	4/50	9/50	6/50	15.5%
	Dynamic lighting	15/50	7/50	31/50	6/50	29.5%
WAM4D	Clean	16/50	6/50	40/50	8/50	35.0%
	Table height	11/50	5/50	23/50	7/50	23.0%
	Texture	8/50	7/50	24/50	9/50	24.0%
	Clutter	9/50	7/50	17/50	6/50	19.5%
	Dynamic lighting	14/50	8/50	36/50	13/50	35.5%

test episodes per task, giving 240/80/80 anchors across the four tasks. An additional 20 episodes are sampled across each task's 500 randomized demonstrations, yielding 160 anchors used only for distribution-shift testing. Every anchor retains its own natural-language episode instruction, 32-step state history, and a normalized expert action chunk of 32 future steps with 16 action dimensions per step. The clean and v4-transformed histories share exactly the same state, instruction, and action label.

Both 5k-step backbones were trained on clean demonstrations with matched appearance-shift augmentation and are frozen for the probe study. At each of layers 12, 14, 16, and 18, the RGB-only WAM readout receives 360 RGB-history tokens and the WAM4D readout receives 240 coarse depth-register tokens. Identically structured probes use four learned attention queries per layer, retain all four pooled outputs, and predict the action chunk from pooled visual features, state history, and instruction embedding. A state-and-instruction probe supplies the no-visual reference. Each probe is trained from scratch on clean anchors with identical source-batch order and three initialization seeds (17, 18, 19). The probe learning rate is 10^{-4} except for its action output head at 2×10^{-5}; training lasts 500 updates and selects the checkpoint with minimum clean-validation MSE. Test appearance variants and randomized demonstrations are never used for probe training or checkpoint selection.

For the appearance test, a video-domain v4 transform changes the input images of a source anchor while retaining its expert action; it is an image perturbation, not a new simulator rollout. We first average the two anchors within each source episode, then average the three probe seeds, and compute 95% intervals by resampling episodes within each task. The clean-test MSE advantage of WAM4D over RGB-only WAM is 0.0043 with interval $[0.0029, 0.0059]$; the WAM4D visual gain over state and instruction alone is 0.0083 with interval $[0.0056, 0.0111]$. On randomized episodes, WAM4D has MSE 0.0316 versus 0.0271 for the state-and-instruction reference, so its visual features do not provide a positive incremental gain under this broader shift. These intervals condition on the fixed 5k backbone checkpoints and source datasets. The clean source episodes were part of backbone training, so episode holdout applies to the newly trained probes, and offline action regression does not establish closed-loop success or rule out partial loss of geometric information.

Feature diversity diagnostic. We additionally examine the frozen full-token features of the 40 clean test episodes per anchor position, without training another readout. Each layer's flattened token vector is normalized to unit length and the four layer Gram matrices are averaged. Effective rank is the entropy exponent of the Gram eigenvalues after subtracting each task's feature mean (maximum rank 36); pairwise cosine is averaged only over different episodes of the same task. The two anchor positions are evaluated separately and averaged. Appendix Table 9 shows that WAM4D has a lower effective rank and slightly higher inter-episode cosine than RGB-only WAM, while same-anchor

features remain much more similar after appearance change. Thus the registers are more compact and appearance-stable, but this diagnostic does not exclude partial information loss; their clean action-probe gain rules out only a completely action-uninformative collapse in this setting.

Table 9: Full-token feature diversity on held-out clean probe episodes. Same-anchor cosine compares clean and v4-transformed histories.

Feature	Effective rank ↑ (max 36)	Inter-episode cosine ↓	Same-anchor cosine ↑
RGB-only WAM	28.88	0.8727	0.8558
WAM4D registers	18.80	0.8829	0.9883

Table 10: RoboTwin ten-task hard diagnostic subset used for model-analysis ablations.

