WORLD4ACT POLICY: LEARNING EXPLICIT 4D WORLD DYNAMICS FOR ROBOT POLICY TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

World–Action Models improve robotic policy learning by coupling action prediction with forecasts of future visual dynamics. However, existing approaches predominantly supervise future evolution in RGB space, leaving the underlying 3D geometry and motion implicit. Although recent 4D world models incorporate geometric or motion cues, dense 3D structure and scene flow have yet to be systematically explored as action-conditioned prediction targets. Here we present World4Act Policy, an action-centred 4D World–Action Model that explicitly learns future visual appearance, 3D geometry and motion alongside robot actions. Given current multi-view RGB observations, robot states and language instructions, the model jointly predicts future action chunks, multi-view RGB frames, image-aligned point maps and 3D scene flow. To transfer pretrained video-generative priors to these 4D modalities, we develop dedicated geometry and motion variational autoencoders, together with a progressive training strategy that proceeds from 4D representation learning to world modelling and, ultimately, joint World–Action learning. For efficient deployment, we incorporate an action-centered causal architecture that separates action generation from future-world prediction, allowing the future-world streams to be omitted during action-only inference. To provide the dense supervision required for learning action-conditioned 4D dynamics at scale, we also curate Robo4D_Dyn Dataset, a large-scale embodied-video dataset annotated with dense point maps and 3D scene flow. Experiments demonstrate that World4Act Policy predicts future 4D dynamics more accurately than representative 4D world models and achieves higher manipulation success rates than both video-based and 4D policy baselines, while preserving low-latency, action-only inference.

1 INTRODUCTION

World models offer a compelling paradigm for robotic manipulation by requiring an agent to anticipate how its actions alter the environment. When coupled with action generation, World–Action Models (WAMs) jointly learn a control policy and a predictive model of action-conditioned future dynamics Bi et al. (2025); Kim et al. (2026); Ye et al. (2026). In contrast to action labels, which supervise only the desired control outputs, future-world prediction provides temporally dense learning signals that characterize object dynamics, physical interactions, and the consequences of robot motion. This predictive objective encourages the model to acquire representations grounded in interaction dynamics and spatial structure, thereby providing a richer foundation for reasoning and closed-loop control Yuan et al. (2026); Zhang et al. (2026); Wang et al. (2026).

Existing WAMs predominantly model future observations in RGB space Yuan et al. (2026); Kim et al. (2026); Ye et al. (2026). Although RGB prediction captures rich appearance and semantic information, it expresses scene evolution through projected pixel changes, leaving the underlying 3D geometry and motion implicit. Consequently, the policy must recover object locations, spatial relationships, and action-induced motion from appearance alone. This limitation is particularly consequential for manipulation: grasping requires accurate 3D localization, placement depends on precise relative geometry Li et al. (2025; 2026a), and more complex tasks demand spatial reasoning and feasible trajectory planning in 3D. Recent embodied 4D world models move beyond RGB-only prediction by incorporating geometric or motion cues Zhen et al. (2025); Xu et al. (2026); Zhu et al. (2025); Li et al. (2026b). However, many of these models are formulated as action-conditioned

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.

simulators: they predict future scene evolution from prescribed robot actions but do not generate the actions themselves Xu et al. (2026); Zhen et al. (2025). Existing approaches therefore either learn control from geometrically implicit RGB supervision or model 4D dynamics without integrating them into an action-generating policy. This dichotomy motivates a unified World–Action Model that directly supervises future 3D structure and motion while jointly learning robot actions.

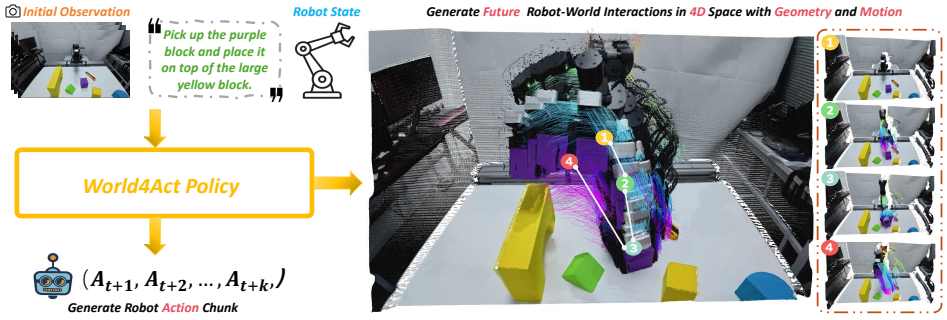

Figure 1: **Illustration of World4Act Policy.** Given visual observations, robot states, and a language instruction, the policy generates an action chunk while anticipating the resulting robot–world interactions in 4D space. Explicitly modeling future 3D geometry and motion provides structured supervision for learning actions grounded in the physical consequences of manipulation.

Developing such a model, however, raises two key challenges. First, dense 3D geometry and motion differ markedly from natural images in both their statistical distributions and semantics, complicating the transfer of pretrained video-generative priors to 4D prediction. Second, jointly generating actions and multiple future-world modalities can impose substantial computational overhead during closed-loop control. RynnWorld-4D Zhao et al. (2026), the work most closely related to ours, connects 4D prediction with policy learning by jointly predicting RGB, depth, and optical flow for action generation. Its 3D motion, however, is recovered indirectly by back-projecting depth and optical flow through camera geometry, rather than learned as a native 3D prediction target. As a result, action-induced motion is not directly constrained in 3D space. Addressing these challenges calls for a unified framework that explicitly represents 3D structure and motion, effectively adapts pretrained video priors to 4D modalities, and exploits dense future-world supervision without retaining world-generation overhead at deployment.

To this end, we introduce **World4Act Policy**, an action-centred 4D World–Action Model for robotic manipulation. As shown in Fig. 1, the central idea is to represent action-conditioned world evolution jointly through RGB, point maps, and 3D scene flow. Point maps encode image-aligned, per-pixel 3D structure, while 3D scene flow directly characterizes its temporal evolution. Because both representations are aligned with the video grid, they provide explicit supervision for geometry and motion while remaining compatible with pretrained video-generative architectures.

Conditioned on current multi-view RGB observations, robot state, and language instructions, World4Act Policy jointly predicts future action chunks, multi-view RGB frames, and head-view point maps and 3D scene flow. To accommodate these heterogeneous prediction targets, we develop dedicated geometry and motion variational autoencoders (VAEs) that construct modality-specific latent spaces, together with a progressive three-stage training strategy that proceeds from 4D representation learning to world-model pretraining and, ultimately, joint World–Action learning. Following the action-centered design of GigaWorld-Policy Ye et al. (2026), we incorporate a causal attention structure into our multimodal model. Action tokens attend only to the current context and other action tokens, while future-world prediction additionally conditions on the action representation. This separation enables low-latency, action-only inference by omitting future-world generation at deployment, while retaining supervision from future 4D dynamics during training.

Learning such action-conditioned 4D dynamics at scale further requires dense and temporally consistent geometric supervision. We therefore curate **Robo4D_Dyn Dataset**, a large-scale collection of embodied videos with temporally aligned point-map and dense 3D-scene-flow annotations across diverse scenes and interactions. By capturing both the 3D structure of the observed environment

and its action-induced evolution over time, Robo4D_Dyn Dataset supports the progressive learning of modality-specific 4D representations, predictive world dynamics, and action-generating policies.

