UNIFIED 4D WORLD ACTION MODELING FROM VIDEO PRIORS WITH ASYNCHRONOUS DENOISING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose X-WAM, a Unified 4D World Model that unifies real-time robotic action execution and high-fidelity 4D world synthesis (video + 3D reconstruction) in a single framework, addressing the critical limitations of prior unified world models (*e.g.*, UWM) that only model 2D pixel-space and fail to balance action efficiency and world modeling quality. To leverage the strong visual priors of pretrained video diffusion models, X-WAM imagines the future world by predicting multi-view RGB-D videos, and obtains spatial information efficiently through a lightweight structural adaptation: replicating the final few blocks of the pretrained Diffusion Transformer into a dedicated depth prediction branch for the reconstruction of future spatial information. Moreover, we propose Asynchronous Noise Sampling (ANS) to jointly optimize generation quality and action decoding efficiency. ANS applies a specialized asynchronous denoising schedule during inference, which rapidly decodes actions with fewer steps to enable efficient real-time execution, while dedicating the full sequence of steps to generate high-fidelity video. Rather than entirely decoupling the timesteps during training, ANS samples from their joint distribution to align with the inference distribution. Pretrained on over 5,800 hours of robotic data, X-WAM achieves 79.2% and 90.7% average success rate on RoboCasa and RoboTwin 2.0 benchmarks, while producing high-fidelity 4D reconstruction and generation surpassing existing methods in both visual and geometric metrics. Our project page is at: `https://anonymous.4open.science/w/X-WAM-Anonymous-BCF1/`

1 INTRODUCTION

The pursuit of general-purpose Embodied AI has been significantly accelerated by the advent of robotic foundation models. Current approaches in this space can be broadly categorized into two paradigms, each targeting a single objective. On the one hand, *policy models* focus on predicting executable actions for robot control. Vision-Language-Action (VLA) models (Zitkovich et al., 2023; Ghosh et al., 2024; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025; Bjorck et al., 2025; Wu et al., 2024) fine-tune pretrained Vision-Language Models (VLMs) to output motor commands, excelling at instruction following and semantic reasoning but lacking the geometric intuition and physical awareness of how actions continuously unfold in the real world (Li et al., 2026b). World Action Models (WAMs) (Li et al., 2026b; Ye et al., 2026b; Yuan et al., 2026) further leverage video generation models to jointly predict future observations and actions, harnessing video priors for stronger physical understanding and generalization. On the other hand, World Models (Hafner et al., 2025; Zhu et al., 2025b; Bruce et al., 2024; Agarwal et al., 2025; Team et al., 2025; Cui et al., 2025) focus on simulating future observations: text-conditioned and action-conditioned world models excel at generating realistic visual predictions of physical dynamics, but do not directly produce executable actions for robot control. These separate paradigms each address a single task, limiting cross-task synergy and representational efficiency.

Recently, a line of work has begun to bridge this divide by constructing unified world action models (Zhu et al., 2025a; Bi et al., 2025; Shen et al., 2025; Liao et al., 2025) that jointly model video generation and action prediction within a single framework. By sharing representations across modalities, these approaches achieve encouraging results in both future prediction quality and policy execution, demonstrating the significant potential of multi-task unified modeling. However, they remain confined to 2D pixel-space observation, lacking explicit spatial awareness and 3D geometric

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

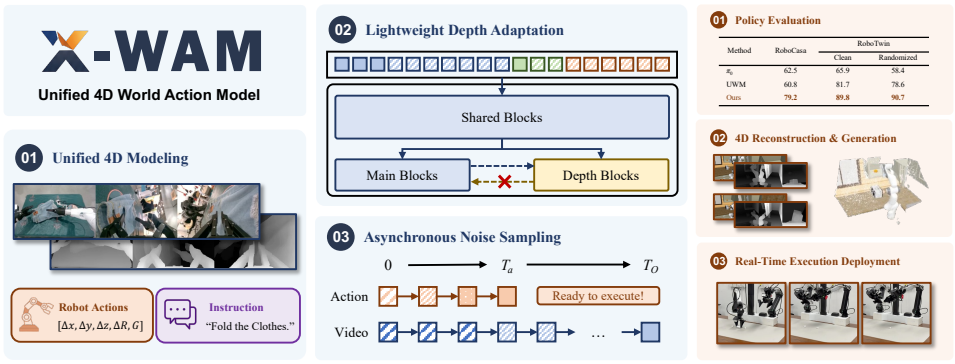


Figure 1: **Overview of X-WAM.** X-WAM is a unified 4D World Action Model that jointly predicts future multi-view RGB-D videos and robot actions from video priors, featuring a lightweight depth adaptation module for spatial reconstruction and Asynchronous Noise Sampling (ANS) for efficient action decoding. X-WAM surpasses existing methods in policy success rate on RoboCasa and RoboTwin 2.0, produces high-fidelity 4D reconstruction and generation, and enables real-time execution deployment on physical robots.

grounding. Since the physical world is fundamentally three-dimensional, this confinement strips away critical geometric structures, causing models to hallucinate physically implausible futures and preventing geometrically faithful 3D reconstruction. To unlock the full potential of unified world action models, it is imperative to elevate them from 2D pixel predictors to spatially aware 4D dynamics simulators that jointly address generation, reconstruction, and policy execution.

Building upon these initial unification efforts, we take a further step by incorporating explicit spatial information into the unified modeling paradigm. We propose X-WAM (Figure 1), a unified 4D World Action Model that simultaneously targets four objectives within a single architecture: high-fidelity video generation, 3D spatial reconstruction, high policy success rate, and efficient action execution. Built on the powerful visual priors of the pretrained video foundation model, X-WAM takes multi-view RGB observations and current robot states as inputs to jointly generate future 4D observations alongside the robot's future states and actions. However, seamlessly integrating 4D spatial awareness and policy execution into such a unified framework presents two fundamental technical challenges.