Simulation Task	Horizon
Handover Block	2
Hanging Mug	2
Move Stapler Pad	1
Open Microwave	1
Pick Diverse Bottles	2
Place Can Basket	2
Press Stapler	1
Put Object Cabinet	2
Stack Bowls Three	3
Turn Switch	1

Model-analysis ablations are conducted on a fixed ten-task hard diagnostic subset of RoboTwin 2.0. We deliberately selected tasks where prior WAM baselines showed difficulty, while preserving diverse manipulation requirements and keeping the ablations computationally tractable. The task list was fixed after inspecting prior baseline performance and before comparing the geometry-ablation variants. The 10k-step ablation budget is smaller than the 50k-step budget used for the full-task experiments, so the ten-task success-rate differences should not be interpreted as average gains over all 50 tasks. The split covers single-step and multi-step tasks involving hanging, articulated-object interaction, contact pressing, placement, handover, object selection, and stacking. Appendix Table 10 lists the tasks and horizons used by all ablation runs. Appendix Table 11 consolidates per-task success rates, model size, and full video/geometry metrics for the nine ten-task ablation runs. No Depth is an otherwise identical model trained under the same conditions without any depth branch or depth loss. VAE DH discretizes depth into 0–255 values and uses the VAE as a depth head to decode depth latents; its depth readout is taken from the output of the layer-26 VA block. All remaining variants use Spatial Registers as the depth readout interface. For the layer configurations in the spatial-register design ablation, Shallow, Middle, Deep, and Uniform use register updates after layers 2/4/6/8, 12/14/16/18, 22/24/26/28, and 6/12/18/24, respectively; Bi-dir uses layers 6/12/18/24 with bidirectional visibility. Rand. DA3 and Train. DA3 use the default middle-layer Spatial Register interface with the DA3 head trained from random and pretrained initialization, respectively. Within each summary or per-task row, the best entry is bolded and the second-best entry is underlined, with ties marked together.

Appendix Table 12 reports the full video-generation and geometry-quality results on RT-1 and Bridge-Data. The LingBot-VA rows include the RGB metrics reported in the main paper; because LingBot-VA does not generate depth, the depth and point-cloud metric columns are grouped as unavailable. For RT-1 and Bridge-Data, all depth and point-cloud metrics are evaluated against frozen DA3 pseudo-depth targets rather than aligned metric-depth ground truth.

I Full RoboTwin 2.0 Success Rates

We evaluate WAM4D on the full RoboTwin 2.0 task suite and compare it with advanced baselines, including Fast-WAM, LingBot-VA, π_0, $\pi_{0.5}$, and Motus. Appendix Table 13 reports the per-task success rates over 50 tasks under both Easy and Hard settings, corresponding to the clean and

Table 11: Per-task success rates and full ablation metrics for the RoboTwin ten-task runs.