We evaluate World4Act Policy across diverse robotic manipulation tasks. The results show that it predicts future 3D structure and motion more accurately than representative 4D world models and achieves higher manipulation success rates than both video-based WAMs and 4D policy baselines. Importantly, these gains are obtained while retaining low-latency, action-only inference at deployment. Our contributions are threefold:

- We propose **World4Act Policy**, a unified 4D World–Action Model that jointly learns future action chunks and action-conditioned world dynamics represented by multi-view RGB, image-aligned point maps, and 3D scene flow. Dedicated geometry and motion VAEs, progressive three-stage training, and a modality-asymmetric causal design enable effective 4D learning and efficient action-only deployment.
- We curate **Robo4D_Dyn Dataset**, a large-scale embodied-video dataset with temporally aligned point-map and dense 3D-scene-flow annotations, providing direct supervision for learning action-conditioned 3D structure and motion at scale.
- We demonstrate that World4Act Policy improves future 4D prediction over representative world models and consistently outperforms video-based and 4D policy baselines across diverse robotic manipulation tasks, while preserving low-latency closed-loop inference.

2 Related Work

World–Action Models World models capture predictive environment dynamics to facilitate planning and decision-making in embodied control Zhu et al. (2025); Liang et al. (2025); Hu et al. (2023). Building on scalable video-generative priors, recent approaches model visually diverse interactions and action-conditioned scene evolution Zhou et al. (2024); Bar et al. (2024). Early formulations derive control commands from predicted goal images, keyframes, or video trajectories through online planning or inverse dynamics Jang et al. (2025); Guo et al. (2025); Huang et al. (2026); Zhu et al. (2024). World–Action Models (WAMs) instead unify future-world prediction and action generation within a single architecture Yuan et al. (2026); Wang et al. (2026), using future prediction as dense auxiliary supervision while permitting action generation without explicit visual rollouts at deployment Bi et al. (2025); Kim et al. (2026); Ye et al. (2026). Whereas existing WAMs predominantly characterize future dynamics through RGB observations, World4Act Policy extends this formulation by explicitly modeling 3D structure and motion through point maps and 3D scene flow, while retaining efficient action-only inference.

4D World Models RGB-based world models capture visual appearance and semantic information but provide limited direct supervision for underlying spatial geometry and physical 3D motion. Recent 4D generative models address this limitation by predicting dynamic geometric representations alongside conventional visual observations Chen et al. (2025b); Liu et al. (2025). Embodied 4D world models further incorporate depth, surface normals, optical flow, or point maps to describe scene evolution over time under agent interaction in complex environments Zhen et al. (2025); Xu et al. (2026); Li et al. (2026b). Most closely related, RynnWorld-4D Zhao et al. (2026) jointly predicts RGB, depth, and optical flow and recovers 3D motion by geometrically lifting these camera-projective outputs into 3D space. This formulation introduces useful geometric and motion cues while retaining compatibility with pretrained video-generation priors. In contrast, World4Act Policy directly models point maps and 3D scene flow as native action-conditioned prediction targets for direct 4D scene prediction, jointly learning multi-view appearance, head-view 4D dynamics, and robot actions within a unified and efficient World–Action Model.

3 Method

3.1 Problem Formulation and Approach Overview

We formulate robotic manipulation as joint action generation and action-conditioned future-world prediction. At control step t, the robot receives synchronized RGB observations from three camera

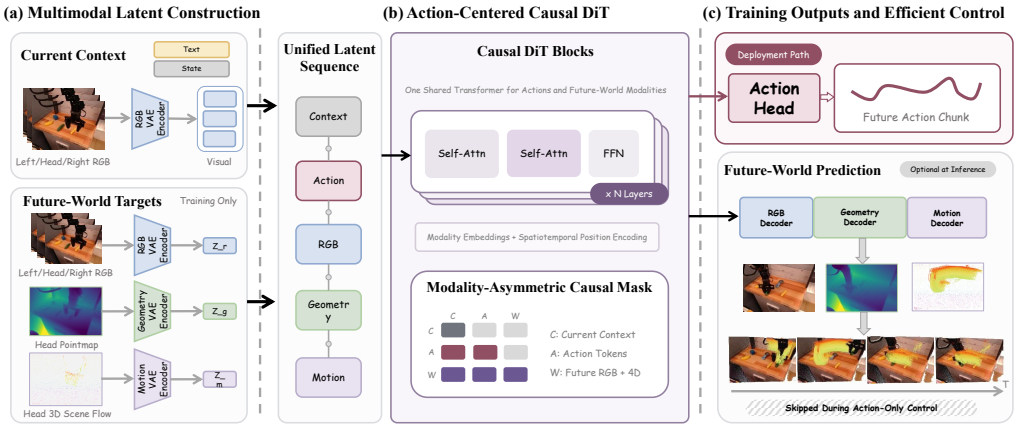

Figure 2: **Overview of World4Act Policy.** Multimodal context and future-world targets are encoded into a unified latent sequence and processed by an action-centered causal DiT. The model jointly learns action generation and future RGB, geometry, and motion prediction, while omitting the future-world branch during action-only inference.

views, $\mathcal{V} = \{\text{left}, \text{head}, \text{right}\}$, together with its proprioceptive state $\mathbf{s}_t$ and a language instruction ℓ. We denote the resulting context by

$$\mathbf{c}_t = (\{\mathbf{I}_t^v\}_{v\in\mathcal{V}}, \mathbf{s}_t, \ell)\,, \tag{1}$$

from which the policy predicts an action chunk $\mathbf{A}_t = \mathbf{a}_{t:t+p-1}$ of horizon p.

Action supervision alone does not explicitly require the model to understand the geometric and dynamic consequences of its decisions. During training, we therefore additionally supervise the model to predict K future observations sampled at temporal stride Δ, with $\tau_k = t + k\Delta$. The future-world target includes multi-view RGB observations, head-view point maps, and head-view 3D scene flow:

$$\mathbf{W}_t = \left(\{\mathbf{I}_{\tau_k}^v\}_{v\in\mathcal{V},k=1}^{K}, \{\mathbf{P}_{\tau_k}\}_{k=1}^{K}, \{\mathbf{F}_{\tau_k\to\tau_{k+1}}\}_{k=1}^{K-1}\right). \tag{2}$$

The $K-1$ scene-flow maps encode forward 3D displacements between consecutive future frames, indexed by their source head-view pixels.

Because future observations are consequences of the robot's actions, whereas actions must be predicted without access to the future, we factorize the joint distribution as

$$p_\theta(\mathbf{A}_t, \mathbf{W}_t \mid \mathbf{c}_t) = p_\theta^a(\mathbf{A}_t \mid \mathbf{c}_t) p_\theta^w(\mathbf{W}_t \mid \mathbf{c}_t, \mathbf{A}_t). \tag{3}$$

Action generation depends only on the current context, while future-world prediction additionally depends on the action representation. World prediction can regularize action learning without leaking future targets into the policy.

As illustrated in Fig. 2, **World4Act Policy** extends the action-centered formulation of GigaWorld-Policy Ye et al. (2026) by explicitly modeling future 3D structure and motion. The model uses modality-specific latent encoders to represent RGB, geometry, and motion, and a shared diffusion Transformer (DiT) to jointly process these representations with robot states and actions. Training proceeds in three stages: adapting the latent encoders to 4D modalities, pretraining the 4D world model, and jointly learning future-world prediction and action generation.