The first challenge lies in effectively injecting 3D perception into the model without destroying its pretrained knowledge or introducing prohibitive computational overhead. Naive approaches to spatial modeling face distinct drawbacks: concatenating depth maps as additional tokens along the sequence dimension doubles the sequence length, incurring quadratic growth in attention cost; alternatively, fusing depth along the channel dimension shifts the input distribution away from the pretrained manifold, substantially increasing learning difficulty. To circumvent these bottlenecks, we introduce a lightweight structural adaptation. Rather than expanding the denoising sequence, X-WAM explicitly models the 4D world by simply replicating the final few blocks of the pretrained Diffusion Transformer (DiT) (Peebles & Xie, 2023) to construct a dedicated depth prediction branch. This elegant design successfully extracts 3D spatial information without altering the original model's core structure, bypassing the sequence length explosion while strictly preserving the integrity of the pretrained visual priors. As shown in our experiments, this depth supervision not only enables high-quality 3D reconstruction but also consistently improves policy success rates, confirming that explicit spatial modeling benefits multiple objectives of the unified framework simultaneously.

The second challenge stems from the inherent modality mismatch when jointly generating high-dimensional video trajectories and low-dimensional robotic actions. While synthesizing high-fidelity video necessitates numerous denoising steps (Song et al., 2021; Karras et al., 2022), low-dimensional actions require far fewer steps (Chi et al., 2025; Black et al., 2024), and can be accurately recovered even from highly noisy video latents (Bi et al., 2025; Pai et al., 2025; Yuan et al., 2026; Ye et al., 2026b; Ma et al., 2026). Motivated by this insight, we propose Asynchronous Noise Sampling (ANS). ANS introduces a specialized asynchronous denoising schedule for inference: it rapidly decodes precise actions using only a fraction of the initial steps to allow them to be immediately executed by the policy, and subsequently completes the remaining steps to render high-fidelity future videos. Driven by this asynchronous inference characteristic, ANS accordingly reformulates the training-stage sampling strategy. Instead of completely decoupling the noise timesteps of these

modalities via independent random sampling, ANS systematically samples from a joint distribution of video and actions that strictly matches the test-time distribution. This elegantly eliminates the inefficiencies of decoupled sampling, maximizing both action inference speed and visual generation quality.

In summary, our primary contributions are threefold:

- We propose X-WAM, a unified 4D World Action Model that incorporates explicit 3D spatial awareness into the joint video-action modeling paradigm. By introducing a lightweight structural adaptation, we achieve high-quality spatial modeling without doubling sequence lengths or disrupting pretrained visual priors.
- We introduce Asynchronous Noise Sampling (ANS) to enhance the joint generation of videos and actions. By sampling from their joint distribution and employing an asynchronous denoising schedule, ANS improves the training efficiency while maximizing both action decoding speed and video generation quality.
- We demonstrate that X-WAM consistently outperforms all baselines on RoboCasa and RoboTwin 2.0 benchmarks and real-world bimanual manipulation experiments, while producing superior 4D reconstruction and generation performance.

2 Related Work

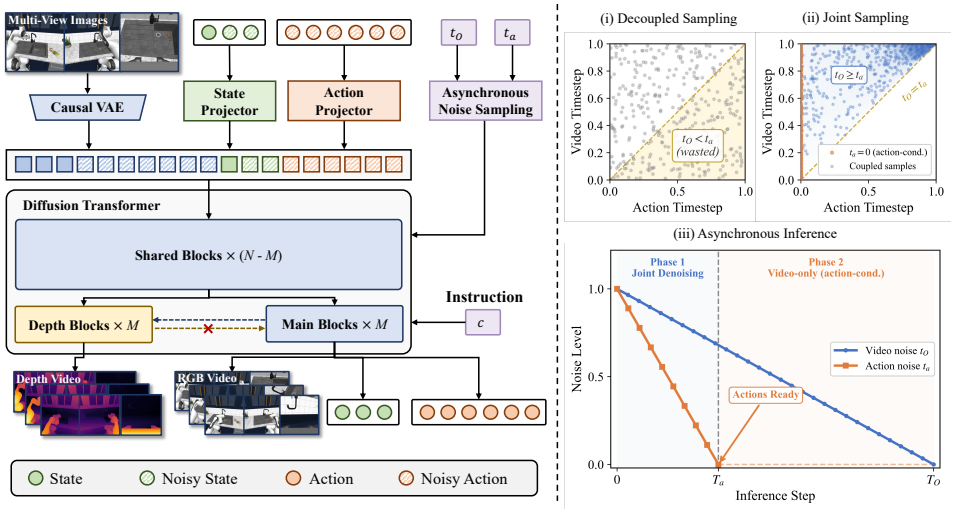
2.1 Unified World Action Modeling

Current general embodied models fall into two complementary paradigms. *Policy models*, predominantly Vision-Language-Action (VLA) models (Zitkovich et al., 2023; Ghosh et al., 2024; Li et al., 2024; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025; Bjorck et al., 2025; Wu et al., 2024; Liu et al., 2025a; Cai et al., 2026), map observations directly to executable robot actions for real-time control. *World models* (Hafner et al., 2025; Zhu et al., 2025b; Bruce et al., 2024; Agarwal et al., 2025; Team et al., 2025; Cui et al., 2025) aim to model environmental dynamics and learn to imagine future observations. Although naturally complementary, the two paradigms have largely evolved in isolation. Some works bridge the gap from the world model side by attaching inverse dynamics models or extracting intermediate representations to convert world models into planners (Du et al., 2023; Hu et al., 2025; Liao et al., 2025; Long et al., 2026; Intelligence et al., 2026). Others augment VLAs with auxiliary future prediction objectives to inject dynamics awareness (Chen et al., 2024; Cen et al., 2025; Zhang et al., 2025a; Wang et al., 2025b; Sun et al., 2026; Hu et al., 2026). While both directions yield improvements, they remain loosely coupled rather than truly unified.