Metric / Task	No Depth	Bi-dir	Deep	Middle	Shallow	Uniform	VAE DH	Rand. DA3	Train. DA3
Transformer Params	5.089B	5.841B	5.690B	5.690B	5.690B	5.690B	5.744B	5.690B	5.690B
Clean SR (100-eval tasks)	71.7%	76.6%	74.5%	75.2%	72.5%	70.6%	70.7%	70.0%	**80.1%**
Randomized SR (100-eval tasks)	69.1%	72.5%	67.4%	69.7%	66.2%	67.0%	68.6%	74.8%	**75.4%**
Video metrics									
FVD ↓	181.2	175.3	171.5	179.8	168.8	175.3	178.0	189.8	**164.5**
PSNR ↑	20.39	20.75	20.63	20.67	**21.19**	20.57	20.72	20.22	21.13
SSIM ↑	0.795	0.798	0.797	0.798	0.808	0.798	0.802	0.794	**0.811**
LPIPS ↓	0.165	0.155	0.158	0.159	**0.143**	0.162	0.153	0.162	**0.143**
Depth and point-cloud metrics									
δ_1 ↑		0.916	0.931	0.945	0.940	0.941	0.901	0.941	**0.948**
δ_2 ↑		0.952	0.963	0.972	0.971	0.971	0.950	0.973	**0.974**
AbsRel ↓		0.074	0.064	0.053	0.058	0.058	0.081	0.059	**0.049**
CD_1 ↓	–	0.0151	0.0130	0.0108	0.0128	0.0121	0.0171	0.0129	**0.0099**
CD_2 ↓		0.0007	0.0005	0.0004	0.0005	0.0005	0.0009	0.0005	**0.0003**
F-score ↑		0.579	0.621	0.685	0.613	0.652	0.519	0.589	**0.710**
F-score-T ↑		0.769	0.776	0.825	0.784	0.824	0.685	0.737	**0.848**
Clean									
Handover Block	97/100	98/100	**100/100**	98/100	**100/100**	98/100	99/100	78/100	**100/100**
Hanging Mug	58/100	57/100	82/100	79/100	56/100	53/100	60/100	45/100	**85/100**
Move Stapler Pad	66/100	69/100	69/100	67/100	**76/100**	56/100	51/100	49/100	65/100
Open Microwave	39/100	69/100	45/100	53/100	44/100	50/100	40/100	53/100	**78/100**
Pick Diverse Bottles	**94/100**	92/100	91/100	90/100	91/100	91/100	93/100	93/100	**94/100**
Place Can Basket	68/100	72/100	67/100	**81/100**	64/100	60/100	77/100	69/100	75/100
Press Stapler	95/100	94/100	94/100	96/100	93/100	95/100	**99/100**	92/100	97/100
Put Object Cabinet	54/100	52/100	55/100	46/100	47/100	49/100	48/100	66/100	**87/100**
Stack Bowls Three	78/100	85/100	77/100	81/100	**88/100**	81/100	79/100	70/100	74/100
Turn Switch	68/100	78/100	65/100	61/100	66/100	73/100	61/100	**85/100**	46/100
Average	71.7%	76.6%	74.5%	75.2%	72.5%	70.6%	70.7%	70.0%	**80.1%**
Randomized									
Handover Block	75/100	95/100	79/100	81/100	70/100	77/100	72/100	**97/100**	**97/100**
Hanging Mug	66/100	69/100	72/100	64/100	66/100	71/100	72/100	79/100	**81/100**
Move Stapler Pad	**61/100**	46/100	49/100	55/100	58/100	54/100	57/100	49/100	49/100
Open Microwave	41/100	78/100	34/100	53/100	35/100	31/100	44/100	**80/100**	79/100
Pick Diverse Bottles	80/100	89/100	**91/100**	87/100	80/100	88/100	80/100	90/100	89/100
Place Can Basket	**75/100**	72/100	72/100	68/100	74/100	71/100	73/100	72/100	**75/100**
Press Stapler	94/100	95/100	93/100	95/100	94/100	94/100	95/100	97/100	**98/100**
Put Object Cabinet	43/100	47/100	47/100	48/100	45/100	45/100	**51/100**	49/100	37/100
Stack Bowls Three	**88/100**	75/100	85/100	82/100	75/100	80/100	81/100	79/100	87/100
Turn Switch	**68/100**	59/100	52/100	64/100	65/100	59/100	61/100	56/100	62/100
Average	69.1%	72.5%	67.4%	69.7%	66.2%	67.0%	68.6%	74.8%	**75.4%**

Table 12: Full video and geometry generation metrics on RT-1 and Bridge-Data.

Dataset	Method	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	δ_1 ↑	δ_2 ↑	AbsRel ↓	CD_1 ↓	CD_2 ↓	F-score ↑	F-score-T ↑
RT-1	LingBot-VA	300.3	20.04	–	0.160				–			
RT-1	X-WAM reimpl.	311.1	19.99	0.738	0.164	0.837	0.893	0.206	0.0432	0.0069	0.224	0.480
RT-1	WAM4D	289.3	20.37	0.753	0.157	0.881	0.917	0.175	0.0323	0.0029	0.295	0.580
Bridge-Data	LingBot-VA	294.6	18.67	–	0.178				–			
Bridge-Data	X-WAM reimpl.	301.5	18.51	0.735	0.186	0.842	0.923	0.148	0.0439	0.0128	0.250	0.487
Bridge-Data	WAM4D	296.5	18.76	0.742	0.182	0.907	0.956	0.106	0.0306	0.0031	0.334	0.582

randomized evaluation protocols, respectively. This full table complements the aggregate results in the main paper and shows the task-level behavior of each method.

J Qualitative 4D Rollout Visualization

The spatial-register depth readout supervises future geometry during training, shaping register features that action queries read directly. It can also be run for qualitative analysis. In this setting, WAM4D has an interpretable 4D rollout capability: starting from only the first observed frame, the model autoregressively predicts future RGB frames and depth maps, and the generated RGB-D frames can be back-projected into point clouds. Fig. 5 visualizes this process. This analysis path runs the temporal upsampler, adapters, and DA3 depth decoder. Default action deployment skips these depth-readout modules but still runs spatial-register prefill and supplies cached register KV to the action predictor.