3.2 World4Act Policy Architecture

We represent head-view 4D dynamics using image-aligned point maps and 3D scene flow. A point map $\mathbf{P}_{\tau_k} \in \mathbb{R}^{H\times W\times 3}$ assigns a 3D coordinate to each valid pixel at time τ_k, while $\mathbf{F}_{\tau_k\to\tau_{k+1}} \in \mathbb{R}^{H\times W\times 3}$ describes the forward 3D displacement of that point. Both quantities are expressed in a common sequence-level reference frame $\mathcal{F}_{\text{ref}}$, such that

$$\mathbf{x}_{\tau_{k+1}}^{\text{adv}}(\mathbf{u}) = \mathbf{P}_{\tau_k}(\mathbf{u}) + \mathbf{F}_{\tau_k\to\tau_{k+1}}(\mathbf{u}), \tag{4}$$

where $\mathbf{u}$ denotes a valid source pixel and $\mathbf{x}_{\tau_{k+1}}^{\text{adv}}(\mathbf{u})$ is its predicted 3D position at the next time step. Unlike unstructured point clouds, this representation preserves pixel-level alignment with RGB

observations, making it possible to reuse the spatial and temporal inductive biases of pretrained video models.

Although head-view RGB, point maps, and scene flow share an image grid, their numerical distributions and physical meanings differ substantially. We therefore encode them in separate latent spaces. We initialize the model from Wan 2.2-TI2V-5B Wan et al. (2025), retaining its pretrained VAE as the RGB encoder–decoder (E_r, D_r). The same pretrained weights are used to initialize a geometry VAE (E_g, D_g) and a motion VAE (E_m, D_m), which are subsequently adapted to point maps and scene flow, respectively. Before encoding, we normalize the 4D modalities using fixed training-set statistics:

$$\widetilde{\mathbf{P}} = \frac{\mathbf{P} - \boldsymbol{\mu}_{\mathcal{D}}}{S_{\mathcal{D}}}, \qquad \widetilde{\mathbf{F}} = \frac{\mathbf{F}}{S_{\mathcal{D}}}. \tag{5}$$

Point maps are centered and scaled because they encode absolute 3D positions, whereas scene flow is only scaled because it represents relative displacement. The same statistics are used for training and decoding to maintain a consistent metric interpretation.

After modality-specific encoding, the latent representations are jointly processed by a shared multimodal DiT. Following GigaWorld-Policy Ye et al. (2026), synchronized RGB observations from the three cameras are arranged into a fixed composite image and encoded as observation tokens $\mathbf{T}_o$. Future RGB frames, point maps, and scene flow are encoded by their corresponding VAEs to obtain

$$\mathbf{T}_f = [\mathbf{T}_f^r; \mathbf{T}_f^g; \mathbf{T}_f^m]. \tag{6}$$

The proprioceptive state and action chunk are separately projected into state tokens $\mathbf{T}_s$ and action tokens $\mathbf{T}_a$, while language features $\mathbf{T}_\ell$ condition the DiT blocks through cross-attention.

The shared Transformer jointly processes action tokens, multi-view RGB tokens, and head-view geometry and motion tokens. We use spatiotemporal positional encodings inherited from the video backbone, together with learned modality embeddings to distinguish the prediction streams. The token layouts and positional indexing are detailed in the supplementary material.

The shared architecture must nevertheless preserve the causal factorization in Eq. (3). Let $\mathbf{T}_c = [\mathbf{T}_o; \mathbf{T}_s]$ denote the current-context tokens. We impose a blockwise causal attention mask whose permitted key tokens are

$$\begin{aligned} \mathcal{K}(\mathbf{T}_c) &= \mathbf{T}_c, \\ \mathcal{K}(\mathbf{T}_a) &= [\mathbf{T}_c; \mathbf{T}_a], \\ \mathcal{K}(\mathbf{T}_f) &= [\mathbf{T}_c; \mathbf{T}_a; \mathbf{T}_f]. \end{aligned} \tag{7}$$

Language conditioning is introduced separately through cross-attention. Under this mask, action tokens can use the current observations and robot state but cannot access future-world tokens. In contrast, future-world tokens can attend to both the current context and the evolving action representation. This causal separation also makes world modeling optional during deployment. Because the action tokens do not depend on future-world tokens, closed-loop inference instantiates only the context and action streams. Future tokens and their modality-specific decoders can be omitted, reducing computational cost while leaving the learned action-generation pathway unchanged.

3.3 WORLD4ACT POLICY TRAINING

Joint learning of appearance, geometry, motion, and actions requires temporally aligned supervision across these modalities. We therefore construct **Robo4D_Dyn**, a large-scale embodied-video dataset containing synchronized multi-view RGB observations, point maps, and 3D scene flow, as shown in Fig. 3. For action-supervised trajectories, Robo4D_Dyn additionally provides language instructions, proprioceptive states, and robot actions. Geometry and motion annotations from all valid camera views are used to adapt the corresponding VAEs. During world-model pretraining and joint world–action learning, RGB prediction is supervised across all three views, while point-map and scene-flow prediction is supervised in the head view.

Stage I: Modality Adaptation. Training begins by adapting the pretrained video VAE to the two additional 4D modalities. The geometry and motion VAEs are trained separately on normalized point-map and scene-flow sequences using validity-masked reconstruction objectives and the regularization terms inherited from the pretrained RGB VAE. Masking prevents missing or unreliable 4D annotations from contributing to the reconstruction loss. Once adapted, the RGB, geometry, and

Table 1: Quantitative comparison with other world modeling methods. We evaluate visual appearance using IQ, MS, SC, PSNR, SSIM, and LPIPS, 3D geometry using relative point error Rel_p and inlier ratio δ_p, and 3D motion using EPE_{3D} and APD. A dash indicates that the method does not produce the corresponding modality. The World4Act results are obtained after Stage-II pretraining.

Method	RGB Prediction						Point Map		3D Scene Flow	
	IQ ↑	MS ↑	SC ↑	SSIM ↑	PSNR ↑	LPIPS ↓	Rel_p ↓	δ_p ↑	EPE_{3D} ↓	APD ↑
Video Generation Models										
CogVideoX (ICLR 2025) Yang et al. (2025)	0.604	0.976	0.866	0.534	12.17	0.577	–	–	–	–
Wan-2.2-TI2V-5B (ArXiv 2025) Wan et al. (2025)	0.555	0.970	0.886	0.593	14.54	0.489	–	–	–	–
Wan-2.1-I2V-14B (ArXiv 2025) Wan et al. (2025)	**0.684**	0.988	0.891	0.536	12.72	0.568	–	–	–	–
4D World Models										
Free4D (ICCV 2025) Liu et al. (2025)	0.354	0.993	0.787	0.492	12.40	0.597	0.813	0.165	–	–
TesserAct (ICCV 2025) Zhen et al. (2025)	0.608	0.992	0.904	0.693	16.91	0.335	0.462	0.372	–	–
4DNeX (ArXiv 2025) Chen et al. (2025b)	0.637	0.994	0.917	0.649	14.47	0.404	0.438	0.401	–	–
Kinema4D (ArXiv 2026) Xu et al. (2026)	0.634	0.994	0.918	0.650	15.13	0.400	0.421	0.423	–	–
RynnWorld-4D (ArXiv 2026) Zhao et al. (2026)	0.635	0.995	**0.957**	0.754	17.85	0.269	0.324	0.510	0.143	0.642
World4Act (Ours)	0.637	**0.997**	0.955	**0.763**	**17.91**	**0.252**	**0.268**	**0.594**	**0.135**	**0.663**

motion VAEs are frozen, ensuring that subsequent stages operate in stable modality-specific latent spaces.