Recently, a line of work has sought to build end-to-end World Action Models (WAMs) from video foundation models. UWM (Zhu et al., 2025a) and Motus (Bi et al., 2025) formulate the problem as a Unified World Model, enabling flexible conditioning and multi-task generation. VideoVLA (Shen et al., 2025) and Cosmos Policy (Kim et al., 2026) directly append action tokens into video sequences for joint prediction. Other works (Pai et al., 2025; Ma et al., 2026; Yuan et al., 2026) employ a Mixture of Transformer architecture with independent parameters and denoising timesteps for each modality. DreamZero (Ye et al., 2026b), LingBot-VA (Li et al., 2026b), and GigaWorld-Policy (Ye et al., 2026a) leverage causal attention masks and KV caching to reduce inference latency. Surveys (Zhang et al., 2026) have shown that such unified approaches generalize better than traditional VLAs. Despite this progress, two limitations persist. First, existing unified models remain confined to 2D pixel-space, lacking explicit 3D spatial awareness. Second, although several recent WAMs (Bi et al., 2025; Zhu et al., 2025a; Ye et al., 2026b) have begun to address the balance between video generation quality and action decoding efficiency by decoupling their sampling timesteps, these approaches rely on independent noise sampling for each modality during training, which introduces a mismatch between the training and inference distributions and thus limits the effectiveness of asynchronous denoising.

2.2 3D Modeling in Embodied Models

Owing to the abundance and accessibility of 2D data, contemporary mainstream embodied models primarily operate within a 2D space for perception, modeling, and prediction. However, the lack of explicit spatial awareness and modeling capabilities, coupled with an over-reliance on purely data-driven fitting, creates a significant bottleneck in tasks that demand spatial comprehension and



(a) Model Architecture (b) Asynchronous Sampling

Figure 2: **Overview of X-WAM.** (a) Model architecture: multi-view RGB observations, proprioceptive states, and noisy actions are encoded and jointly denoised by a Diffusion Transformer initialized from Wan2.2-5B, with a lightweight interleaved depth branch for spatial modeling. (b) Asynchronous Noise Sampling (ANS): (i) standard decoupled sampling wastes training on configurations where $t_O < t_a$; (ii) our coupled joint sampling ensures $t_O \geq t_a$, faithfully matching the inference distribution; (iii) during inference, actions are decoded in T_a steps and immediately dispatched, while video denoising continues for T_O steps.

out-of-distribution generalization. To address this issue, numerous studies have incorporated 3D information into the training pipeline to further enhance the capabilities of embodied models. Within the VLA framework, one category of research (Zhen et al., 2024; Zhang et al., 2025a; Qu et al., 2025; Lin et al., 2025b; Li et al., 2025a; Zhang et al., 2025b) encodes 3D features to serve as predictive targets or supervisory signals within the model's sequence. Another category (Li et al., 2026a; Sun et al., 2025; Li et al., 2025b) directly utilizes explicit 3D representations as inputs, performing predictions natively within the 3D space.

In the context of world models and world action models, several approaches (Zhen et al., 2025; Guo et al., 2026; Huang et al., 2025; Liu et al., 2025b; Huang et al., 2026; Qian et al., 2025; Wang et al., 2026) introduce 3D supervisory signals during the video generation process, endowing the models with multi-view consistency and superior spatial reasoning. ManiGaussian (Lu et al., 2024) and GWM (Lu et al., 2025) construct world models entirely within 3D representations, utilizing the neural rendering technique of 3D Gaussian Splatting (Kerbl et al., 2023) to build high-fidelity 3D world models. Given that current open-source robotic datasets are predominantly composed of 2D videos, existing 3D modeling methods frequently rely on pre-trained feed-forward 3D reconstruction models (Wang et al., 2024; 2025a; Chen et al., 2025a; Lin et al., 2025a) to extract spatial information from robotic data. This strategy effectively transfers the spatial reasoning capabilities of reconstruction models to the video generation pipelines. To the best of our knowledge, no existing work has incorporated explicit spatial information into the unified world action modeling paradigm, nor has any unified model demonstrated the ability to simultaneously serve as a high-fidelity video generator, a 3D reconstruction system, and an efficient policy model within a single framework.

3 Methodology

Building upon recent advances in unified world action modeling, we propose X-WAM as a unified framework that simultaneously addresses video generation, 3D spatial reconstruction, policy success rate, and efficient action execution. This is achieved through two core designs: a lightweight depth adaptation module (Section 3.2) that enables spatial reconstruction, and Asynchronous Noise Sampling (Section 3.3) that jointly optimizes generation quality and action decoding efficiency. We begin by presenting the overall model architecture (Section 3.1), and then detail the training procedure including data processing and the training pipeline (Section 3.4).