K Failure Analysis

Fig. 6 shows a representative failure case in long autoregressive rollout. Because WAM4D does not introduce an explicit long-term memory, objects that become occluded or leave the visible context may be completed as different objects when the model continues rolling out the scene. During control, the model continuously receives fresh observations from the real environment or simulator rather than relying solely on its own generated rollout. This qualitative failure therefore does not by itself establish the same failure mode in closed-loop policy execution; its effect on policy success has not been isolated.

Table 13: Per-task RoboTwin 2.0 success rates. All entries are percentages. Easy and Hard correspond to the clean and randomized settings, respectively. Baseline results are taken from the full LingBot-VA RoboTwin 2.0 report (Li et al., 2026b) and the Fast-WAM paper (Yuan et al., 2026b).

Simulation Task	Horizon	WAM4D		Fast-WAM		LingBot-VA		π_0		$\pi_{0.5}$		Motus	
		Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Adjust Bottle	1	**100**	**99**	100	100	90	94	99	95	100	99	89	93
Beat Block Hammer	1	**99**	97	99	97	96	98	79	84	96	93	95	88
Blocks Ranking RGB	3	**100**	97	100	100	99	98	80	63	92	85	99	97
Blocks Ranking Size	3	**98**	84	94	98	94	96	14	5	49	26	75	63
Click Alarmclock	1	**100**	**100**	100	100	99	100	77	68	98	89	100	100
Click Bell	1	**100**	**100**	100	100	100	100	71	48	99	66	100	100
Dump Bin Bigbin	1	92	94	97	96	89	96	88	83	92	97	95	91
Grab Roller	1	**100**	**100**	100	100	100	100	98	94	100	100	100	100
Handover Block	2	96	**93**	95	81	99	78	47	31	66	57	86	73
Handover Mic	2	94	94	99	100	94	96	97	97	98	97	78	63
Hanging Mug	2	**64**	**57**	58	62	40	28	14	11	18	17	38	39
Lift Pot	1	**100**	88	100	100	100	99	80	72	96	85	96	99
Move Can Pot	1	**96**	**97**	90	88	94	97	68	48	51	55	34	74
Move Pillbottle Pad	1	**100**	**99**	100	99	99	99	67	46	84	61	93	96
Move Playingcard Away	1	**100**	**100**	100	100	100	99	74	65	96	84	100	96
Move Stapler Pad	1	79	76	77	64	91	79	41	24	56	42	83	85
Open Laptop	1	**100**	66	98	100	92	94	71	81	90	96	95	91
Open Microwave	1	82	77	62	45	82	86	4	32	34	77	95	91
Pick Diverse Bottles	2	**98**	**92**	80	85	89	82	69	31	81	71	90	91
Pick Dual Bottles	2	**100**	**99**	100	96	100	99	59	37	93	63	96	90
Place A2B Left	1	**97**	86	95	93	97	93	43	47	87	82	82	79
Place A2B Right	1	94	79	93	99	97	95	39	34	87	84	90	87
Place Bread Basket	1	93	**97**	91	93	97	95	62	46	77	64	91	94
Place Bread Skillet	2	93	87	90	93	95	90	66	49	85	66	86	83
Place Burger Fries	2	96	97	96	99	97	95	81	76	94	87	98	98
Place Can Basket	2	**87**	83	71	69	81	84	55	46	62	62	81	76
Place Cans Plasticbox	2	**100**	**100**	99	96	100	99	63	45	94	84	98	94
Place Container Plate	1	98	91	96	100	99	97	97	92	99	95	98	99
Place Dual Shoes	2	**96**	**97**	94	88	94	89	59	51	75	75	93	87
Place Empty Cup	1	**100**	**100**	100	100	100	100	91	85	100	99	99	98
Place Fan	1	**100**	**95**	96	96	99	93	66	71	87	85	91	87
Place Mouse Pad	1	**96**	**97**	83	89	93	96	20	20	60	39	66	68
Place Object Basket	2	88	**90**	89	88	91	88	67	70	80	76	81	87
Place Object Scale	1	**99**	**97**	90	97	96	95	57	52	86	80	88	85
Place Object Stand	1	98	**100**	90	94	99	96	82	68	91	85	98	97
Place Phone Stand	1	94	81	97	99	97	97	49	53	81	81	87	86
Place Shoe	1	**100**	**99**	96	99	98	98	76	76	92	93	99	97
Press Stapler	1	**97**	94	90	97	85	82	44	37	87	83	93	98
Put Bottles Dustbin	3	87	85	95	90	87	91	65	56	84	79	81	79
Put Object Cabinet	2	81	42	94	89	85	87	73	60	80	79	88	71
Rotate QRcode	1	95	**95**	93	89	96	91	74	70	89	87	89	73
Scan Object	2	92	78	89	92	96	91	55	42	72	65	67	66
Shake Bottle Horizontally	1	**100**	99	100	100	100	99	98	92	99	99	100	98
Shake Bottle	1	**100**	99	100	100	100	97	94	91	99	97	100	97
Stack Blocks Three	3	98	97	95	97	99	98	72	52	91	76	91	95
Stack Blocks Two	2	**100**	**100**	100	100	100	98	93	79	97	100	100	98
Stack Bowls Three	3	83	84	80	81	86	83	77	75	77	71	79	87
Stack Bowls Two	2	96	95	92	98	94	98	94	95	95	96	98	98
Stamp Seal	1	**96**	87	90	94	96	97	46	33	79	55	93	92
Turn Switch	1	60	64	61	59	44	45	41	42	62	54	84	78
Average (50 tasks)	–	94.24	90.08	91.88	91.78	92.90	91.50	65.92	58.40	82.74	76.76	88.52	87.04