Stage II: 4D World-Model Pretraining. We next pretrain the shared DiT as a 4D world model before introducing action supervision. Given the current multi-view observations, the model learns to generate future RGB, geometry, and motion latents through conditional flow matching. For each modality $c \in \mathcal{C} = \{r, g, m\}$ with clean latent $\mathbf{x}^c$, we independently sample $\boldsymbol{\epsilon}^c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and use a shared flow time $\sigma \sim \mathcal{U}(0, 1)$. The interpolated latent, target velocity, and modality-specific objective are

$$\begin{aligned} \mathbf{x}_\sigma^c &= (1-\sigma)\boldsymbol{\epsilon}^c + \sigma \mathbf{x}^c, \\ \mathbf{u}^c &= \mathbf{x}^c - \boldsymbol{\epsilon}^c, \\ \mathcal{L}_c &= \mathbb{E}\left[\left\| \widehat{\mathbf{u}}_\theta^c - \mathbf{u}^c \right\|_2^2 \right]. \end{aligned} \tag{8}$$

The modalities share the same flow time within each training example to synchronize their denoising trajectories, while using independent noise samples to preserve modality-specific variation. The resulting world-model objective is

$$\mathcal{L}_{\mathrm{WM}} = \lambda_r \mathcal{L}_r + \lambda_g \mathcal{L}_g + \lambda_m \mathcal{L}_m, \tag{9}$$

where λ_r, λ_g, and λ_m balance appearance, geometry, and motion prediction.

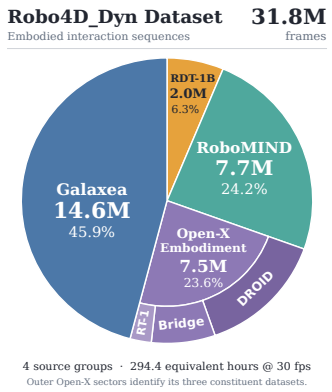

Figure 3: **Composition of the Robo4D_Dyn Dataset.** The dataset contains 31.8M frames from four source groups.

Stage III: Joint World–Action Learning. After the model has learned general 4D scene dynamics, we introduce language, robot-state, and action supervision for joint world–action learning. Action chunks are treated as an additional generative stream and follow the same flow-matching formulation in Eq. (8), with $c = a$ and $\mathbf{x}^a = \mathbf{A}_t$. The action, RGB, geometry, and motion streams share the same flow time but use independently sampled noise. The complete training objective is

$$\mathcal{L}_{\mathrm{WAM}} = \lambda_a \mathcal{L}_a + \lambda_r \mathcal{L}_r + \lambda_g \mathcal{L}_g + \lambda_m \mathcal{L}_m. \tag{10}$$

Through the causal attention mask, future-world prediction is conditioned on the evolving action representation, encouraging the action stream to encode information predictive of future 4D outcomes. Meanwhile, the action stream remains independent of future targets and can therefore serve as a valid closed-loop policy. During inference, only the action flow is sampled, and the generated action chunk is executed in a receding-horizon manner.

4 Experiments

4.1 Experimental Setup

Implementation Details. We build World4Act Policy on Wan 2.2-TI2V-5B Wan et al. (2025), adopting the shared causal DiT architecture of GigaWorld-Policy Ye et al. (2026). Given synchronized left-, head-, and right-view RGB observations, the model predicts future RGB for all three

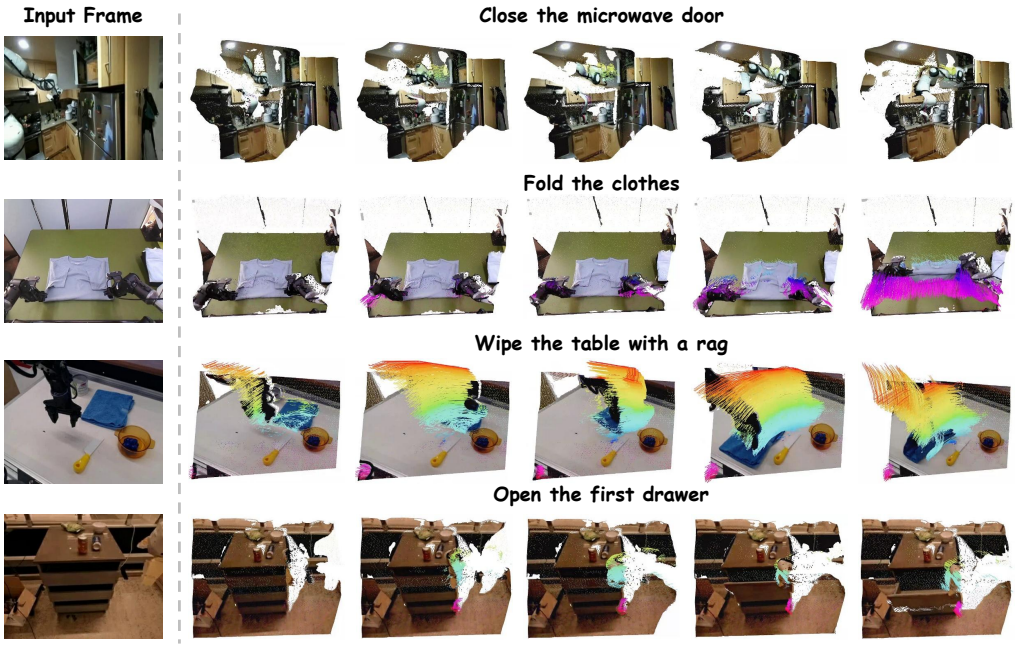

Figure 4: **Qualitative results of future-world 4d dynamic prediction.**

views, together with head-view point maps and 3D scene flow. Training follows the three stages described in Sec. 3, with all VAEs frozen after modality adaptation. During deployment, future-world tokens and VAE decoding are omitted for efficient action-only inference. Full architectural, optimization, and sampling details are provided in the supplementary material.

Evaluation on world modelling. We evaluate the Stage-II world model on a held-out split of Robo4D_Dyn that is disjoint from the training set. We compare against CogVideoX Yang et al. (2025) and Wan 2.2 Wan et al. (2025) for RGB prediction; TesserAct Zhen et al. (2025) and Kinema4D Xu et al. (2026) for geometry-aware world modeling; and RynnWorld-4D Zhao et al. (2026) for joint RGB, depth, and optical-flow prediction. Table 1 reports the future-world prediction performance of World4Act after Stage-II pretraining, before world–action learning. Each baseline retains its native conditioning interface, with unsupported modalities marked as not applicable. For projective baselines, depth and optical flow are converted to point maps and 3D scene flow using the camera parameters. RGB prediction is evaluated using IQ, MS, SC, PSNR, SSIM, and LPIPS, and 3D prediction using Rel_p, δ_p, EPE_{3D}, and APD. For each predicted point-map sequence, we apply a single scale and translation, with the same scale applied to its scene flow, and compute 3D metrics over valid annotated points. These scores measure agreement with the reference annotations after alignment, rather than absolute scale recovery. Baseline conditioning, geometric lifting, alignment, metric definitions, and thresholds are detailed in the supplementary material.