3.1 Model Architecture

X-WAM takes a language instruction c, the initial proprioceptive state s_0, and multi-view initial RGB observations O_0 as conditions, and jointly predicts future RGB videos $O_{1:H}$, depth videos $D_{1:H}$, proprioceptive states $s_{1:H}$, and actions $a_{1:K}$, where H and K denote the prediction horizons for video/state and actions, respectively. Following Shen et al. (2025); Ye et al. (2026b), X-WAM is fine-tuned from a pretrained video generation Diffusion Transformer (Peebles & Xie, 2023), specifically Wan2.2-TI2V-5B (Wan et al., 2025) in this work. RGB videos are encoded into latent representations via the original causal VAE encoder $\mathcal{E}$, *i.e.*, $\mathbf{z}_O = \mathcal{E}(O)$, while proprioceptive states and robot actions are projected into the latent space via learnable MLPs: $\mathbf{z}_s = \mathrm{MLP}_s(s)$ and $\mathbf{z}_a = \mathrm{MLP}_a(a)$. After denoising, the clean latents are decoded back to the original physical spaces via symmetric MLPs: $\hat{s} = \mathrm{MLP}_s^{\mathrm{dec}}(\mathbf{z}_s)$ and $\hat{a} = \mathrm{MLP}_a^{\mathrm{dec}}(\mathbf{z}_a)$. The three modalities are concatenated into a unified denoising sequence:

$$\mathbf{Z} = [\mathbf{z}_{O_0}, \mathbf{z}_{O_{1:H}}, \mathbf{z}_{s_0}, \mathbf{z}_{s_{1:H}}, \mathbf{z}_{a_{1:K}}], \tag{1}$$

which is processed with bidirectional full attention, with depth reconstructed from the generated RGB video sequence. The initial observation $\mathbf{z}_{O_0}$ and state $\mathbf{z}_{s_0}$ remain fixed throughout the denoising process with their noise timestep set to $t = 0$ (*i.e.*, treated as clean samples).

Concretely, given 1 conditioning RGB frame and 1 initial state, X-WAM predicts $H = 8$ future RGB frames, $H = 8$ future states, and $K = 32$ future actions. This asymmetric design reflects the different temporal requirements of each modality: actions demand a higher control frequency for smooth and responsive robot execution, while RGB frames and states can be predicted at a lower frequency sufficient for visual generation and 4D reconstruction. The states are temporally aligned with the video frames to enable frame-wise multi-view RGB-D fusion for 3D reconstruction, whereas the actions are uniformly distributed across the same time span at $K/H = 4\times$ the video frame rate.

The original video diffusion model is designed for single-view 2D generation, employing 3D Rotary Position Embeddings (RoPE) (Su et al., 2024) to encode temporal and spatial positions within the sequence. To enable multi-view compatibility without disrupting the pretrained positional encodings, we augment the tokens of each viewpoint with learnable view embeddings to indicate the view index. For proprioceptive states and actions, we apply the same temporal RoPE as the video tokens along the time dimension, allowing the model to infer the temporal correspondence between states/actions and video frames through positional proximity.

X-WAM is designed to simultaneously reconstruct and generate the future world. Reconstructing 3D representations (*e.g.*, point clouds) from multi-view RGB-D outputs requires camera poses for each viewpoint. Unlike prior works (Huang et al., 2025) that explicitly encode camera extrinsics or ray direction maps as tokens, we adopt a more principled approach grounded in the structure of robotic systems. We observe that cameras in robotic manipulation setups can be categorized into two types: *static* cameras (first-person and third-person views), whose poses remain constant throughout task execution, and *dynamic* cameras (wrist-mounted), which are rigidly attached to the robot arm with a fixed hand-eye calibration matrix that depends solely on the robot model. Therefore, instead of predicting explicit camera extrinsics, X-WAM predicts the end-effector pose $\mathbf{T}_{\text{ee}} \in SE(3)$ and derives the wrist camera pose via the fixed hand-to-eye calibration matrix $\mathbf{T}_{\text{h2e}}$:

$$\mathbf{T}_{\text{wrist}} = \mathbf{T}_{\text{ee}} \cdot \mathbf{T}_{\text{h2e}}. \tag{2}$$

This conversion enables the fusion of 3D information across all viewpoints to reconstruct a unified 3D representation.

3.2 Lightweight Depth Adaptation

In X-WAM, depth maps are predicted in the same VAE latent space as RGB frames: each single-channel depth map is replicated three times along the channel dimension to form a pseudo-RGB image, encoded by the same causal VAE $\mathcal{E}$, and decoded back via the symmetric VAE decoder after denoising. Rather than concatenating depth tokens along the sequence dimension (which doubles attention cost) or fusing along the channel dimension (which shifts the input distribution away from the pretrained manifold), we propose a lightweight depth adaptation module that modifies the pretrained DiT architecture. Specifically, given a model with N DiT blocks, we replicate the final M blocks ($M < N$) to construct an auxiliary depth prediction branch. After the shared first

$N-M$ blocks produce hidden states $\mathbf{H}$, the depth branch and the main branch are initialized as $\mathbf{Z}_D^{(0)} = \mathbf{Z}_{\text{m}}^{(0)} = \mathbf{H}$ and executed in an *interleaved* fashion. At each layer $j \in \{1, \ldots, M\}$:

$$\mathbf{Z}_D^{(j)} = \text{DepthBlock}_j\left(\mathbf{Z}_D^{(j-1)} \mid \mathbf{Z}_{\text{m}}^{(j-1)}\right), \quad \mathbf{Z}_{\text{m}}^{(j)} = \text{DiTBlock}_{N-M+j}\left(\mathbf{Z}_{\text{m}}^{(j-1)}\right), \tag{3}$$

where DepthBlock_j attends to the main branch's input $\mathbf{Z}_{\text{m}}^{(j-1)}$ at the same layer via cross-attention, while the main branch remains unaffected by depth tokens. We term this asymmetric connectivity *unilateral attention*: the depth branch can read from the main branch, but not vice versa, thereby strictly preserving the integrity of the pretrained weights. The depth branch is trained to regress the inverse depth of the current video frame using mean squared error (MSE) loss, consistent with established depth estimation models (Chen et al., 2025a; Lin et al., 2025a).