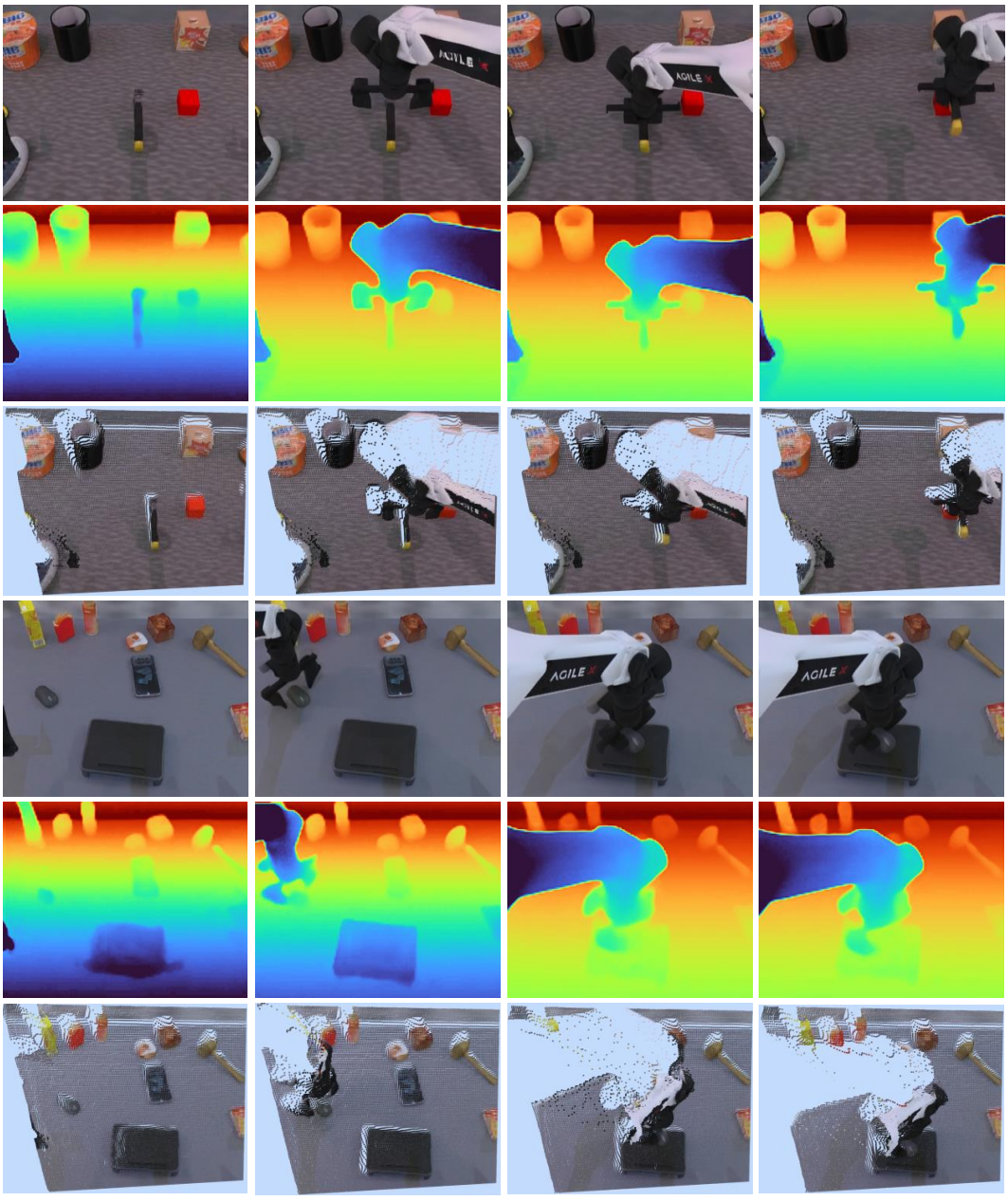


Figure 5: RGB-D and point-cloud rollout visualization. Starting from a single initial frame, WAM4D autoregressively predicts future RGB frames and depth maps; the predicted RGB-D frames are then back-projected into point clouds to visualize the induced 4D scene evolution.



Figure 6: Failure case of long autoregressive rollout. Without an explicit long-term memory, the model may complete an object as a visually plausible but different object after it becomes occluded during rollout.

L LIMITATIONS

WAM4D improves geometry-aware video-action representations without requiring dense geometry during policy deployment. Depth-supervised training incurs additional compute, and datasets without measured depth require offline pseudo-depth generation. DA3-derived targets inherit estimator errors and temporal inconsistency; training with such targets has not been validated on Internet-scale RGB datasets. As discussed in Sec. K, the model also does not maintain an explicit long-term object memory during autoregressive rollout. This can lead to identity-inconsistent completions after severe occlusion in qualitative generated futures. Future work could add persistent object memory or scene-state tracking while preserving the lightweight observation-to-action deployment path.

M COMPUTE DETAILS

Table 14: Compute and latency comparison on a single A800 80GB GPU. Latency is reported as mean $\pm$ std in ms when available. Peak memory follows the VRAM measurement in the main paper.

Method / Mode	Denoising steps	Latency / Chunk (ms)	Peak Mem.