Evaluation on policy training. We evaluate policy performance on RoboTwin 2.0 Chen et al. (2025a) under both clean and randomized conditions, as well as on four real-world tasks: *Clean the Desk*, *Scan a QR Code*, *Sweep up Trash*, and *Stack Bowls*. We report per-task and average success rates and compare against $\pi_0.5$ Intelligence et al. (2025), GigaBrain-0 Team et al. (2025), Motus Bi et al. (2025), Cosmos-Policy Kim et al. (2026), GigaWorld-Policy Ye et al. (2026), and RynnWorld-4D-Policy Zhao et al. (2026). Where supported by released implementations, reproduced baselines use matched demonstrations, camera configurations, action horizons, initial-state randomization, and evaluation protocols. Task definitions are provided in the supplementary material.

4.2 Main Results

Performance of world modelling. Table 1 demonstrates that World4Act delivers consistently strong future-prediction performance across appearance, geometry, and motion, while preserving competitive RGB synthesis quality. Relative to RynnWorld-4D, which reconstructs 3D predictions

Table 2: Manipulation policy evaluation results using RoboTwin 2.0 and real-world scenarios. We report Success Rate under clean and randomized simulation settings and across four real-world tasks.

Method	RoboTwin 2.0		Real-World Manipulation				
	Clean	Randomized	Clean Desk	Scan QR	Sweep Trash	Stack Bowls	Average
Vision–Language–Action Models							
$\pi_0.5$ (CoRL 2025) Intelligence et al. (2025)	0.43	0.44	0.75	0.55	0.65	0.80	0.69
GigaBrain-0 (ArXiv 2025) Team et al. (2025)	0.42	0.42	0.70	0.65	0.60	0.75	0.68
World–Action Models							
Motus (CVPR 2026) Bi et al. (2025)	0.89	0.87	0.80	0.75	0.70	0.80	0.76
Cosmos-Policy (ICLR 2026) Kim et al. (2026)	0.51	0.52	0.65	0.50	0.45	0.70	0.58
GigaWorld-Policy (ArXiv 2026) Ye et al. (2026)	0.87	0.85	0.90	0.75	0.75	**0.90**	0.83
RynnWorld-4D-Policy (ArXiv 2026) Zhao et al. (2026)	0.82	0.84	0.88	0.75	0.75	0.89	0.82
World4Act Policy (Ours)	**0.90**	**0.90**	**0.92**	**0.78**	**0.79**	0.89	**0.85**

by lifting estimated depth and optical flow, World4Act reduces Rel_p from 0.324 to 0.268 and EPE_{3D} from 0.143 to 0.135, corresponding to relative improvements of 17.3% and 5.6%, respectively. It further improves δ_p and APD by 8.4 and 2.1 percentage points. These results substantiate the benefit of directly learning future 3D coordinates and displacements within a pretrained video generative model. Fig. 4 further visualizes the predicted 4D scene evolution across four manipulation tasks. The predictions preserve clear object boundaries and recognizable geometric structure, while capturing coherent motion patterns during manipulation. The predicted scene flow follows the movement of the robot and manipulated objects, with relatively limited displacement in static background regions. Together, the evolving point clouds illustrate the model's ability to represent future geometry and motion coherently throughout robot–world interactions.

Performance of robot manipulation policy. Table 2 reports Success Rates on RoboTwin 2.0 and four real-world manipulation tasks. World4Act Policy achieves the highest success rates in both clean and randomized simulations, along with the highest average success rate across real-world tasks. Compared with the RGB-based GigaWorld-Policy, it improves the Success Rate by 3.0 percentage points in clean simulation and 5.0 points under randomized conditions. On real-world tasks, it increases the average Success Rate from 83.0% to 85.0%. The largest gain is observed on *Sweep up Trash* (+4.0 points), which requires anticipating contact-induced object motion, while performance on *Scan a QR Code* improves by 3.0 points, where success depends on precise relative positioning and coordinated alignment. These results support our hypothesis that direct supervision of future 3D structure and motion provides action-relevant spatial cues beyond those learned from RGB prediction alone.

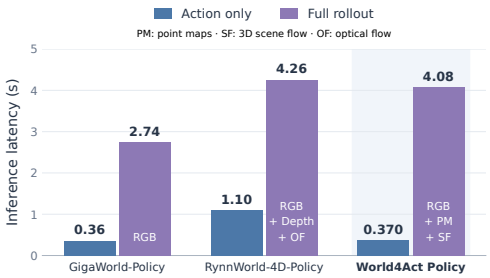

Figure 5: **Inference efficiency.** Latency of action-only inference and full rollout (lower is better). RynnWorld-4D-Policy retains its predictive backbone, whereas ours omits all future-world computation. All methods use identical evaluation settings. PM, SF, and OF denote point maps, 3D scene flow, and optical flow.

Efficiency analysis. In our work, the modality-asymmetric causal mask decouples action generation from future-world prediction: action tokens attend only to current observations, robot state, language instructions, and action tokens. Thus, action-only inference omits both future-world Transformer computation and VAE decoding, while full 4D rollout remains available when explicit prediction is required. As shown in Fig. 5, RynnWorld-4D-Policy still executes its predictive 4D backbone to extract policy features even without decoding future outputs. By contrast, our causal factorization reduces action-only latency to 0.37 s, nearly matching the 0.36 s of the RGB-based GigaWorld-Policy while being substantially more efficient than RynnWorld-4D-Policy.

4.3 Efficiency and Ablation Studies

Effect of 4D supervision. Table 3(a) evaluates the contribution of each future-world prediction target. Relative to RGB-only supervision, point maps and 3D scene flow improve randomized

Table 3: Ablation studies on future-world supervision and progressive training. RGB only denotes a controlled ablation of World4Act. Prediction metrics use Stage-II checkpoints, except for w/o Stage-II Pretraining, which uses its policy-trained checkpoint. All variants follow the same evaluation protocol; projective outputs are converted to point maps and 3D scene flow before computing 3D metrics. Best and second-best results are bold and underlined, respectively. Policy metrics use the corresponding policy-trained checkpoints.

(a) Effect of future-world supervision

Variant	Geometry Target	Motion Target	$\mathrm{Rel}_p \downarrow$	$\mathrm{EPE}_{3D} \downarrow$	RoboTwin Rand. ↑	Real-World Avg. ↑
RGB only	–	–	–	–	0.81	0.76
RGB + Point Map	Point Map	–	0.294	–	0.83	0.78
RGB + 3D Scene Flow	–	3D Flow	–	0.186	0.84	0.80
Projective 4D	Depth	2D Flow	<u>0.280</u>	<u>0.151</u>	<u>0.87</u>	<u>0.82</u>
Direct 4D (Ours)	Point Map	3D Flow	**0.268**	**0.135**	**0.90**	**0.85**

(b) Effect of progressive training

Variant	Specialized VAEs	Stage-II 4D WM	Robo4D_Dyn	$\mathrm{Rel}_p \downarrow$	$\mathrm{EPE}_{3D} \downarrow$	RoboTwin Rand. ↑	Real-World Avg. ↑
Shared RGB VAE	–	✓	✓	0.317	0.174	0.80	0.78
w/o Stage-II Pretraining	✓	–	✓	0.293	0.158	0.83	0.80
Target-Robot Data Only	✓	✓	–	<u>0.282</u>	<u>0.150</u>	<u>0.85</u>	<u>0.82</u>
Full Three-Stage Training	✓	✓	✓	**0.268**	**0.135**	**0.90**	**0.85**

RoboTwin success by 2.0 and 3.0 percentage points and average real-world success by 2.0 and 4.0 percentage points, respectively. Combining both signals yields the strongest performance, reaching 90.0% on randomized RoboTwin and an average of 85.0% across real-world tasks. Compared with predicting depth and 2D optical flow, direct 4D supervision reduces Rel_p and EPE_{3D} by 4.3% and 10.6%, respectively, while improving both policy success rates by 3.0 percentage points. These results support the complementary value of future 3D structure and motion and the benefits of modeling them directly rather than recovering them through geometric lifting.