This design rests on the fundamental assumption that depth information can be inferred from RGB features without requiring fully independent generation. The detailed single-step procedure is presented in Algorithm 1 (Appendix A). Depth supervision during training enhances the model's spatial structure perception. During inference, this auxiliary depth branch does not need to participate in every denoising step and can be flexibly toggled on or off, substantially reducing rollout overhead.

3.3 Asynchronous Noise Sampling

Prior works (Bi et al., 2025; Pai et al., 2025; Ye et al., 2026b; Ma et al., 2026; Yuan et al., 2026) decouple the noise timesteps of video and actions to allow faster action decoding during inference. However, independently sampling each modality's timestep introduces training configurations that never arise at test time (*e.g.*, $t_O < t_a$), degrading training efficiency. To align the training and inference noise distributions, X-WAM adopts Asynchronous Noise Sampling (ANS). The complete procedure is detailed in Algorithm 2 (Appendix A).

Asynchronous inference. During inference, ANS applies asynchronous denoising timesteps for video and actions, as illustrated in Figure 2(b-iii). We allocate T_a denoising steps for proprioceptive states and actions, and T_O denoising steps for video ($T_a < T_O$). Both modalities start from pure noise and are denoised with step sizes of $\frac{1}{T_a}$ and $\frac{1}{T_O}$, respectively. After T_a forward passes, the noise-free actions are obtained and can be immediately dispatched to the downstream robot for execution. If a clear, complete video is desired, the remaining $T_O - T_a$ steps are continued, during which the actions serve as a clean modality and undergo no further denoising. In this regime, the inference process naturally becomes an action-conditioned world model.

Coupled noise sampling during training. To realize this asynchronous inference behavior, we also apply different noise timesteps to video and actions during training. However, unlike prior works that sample from two independent distributions, we place the video and action noise levels into a joint distribution and perform coupled sampling. Formally, the joint noise level (t_O, t_a) is drawn from the following mixture:

$$(t_O, t_a) \sim \begin{cases} t_a = 0, \;\; t_O \sim \text{U}(0,1) & \text{w.p. } p, \\ t_a \sim \text{U}(0,1), \;\; t_O = t_a + (1-t_a)\cdot b, \;\; b \sim \text{Beta}(1.5,1) & \text{w.p. } 1-p, \end{cases} \tag{4}$$

where the first case corresponds to action-conditioned video generation with noise-free actions, and the second case represents asynchronous joint generation. The $\text{Beta}(1.5, 1)$ distribution, rescaled to $[t_a, 1]$, biases t_O toward higher noise levels, reflecting the fact that video typically requires more denoising steps than actions. Crucially, t_O is sampled *conditioned on* t_a, making them dependent rather than independent random variables. This coupled sampling strategy more faithfully reflects the inference-time distribution, enabling more efficient training of the WAM.

3.4 Training Details

Consistent with the pretrained Wan2.2-5B model (Wan et al., 2025), we fine-tune X-WAM using the flow matching framework (Lipman et al., 2023). The model f_θ is trained to predict the velocity field $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{z}^0$ given noisy inputs at timestep t. For a modality $m \in \{O, s, a\}$ with corresponding timestep t_m, the velocity prediction loss is:

$$\mathcal{L}_m = \left\| f_\theta^m(\mathbf{z}_m^{t_m}, t_m) - (\boldsymbol{\epsilon}_m - \mathbf{z}_m^0) \right\|^2, \tag{5}$$

where t_O and t_a denote the video and action noise timesteps sampled via ANS (Eq. 4), with $t_s = t_a$. The depth branch is supervised with a direct MSE regression loss on inverse depth: $\mathcal{L}_{\text{depth}} = \left\| \hat{D} - D^* \right\|^2$, where D^* denotes the ground-truth inverse depth. The total training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_O + \lambda_s \mathcal{L}_s + \lambda_a \mathcal{L}_a + \lambda_D \mathcal{L}_{\text{depth}}, \tag{6}$$

where λ_s, λ_a, and λ_D are weighting coefficients.

To build a unified 4D model capable of generation, reconstruction, and manipulation, we train the model on over 5,800 hours of data, encompassing both real-robot and simulated datasets spanning diverse manipulation tasks. All datasets undergo preprocessing and filtering, and are unified into a consistent coordinate system and representation. We define a universal interface to represent robot states and actions across heterogeneous datasets. The state is defined as the poses and gripper positions of a dual-arm end-effector, while the action is defined as the corresponding changes in end-effector poses and gripper positions. For single-arm robots, we treat the single arm as the left arm and do not supervise the right-arm output. During inference, we employ the UniPC (Zhao et al., 2023) multistep scheduler recommended by Wan2.2, maintaining separate scheduler instances with different step sizes for video and state/action modalities, following the asynchronous inference procedure described in Section 3.3. Full details on pretraining data, implementation, and baselines are provided in Appendix B.

4 Experiments

We evaluate X-WAM across three complementary dimensions: policy execution (Section 4.1), 4D reconstruction and generation (Section 4.2), and ablation studies that jointly analyze both objectives (Section 4.3). We additionally deploy X-WAM on a real-world dual-arm platform and report both qualitative rollouts and quantitative results on bimanual manipulation tasks (Appendix C).

4.1 Policy Evaluation

We first evaluate the policy execution capability of X-WAM by deploying it in closed-loop simulation and measuring task success rates on two representative robotic manipulation benchmarks.