π_0 inference	10 action steps	64.16 ± 0.06	8.45 GiB
$\pi_{0.5}$ inference	10 action steps	72.03 ± 0.06	8.45 GiB
Motus inference	5 video steps & 10 action steps	1516.30 ± 10.64	11.55 GiB
LingBot-VA inference	5 video steps & 10 action steps	843.57 ± 11.55	12.97 GiB
Fast-WAM inference	10 action steps	425.53 ± 6.01	11.55 GiB
WAM4D inference	10 action steps	592.34 ± 7.12	10.09 GiB

Appendix Table 14 reports the inference costs corresponding to the main paper. The WAM4D row measures the checkpoint at training step 10,000. All latency and peak-memory measurements are collected on a single A800 80GB GPU. For WAM4D, we use batch size one, BF16, guidance scale one, and eager execution. We report the mean and population standard deviation over 50 calls after five warmup calls, using CUDA events with synchronization. Each call conditions on nine observed RGB frames, 32 observed states, and the natural-language episode instruction from a fixed clean hanging-mug demonstration, retains 240 spatial registers, and generates 32 actions with 10

denoising steps. Timing includes production history/register/state prefill, scheduler and noise setup, action denoising, and action postprocessing; model loading, reset, RGB/text encoding, and network transport are excluded. Peak allocated memory is reset after warmup, with the VAE offloaded and the text encoder and unused dense depth decoder absent. WAM4D trains spatial registers, the temporal upsampler, four adapters, and the pretrained-initialized DA3 depth decoder. Action inference runs spatial-register prefill and caches register KV for denoising; the depth-readout modules can be skipped.