Effect of progressive training. Table 3(b) evaluates each component of the three-stage training pipeline and its contribution to policy learning. Reusing the RGB VAE for all modalities degrades both geometry and motion prediction, supporting the use of modality-specific latent adaptation. Removing Stage-II 4D world-model pretraining decreases randomized RoboTwin and average real-world success by 7.0 and 5.0 percentage points, respectively, highlighting the benefit of learning predictive 4D dynamics before action supervision. Replacing large-scale Robo4D_Dyn dataset pretraining with target-robot data alone reduces the two success rates by 5.0 and 3.0 percentage points, respectively. The complete pipeline achieves the highest policy success rates among the evaluated variants, supporting the use of specialized VAEs, 4D world-model pretraining, and large-scale Robo4D_Dyn data supervision for policy learning.

5 Conclusion

We have introduced **World4Act Policy**, an action-centered World–Action Model that jointly predicts future appearance, image-aligned 3D geometry, and scene flow for embodied manipulation. Through modality-specific VAEs, progressive three-stage training, and large-scale supervision from the **Robo4D_Dyn Dataset**, our framework models 3D structure and motion as native prediction targets rather than recovering them from projective outputs. Its modality-asymmetric causal design provides dense 4D supervision during training while eliminating future-world computation during action-only inference. Experiments demonstrate consistent improvements in 4D prediction and manipulation performance, with control latency comparable to the RGB-based GigaWorld-Policy.

Limitations. The current model predicts point maps and scene flow only from the head view and does not explicitly enforce multi-view geometric consistency or physical constraints. Moreover, complete 4D rollouts remain computationally demanding for long-horizon online planning. Future work will investigate consistent multi-view prediction and more efficient, physically grounded rollout-based control for complex manipulation tasks.

AI USE STATEMENT

We used generative AI assistants in the preparation of this manuscript, including for language editing and the refinement of technical descriptions. The authors tested the code, conducted spot checks of the annotations, and cross-checked the reported results against experimental records. We take full responsibility for the final content of this work, including all AI-assisted writing and verification.

ETHICS STATEMENT

This work studies robot manipulation using existing embodied-video datasets and simulation and real-world evaluations. Relevant ethical considerations include privacy and redistribution restrictions in source data, errors and biases in automatically generated annotations, and physical risks from incorrect robot actions. Geometric and temporal validation and manual spot checks help reduce annotation errors, but do not guarantee error-free annotations or safe robot behavior. Deployment beyond the evaluated settings requires task-specific safety assessment and appropriate safeguards. Use and redistribution of source-derived data must respect the original datasets' licenses and applicable privacy requirements.

REPRODUCIBILITY STATEMENT

The main paper and supplementary material provide methodological and experimental details to support independent reproduction of our results. Section 3 describes the model architecture, 4D representations, normalization, and causal attention mechanism, while Section 3.3 specifies the three-stage training procedure and optimization objectives. The experimental sections and supplementary material describe the data processing, experimental settings, and evaluation protocols. The source code cannot currently be shared because its release is subject to company approval. We will release the code once this approval is obtained.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. *arXiv preprint arXiv:2412.03572*, 2024.

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

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*, 2025a.

Zhaoxi Chen, Tianqi Liu, Long Zhuo, Jiawei Ren, Zeng Tao, He Zhu, Fangzhou Hong, Liang Pan, and Ziwei Liu. 4dnex: Feed-forward 4d generative modeling made easy. *arXiv preprint arXiv:2508.13154*, 2025b.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.

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.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025.

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

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.

Haoyuan Li, Qihang Cao, Tao Tang, Kun Xiang, Zihan Guo, Jianhua Han, JiaWang Bian, Hang Xu, and Xiaodan Liang. Thinking with geometry: Active geometry integration for spatial reasoning. *arXiv preprint arXiv:2602.06037*, 2026a.

Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Jian Tang, et al. Wam4d: Fast 4d world action model via spatial register tokens. *arXiv preprint arXiv:2606.14048*, 2026b.

Dingkang Liang, Dingyuan Zhang, Xin Zhou, Sifan Tu, Tianrui Feng, Xiaofan Li, Yumeng Zhang, Mingyang Du, Xiao Tan, and Xiang Bai. Seeing the future, perceiving the future: A unified driving world model for future generation and perception. *arXiv preprint arXiv:2503.13587*, 2025.

Tianqi Liu, Zihao Huang, Zhaoxi Chen, Guangcong Wang, Shoukang Hu, Liao Shen, Huiqiang Sun, Zhiguo Cao, Wei Li, and Ziwei Liu. Free4d: Tuning-free 4d scene generation with spatial-temporal consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25571–25582, 2025.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094*, 2026.

GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv preprint arXiv:2510.19430*, 2025.

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.

Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, et al. Latent-wam: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026.

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*, 2026.

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*, 2026.

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*, 2026.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, et al. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026.

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.

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.

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.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. *arXiv preprint arXiv:2406.14540*, 2024.

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8535–8546, 2025.

A Appendix

A.1 Implementation Details

Data and token layout. Reconstruction uses 4RC Luo et al. (2026); clips must pass forward–backward consistency after Sim(3) alignment and a projected-track check against CoTracker3 Karaev et al. (2025). These checks assess temporal consistency and agreement with independently estimated image trajectories; they do not establish error-free physical ground truth. Qwen3-VL Bai et al. (2025) provides video descriptions. Complete trajectories are assigned to disjoint training and evaluation splits before clip extraction.

Current observations and future RGB targets contain the left, head, and right views in a fixed horizontal composite. Future geometry and motion use the head view. Following the main paper, K future RGB frames and point maps correspond to $K-1$ forward scene-flow maps. Each flow map is indexed by its source head-view pixels and describes displacement to the next future timestamp, rather than to the same pixel in the next image.

The stream order is $[T_o; T_s; T_a; T_f^r; T_f^g; T_f^m]$, with separate language cross-attention. Flatten video latents in time–row–column order and project each stream to the inherited DiT width. For a head-view modality with spatial token grid $n_y^c \times n_x^c$ and RGB grid $n_y^r \times n_x^r$, map token centers into the head tile:

$$\pi_x^c(j) = \frac{n_x^r}{3}\left(1 + \frac{j+1/2}{n_x^c}\right) - \frac{1}{2}, \qquad \pi_y^c(i) = n_y^r \frac{i+1/2}{n_y^c} - \frac{1}{2}. \tag{11}$$

Temporal indices follow RGB timestamps and the inherited temporal compression; flow uses source times. Learned modality embeddings distinguish streams. State and action tokens use separate projections, with actions indexed by chunk order.

Architecture and optimization. The Transformer and RGB VAE inherit Wan 2.2-TI2V-5B Wan et al. (2025). Geometry/motion VAEs copy the RGB VAE initialization and are adapted separately on all valid annotated views. For fixed normalization, use the mean valid training point for μ_D and

Table 4: Proposed training settings; batch sizes are global.