RoboCasa (Nasiriany et al., 2024) is a large-scale simulation benchmark featuring diverse kitchen manipulation tasks with realistic scenes and object variations. We report the average success rate (SR) across 24 manipulation tasks. We compare against three VLA baselines: π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), and Spatial Forcing (Li et al., 2025a), and three WAM baselines: UWM (Zhu et al., 2025a), GigaWorld-Policy (Ye et al., 2026a), and Motus (Bi et al., 2025).

RoboTwin 2.0 (Chen et al., 2025b) is a dual-arm manipulation benchmark that evaluates policy generalization under two settings: *Clean*, where the environment matches a certain distribution, and *Randomized*, where object poses, appearances, and distractors are randomized to test robustness. Following Bi et al. (2025), we train X-WAM on all trajectories of AgileX arms, including 50 clean and 500 randomized trajectories on 50 tasks. We compare against three VLA baselines: π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), and Spatial Forcing (Li et al., 2025a), and three WAM baselines: UWM (Zhu et al., 2025a), GigaWorld-Policy (Ye et al., 2026a), and Motus (Bi et al., 2025).

As shown in Table 1, X-WAM consistently outperforms all baselines on both benchmarks. On RoboCasa, X-WAM attains 79.2% average SR, surpassing the strongest baseline GigaWorld-Policy (68.9%) by 10.3 percentage points. On RoboTwin 2.0, X-WAM achieves 89.8% and 90.7% under the Clean and Randomized settings, respectively, outperforming the prior method Motus (88.7% / 87.0%) across both protocols. These results validate that incorporating explicit 3D spatial awareness and large-scale pretraining into the unified world action modeling framework yields substantial performance gains.

4.2 4D Reconstruction and Generation

We next evaluate the 4D reconstruction and generation capabilities of X-WAM on the RoboCasa environment. Specifically, we execute the policy in simulation and compare the predicted multi-view

Table 1: Average success rate (%) on the RoboCasa benchmark (24 manipulation tasks) and the RoboTwin 2.0 benchmark (50 tasks, under the *Clean* and *Randomized* settings). Blank cells indicate that the method has not been evaluated on that benchmark. **Bold**: best.

Method	RoboCasa	RoboTwin 2.0	
	Avg SR	Clean	Randomized
VLA-based methods			
π_0 (Black et al., 2024)	62.5	65.9	58.4
$\pi_{0.5}$ (Intelligence et al., 2025)	65.8	82.7	76.8
Spatial Forcing (Li et al., 2025a)	67.8	85.6	84.8
WAM-based methods			
UWM (Zhu et al., 2025a)	60.8	81.7	78.6
GigaWorld-Policy (Ye et al., 2026a)	68.9	87.0	85.0
Motus (Bi et al., 2025)	67.6	88.7	87.0
X-WAM (Ours)	**79.2**	**89.8**	**90.7**

Table 2: 4D reconstruction quality on RoboCasa. ↑ indicates higher is better; ↓ indicates lower is better.

Method	RGB			Depth		Point Cloud
	PSNR↑	SSIM↑	LPIPS↓	AbsRel↓	δ_1↑	CD↓
DreamZero (Ye et al., 2026b) + DA3 (Lin et al., 2025a)	21.12	0.7788	0.1580	0.1362	0.8594	0.0680
Robot4DGen (Liu et al., 2025b)	22.67	0.8207	0.1026	0.0736	0.9443	0.0134
X-WAM w/o depth + DA3 (Lin et al., 2025a)	23.09	0.8916	0.0548	0.1045	0.9089	0.0401
X-WAM (Ours)	**23.46**	**0.8942**	**0.0513**	**0.0349**	**0.9738**	**0.0049**

RGB-D observations against the ground-truth observations rendered by the simulator. We adopt three groups of metrics: PSNR, SSIM, and LPIPS for visual fidelity, absolute relative error (AbsRel) and δ_1 accuracy for depth quality, and Chamfer Distance (CD) for the quality of the reconstructed point clouds. The depth metrics (AbsRel, δ_1) evaluate single-view reconstruction capability, measuring how accurately the model predicts per-pixel geometry from each individual viewpoint. In contrast, Chamfer Distance evaluates multi-view consistency and the accuracy of camera extrinsic estimation (for static cameras) and robot proprioceptive state prediction (for wrist cameras), as it requires lifting per-view depth maps into a shared 3D coordinate frame and fusing them into a unified point cloud. Note that pixel-level metrics (PSNR, SSIM, LPIPS, AbsRel, δ_1) are computed only on the two static cameras (first-person and third-person views), as the wrist camera suffers from pixel misalignment due to minor errors in the predicted end-effector pose, rendering per-pixel comparison unreliable. The quality of wrist-camera predictions can instead be assessed through Chamfer Distance and qualitative visualizations. As baselines, we consider a two-stage approach that combines DreamZero (Ye et al., 2026b) for RGB video generation with Depth Anything 3 (Lin et al., 2025a) for post-hoc depth estimation, and Robot4DGen (Liu et al., 2025b), a geometry-aware 4D video generation method. We also include an ablative variant, X-WAM w/o depth + DA3, which removes our depth branch and instead relies on Depth Anything 3 for depth estimation.

As shown in Table 2, X-WAM achieves the best performance across all metrics. Compared with the two-stage pipeline of DreamZero + DA3, X-WAM improves PSNR by 2.34 dB and reduces Chamfer Distance from 0.0680 to 0.0049, demonstrating that end-to-end joint modeling produces substantially more accurate spatial reconstructions than post-hoc depth estimation applied to independently generated videos. Robot4DGen, which incorporates geometric priors during generation, achieves competitive depth metrics but falls short on visual fidelity (LPIPS 0.1026 vs. 0.0513). Notably, replacing our depth branch with Depth Anything 3 (X-WAM w/o depth + DA3) preserves strong RGB quality but degrades depth accuracy (AbsRel 0.1045 vs. 0.0349) and point cloud quality (CD 0.0401 vs. 0.0049), confirming that the integrated depth branch produces more geometrically consistent predictions than a general-purpose monocular estimator.