Stage	Training signal	Learning rate	Updates	Batch
I	Separate geometry/motion VAE adaptation	10^{-5}	50k each	16
II	RGB, point maps, and scene flow	10^{-4}	100k	32
III	Joint world prediction and actions	2×10^{-5}	50k	32

the RMS of centered coordinates for S_D; decoding inverts the transform. All VAEs are frozen after Stage I. Stage II conditions on current RGB; Stage III adds language, state, and actions.

Use AdamW Loshchilov (2017) ($\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01), bfloat16, gradient clipping at 1, and default seed 42; warm up for 1,000 updates, then cosine-decay to 0.1 times the initial rate. Stage I uses masked L1 reconstruction with inherited VAE regularization. Stages II/III use the main paper's flow-matching objectives, sharing $\sigma \sim U(0, 1)$ but sampling independent modality noise. Set loss weights to 1 and average each loss over valid scalar entries, reducing fixed validity masks to latent resolution.

Representations and token organization. The Transformer and RGB VAE inherit Wan 2.2-TI2V-5B Wan et al. (2025). Geometry and motion VAEs are initialized from the RGB VAE and adapted separately. Fixed training-set statistics normalize the modalities as $\widetilde{P} = (P - \mu_D)/S_D$ and $\widetilde{F} = F/S_D$. Decoding applies the inverse transforms. Translation affects point coordinates but not displacement vectors; training normalization is distinct from the sequence alignment used for evaluation.

The Stage-III token groups are $[T_o; T_s; T_a; T_f^r; T_f^g; T_f^m]$, with language supplied through cross-attention. Separate projections accommodate the modalities, and learned modality embeddings identify their streams. Spatiotemporal positional encodings follow the video backbone, with geometry indexed by head-view pixels and flow by source-frame positions. For equal-width left/head/right tiles, a normalized head-view horizontal coordinate q corresponds to $(1 + q)/3$ in the RGB composite. This spatial correspondence does not require the separate modality streams to contain equal numbers of tokens. Action tokens are indexed by their order within the predicted chunk.

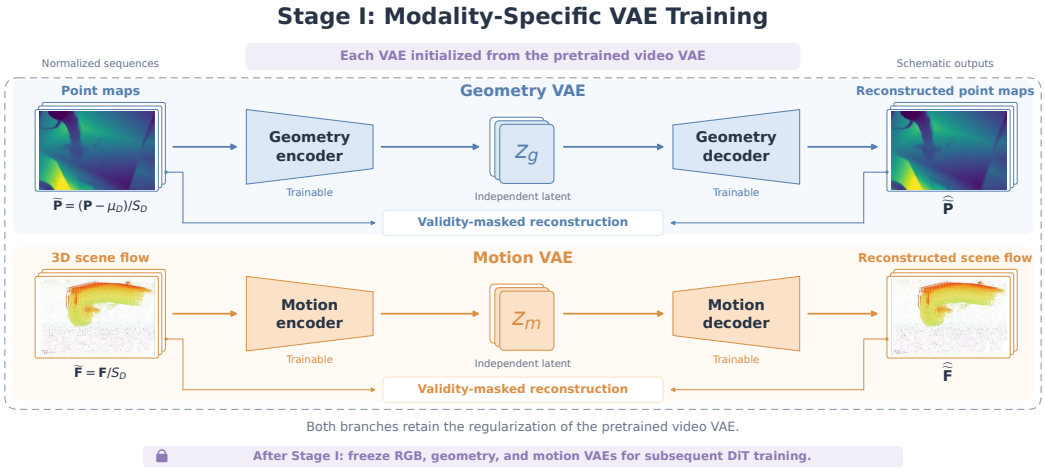

Figure 6: **Stage I: Modality-specific VAE adaptation.** The geometry and motion VAEs are initialized from the pretrained video VAE and independently trained to reconstruct normalized point maps and 3D scene flow. Both encoders and decoders are optimized using validity-masked reconstruction objectives while retaining the pretrained VAE regularization. After Stage I, all three VAEs are frozen for subsequent DiT training. The displayed reconstruction outputs are schematic.

Stage I: Modality-specific VAE adaptation. Figure 6 illustrates the adaptation of the pretrained video VAE to geometry and motion representations. We initialize separate geometry and motion

VAEs from the same pretrained weights and train them independently to reconstruct point-map and 3D scene-flow sequences, respectively. Point maps are centered and scaled using fixed training-set statistics, whereas scene flow is only scaled to preserve its interpretation as displacement. For each branch, both the encoder and decoder are optimized using a validity-masked reconstruction objective, excluding missing or unreliable annotations from the reconstruction loss while retaining the regularization of the pretrained VAE. This stage establishes dedicated latent spaces for the two modalities before world-model training. After adaptation, the RGB, geometry, and motion VAEs are frozen throughout the subsequent DiT pretraining and joint world–action learning stages.

A.2 Evaluation Details

Baseline conditioning. Table 5 summarizes public native interfaces. Actual supplied inputs, checkpoint revisions, and adaptation settings require evaluation records. Use held-out Robo4D_Dyn clips with synchronized three-view RGB and verified reference scale; match clips/timestamps/views and generate one sample per clip where supported. Distinguish future action/robot-trajectory conditioning from observation-only conditioning.

Table 5: Native interfaces documented by the original authors; these do not establish which inputs were supplied in our recorded runs.

Model	Publicly documented conditioning
CogVideoX Yang et al. (2025); Wan 2.2 Wan et al. (2025);	Initial RGB and text in their image-to-video variants.
Free4D Liu et al. (2025)	Image/text via generated video; predefined camera trajectory.
TesserAct Zhen et al. (2025)	Initial RGB/depth/normals and textual action description.
4DNeX Chen et al. (2025b)	Initial RGB, text, and constructed XYZ initialization.
Kinema4D Xu et al. (2026)	Initial RGB and kinematics-derived robot point-map trajectory/mask.
RynnWorld-4D Zhao et al. (2026)	Initial RGB-D and language instruction.

Geometric lifting and reference coordinates. Projective outputs require conversion to a common 3D reference frame before comparison. For camera-axis depth D_k, intrinsics $\mathbf{K}_k$, and camera-to-reference transform (R_k, t_k), the geometric conversion is

$$P_k(u) = R_k\big[D_k(u)\mathbf{K}_k^{-1}\bar{u}\big] + t_k, \qquad F_{k\to k+1}(u) = P_{k+1}\big(u + f_k(u)\big) - P_k(u), \tag{12}$$

where $\bar{u}$ is the homogeneous source pixel and f_k is forward optical flow. The destination point map is evaluated at the flow correspondence. Both positions must use the same reference frame; otherwise camera motion is mixed with scene displacement. The conversion is defined only where the required depth and correspondence information is valid. For directly predicted geometry and motion, $P_k(u) + F_{k\to k+1}(u)$ is the advected position of the source point.

Alignment and metric interpretation. For each predicted point-map sequence, evaluation applies one scale s and translation b, shared across its timestamps. The same scale is applied to scene flow:

$$\widehat{P}'_k = s\widehat{P}_k + b, \qquad \widehat{F}'_{k\to k+1} = s\widehat{F}_{k\to k+1}. \tag{13}$$

Translation cancels for displacements. Scores use valid reference annotations. Rel_p and δ_p assess point-map prediction, while EPE_{3D} and APD assess motion. The endpoint error measures the Euclidean discrepancy between predicted and reference displacements in their common coordinate system. These scores quantify agreement with automatically generated reference annotations after alignment, rather than absolute physical-scale recovery or physical consistency. Threshold-dependent metrics are meaningful only under a common metric implementation and threshold convention.

Policy evaluation and aggregation. Simulation evaluation uses RoboTwin 2.0 Chen et al. (2025a), with clean and randomized conditions reported separately. Real-world evaluation covers Clean the Desk, Scan a QR Code, Sweep up Trash, and Stack Bowls. Success refers to completion of the manipulation task rather than visual plausibility of the predicted rollout. Evaluation uses two

training seeds. For each seed, each simulated task is evaluated for 50 episodes per condition, and each real-world task for 100 trials. Thus, each real-world task has 200 trials across the two seeds.

For task j and seed r, let $S_{j,r}$ and $N_{j,r}$ be the numbers of successful and attempted trials. The reported task and overall success rates are

$$\mathrm{SR}_j = \frac{1}{2}\sum_{r=1}^{2}\frac{S_{j,r}}{N_{j,r}}, \qquad \mathrm{SR}_{\mathrm{avg}} = \frac{1}{J}\sum_{j=1}^{J}\mathrm{SR}_j, \tag{14}$$

where J is the number of tasks in the evaluation setting. Real-world averages use $J = 4$. Clean and randomized simulation results are not pooled. These averages describe empirical performance in the evaluated settings; mean differences alone do not establish statistical significance.

Inference and control. Action-only inference retains observation encoding and action generation but omits future-world tokens and their decoders. Full rollout additionally generates future-world modalities. The policy executes part of each predicted action chunk before replanning from updated observations. Inference-efficiency comparisons use the same hardware, precision, batch size, action horizon, and ODE sampling configuration, as specified in the main paper.

Checkpoint provenance. Table 1 reports world-prediction results after Stage II. In Table 3, prediction metrics use the corresponding Stage-II checkpoint when available; the variant without Stage-II pretraining uses its policy-trained checkpoint. Policy scores use the corresponding policy-trained models. GigaWorld-Policy in Table 2 is an external method baseline, whereas RGB-only in Table 3 is a controlled variant of our framework. They represent distinct experimental comparisons.

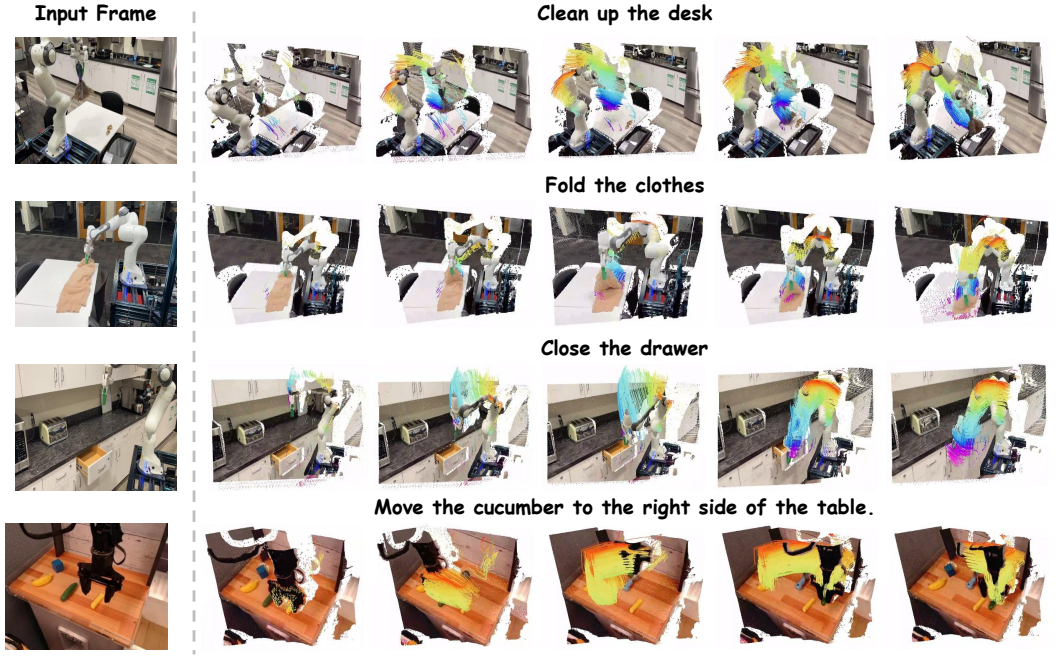

Figure 7: **Additional future-world predictions.** From top to bottom: cleaning up a desk, folding clothes, closing a drawer, and moving a cucumber to the right side of a table. The leftmost column shows the input RGB image, while the remaining columns show successive 3D visualizations with colored motion trajectories

A.3 Additional Visualizations and Failure Cases

Additional visualizations. Figure 7 presents additional qualitative examples of robot manipulation, including cleaning up a desk, folding clothes, closing a drawer, and moving a cucumber to

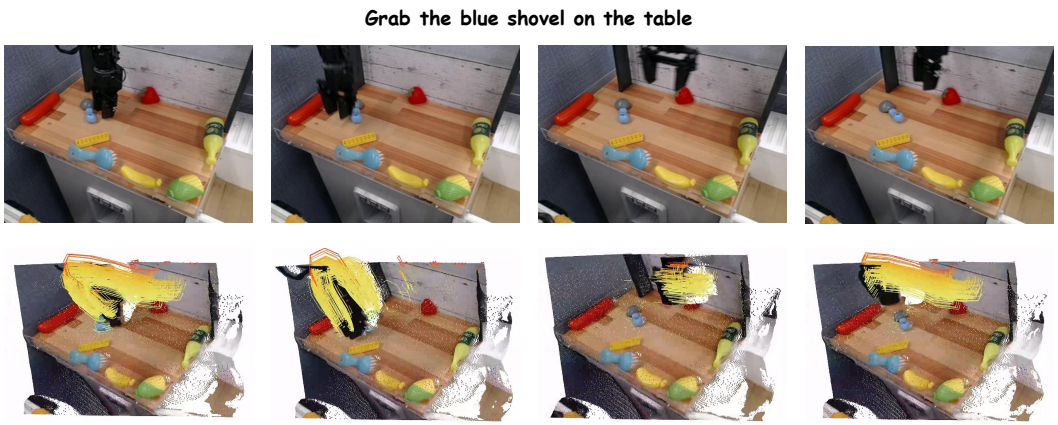

Figure 8: **Failure case.** A failed grasping attempt for the instruction "Grab the blue shovel on the table." The top row shows RGB observations, and the bottom row presents the corresponding 3D visualizations with motion trajectories. The gripper retracts without lifting the target object, which remains on the table.

the right side of a table. These examples cover different object properties and interaction patterns, ranging from tool use and deformable object manipulation to articulated object interaction and rigid object relocation. For each example, we show the input RGB image alongside successive 3D visualizations with motion trajectories. Together, these views illustrate how the robot's motion evolves relative to the manipulated objects and the surrounding scene, providing a more detailed view of the spatial and temporal structure of each manipulation sequence.

Failure case analysis. Figure 8 shows a failed grasping attempt for the instruction "Grab the blue shovel on the table." The gripper moves toward the object and subsequently retracts upward, but the target remains on the table. The attempted lifting motion therefore does not result in a successful pickup, indicating that a secure grasp was not established. This example illustrates that an approach-and-lift motion alone is insufficient for successful manipulation: the gripper must also be appropriately aligned with the object and maintain stable contact during lifting. The failure highlights the remaining challenge of reliable grasp execution, particularly during the transition from approaching the object to lifting it.