4.3 Ablation Studies

We conduct ablation studies on the RoboCasa benchmark to validate the key design choices of X-WAM. Due to computational constraints, all ablation variants are fine-tuned directly from the

Table 3: Ablation studies on the RoboCasa benchmark. **Bold**: best; underline: second best.

Variant	SR↑	Latency (ms)↓	RGB			Depth		Point Cloud
			PSNR↑	SSIM↑	LPIPS↓	AbsRel↓	δ_1↑	CD↓
(a) Depth architecture design								
No depth	63.0	**1033**	23.09	0.8916	0.0548	–	–	–
Sequence concatenation	**68.7**	1888	**23.60**	**0.8987**	**0.0488**	**0.0332**	**0.9774**	**0.0037**
Channel concatenation	64.2	1266	23.20	0.8933	0.0522	0.0377	0.9728	0.0052
Interleaved branch (Ours)	67.8	**1033**	23.46	0.8942	0.0513	0.0349	0.9738	0.0049
(b) Noise scheduling strategy								
Sync train + Sync infer	66.4	4665	**23.48**	**0.8963**	**0.0502**	0.0375	0.9725	**0.0041**
Decoupled train + Sync infer	66.3	4665	23.17	0.8937	0.0529	0.0397	0.9701	0.0050
Decoupled train + Async infer	67.2	**1033**	22.60	0.8878	0.0561	0.0430	0.9680	0.0061
ANS train + Async infer (Ours)	**67.8**	**1033**	23.46	0.8942	0.0513	**0.0349**	**0.9738**	0.0049

Wan2.2-TI2V-5B weights on the benchmark data without the large-scale pretraining stage described in Section 3.4. Unless otherwise stated, all variants share the same data, hyperparameters, and training schedule. Action latency is measured on a single NVIDIA RTX 3090 GPU.

Depth architecture design. We compare four depth incorporation strategies (Table 3(a)). The Latency column reports the action generation latency during policy execution. Sequence concatenation achieves the best quality metrics by treating depth as explicit tokens, but nearly doubles the action latency to 1888 ms due to the expanded sequence length. Channel concatenation also introduces noticeable overhead (1266 ms). In contrast, our interleaved branch matches the latency of the no-depth variant (1033 ms), since the depth branch can be toggled off during action decoding, while delivering clearly superior quality over both the no-depth and channel-concatenation variants. Notably, removing depth supervision entirely causes the policy success rate to drop from 67.8% to 63.0%, confirming that explicit spatial modeling is essential for robust manipulation. Channel concatenation also underperforms in success rate (64.2%), as fusing depth along the channel dimension shifts the input distribution away from the pretrained manifold.

Effect of ANS. We compare four noise scheduling configurations (Table 3(b)), where synchronous variants use 25 joint steps and asynchronous variants decode actions in only 5 steps. The synchronous baselines yield strong RGB metrics but force the policy to wait for all 25 denoising steps, resulting in an action latency of 4665 ms. Asynchronous inference reduces this to 1033 ms, a $4.5\times$ speedup, by decoding actions in only the first 5 steps. Among the asynchronous variants, Decoupled-Async achieves a competitive success rate (67.2%) but its reconstruction quality degrades significantly (PSNR 22.60, AbsRel 0.0430), because the video branch must continue denoising conditioned on clean actions, a regime never seen during independently sampled training. Our ANS closes this gap by coupling the training noise distribution to faithfully cover the asynchronous inference regime. As a result, ANS achieves the highest success rate (67.8%) and the best depth metrics at the same 1033 ms latency, while maintaining RGB quality competitive with the synchronous baseline.

5 Conclusion

In this work, we presented X-WAM, a unified 4D World Action Model that extends unified world action modeling into spatially aware 4D dynamics simulation. Through a lightweight depth adaptation module that replicates the final DiT blocks as an interleaved depth branch, X-WAM achieves high-quality spatial reconstruction without increasing sequence length or compromising pretrained visual priors. Asynchronous Noise Sampling further aligns training and inference noise distributions across modalities, enabling rapid action decoding while preserving video generation quality. Experiments on RoboCasa and RoboTwin 2.0 demonstrate that X-WAM consistently outperforms all baselines in both policy success rate and 4D reconstruction quality, confirming that a single framework can jointly optimize policy execution, visual generation, and spatial reconstruction.

AI USE STATEMENT

We have not used generative AI tools for any of the tasks with required disclosure, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for two tasks with recommended disclosure: editing the paper to improve readability, and searching for and identifying relevant literature. We have reviewed all AI-assisted work: all AI-edited text was checked against our original wording to ensure that no claim or result was altered, and every reference identified with AI assistance was verified against the original publication before being cited. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies a unified world action model for robotic manipulation, and we do not foresee ethical concerns arising from it. The research involves no human subjects, no crowdsourcing, and no personally identifiable or otherwise sensitive data; all real-robot data were collected by the authors in a controlled laboratory workspace containing only everyday objects. We introduce no new dataset, and all datasets, benchmarks, pretrained models, and baselines we build on are publicly available, cited at the point of use, and used in accordance with their licenses and intended research use. Our real-robot experiments were conducted by trained operators with an emergency stop available throughout. We therefore see no foreseeable risk specific to this work beyond those common to general-purpose robotics research, and we adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We have aimed to make X-WAM reproducible from the paper alone. The model architecture, the lightweight depth adaptation module, and the Asynchronous Noise Sampling objective are specified in Section 3, with the complete denoising and ANS procedures given as Algorithms 1 and 2 in Appendix A. Appendix B reports the pretraining corpus and its preprocessing (Table 4), the universal state and action representation, the full set of hyperparameters and hardware for both the pretraining and fine-tuning stages, and the exact inference configuration; the real-robot protocol is described in Appendix C and the latency measurement protocol in Appendix D. We further commit to releasing our code upon acceptance.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *CoRR*, abs/2512.13030, 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith LLontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024.

Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *CoRR*, abs/2602.12684, 2026.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. Worldvla: Towards autoregressive action world model. *CoRR*, abs/2506.21539, 2025.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22831–22840, 2025a.

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*, 2025b.

Yi Chen, Yuying Ge, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for robot manipulation. *CoRR*, abs/2412.04445, 2024.

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, 2025.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3.5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 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. In *Advances in Neural Information Processing Systems*, 2023.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

Jun Guo, Xiaojian Ma, Yikai Wang, Min Yang, Huaping Liu, and Qing Li. Flowdreamer: A RGB-D world model with flow-based motion representations for robot manipulation. *IEEE Robotics Automation Letters*, 11(3):2466–2473, 2026.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

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

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, Wei Li, and Jianyu Chen. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *CoRR*, abs/2602.09849, 2026.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yutao Hu, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse: Envisioning embodied future space for robotics manipulation. *CoRR*, abs/2501.01895, 2025.

Wenbing 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. *CoRR*, abs/2601.03782, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025.

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

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in neural information processing systems*, volume 35, pp. 26565–26577, 2022.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pp. 2679–2713, 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, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *CoRR*, abs/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 Autom. Lett.*, 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. *CoRR*, abs/2510.12276, 2025a.

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

Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *CoRR*, abs/2506.07961, 2025b.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *The Twelfth International Conference on Learning Representations*, 2024.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Hu Yue, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie envisioner: A unified world foundation platform for robotic manipulation. *CoRR*, abs/2508.05635, 2025.

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. *CoRR*, abs/2511.10647, 2025a.

Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *CoRR*, abs/2507.00416, 2025b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

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 *The Thirteenth International Conference on Learning Representations*, 2025a.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4d video generation for robot manipulation. *CoRR*, abs/2507.01099, 2025b.

Qian Long, Yueze Wang, Jiaxi Song, Junbo Zhang, Peiyan Li, Wenxuan Wang, Yuqi Wang, Haoyang Li, Shaoxuan Xie, Guocai Yao, Hanbo Zhang, Xinlong Wang, Zhongyuan Wang, Xuguang Lan, Huaping Liu, and Xinghang Li. Scaling world model for hierarchical manipulation policies. *CoRR*, abs/2602.10983, 2026.

Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *18th European Conference on Computer Vision*, pp. 349–366, 2024.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. GWM: towards scalable gaussian world models for robotic manipulation. *CoRR*, abs/2508.17600, 2025.

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. *CoRR*, abs/2603.10448, 2026.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *RSS Workshop: Data Generation for Robotics*, 2024.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Zezhong Qian, Xiaowei Chi, Yuming Li, Shizun Wang, Zhiyuan Qin, Xiaozhu Ju, Sirui Han, and Shanghang Zhang. Wristworld: Generating wrist-views via 4d world models for robotic manipulation. *CoRR*, abs/2510.07313, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *CoRR*, abs/2501.15830, 2025.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *CoRR*, abs/2512.06963, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: enhancing vision-language-action model with latent world model. *CoRR*, abs/2602.10098, 2026.

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

GigaWorld Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jiagang Zhu, Kerui Li, Mengyuan Xu, et al. Gigaworld-0: World models as data engine to empower embodied ai. *arXiv preprint arXiv:2511.19861*, 2025.

Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, et al. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 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.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. VGGT: visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, and Xiangyu Yue. MVISTA-4D: view-consistent 4d world model with test-time action inference for robotic manipulation. *CoRR*, abs/2602.09878, 2026.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *CoRR*, abs/2506.19850, 2025b.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *The Twelfth International Conference on Learning Representations*, 2024.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigaworld-policy: An efficient action-centered world-action model. *CoRR*, abs/2603.17240, 2026a.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *CoRR*, abs/2602.15922, 2026b.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *CoRR*, abs/2603.16666, 2026.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *CoRR*, abs/2507.04447, 2025a.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, Feng Wen, Xinyu Wang, Xingyue Quan, and Yingxue Zhang. Do world action models generalize better than vlas? A robustness study. *CoRR*, abs/2603.22078, 2026.

Zhengshen Zhang, Hao Li, Yalun Dai, Zhengbang Zhu, Lei Zhou, Chenchen Liu, Dong Wang, Francis E. H. Tay, Sijin Chen, Ziwei Liu, Yuxiao Liu, Xinghang Li, and Pan Zhou. From spatial to actions: Grounding vision-language-action model in spatial foundation priors. *CoRR*, abs/2510.17439, 2025b.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49842–49869, 2023.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. In *Forty-first International Conference on Machine Learning*, pp. 61229–61245, 2024.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: Learning 4d embodied world models. *CoRR*, abs/2504.20995, 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *CoRR*, abs/2504.02792, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025b.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183, 2023.

A Detailed Algorithms

We provide the complete algorithmic procedures for X-WAM. Algorithm 1 details the single denoising step, which jointly processes the multi-modal sequence through the shared DiT trunk and the interleaved depth branch to produce velocity predictions and depth estimates. Algorithm 2 presents the full Asynchronous Noise Sampling (ANS) procedure for both training and inference, illustrating how coupled noise sampling during training aligns with the asynchronous denoising schedule at inference time.

Algorithm 1 DENOISE: Single Denoising Step of X-WAM

Require: Noisy video latent $\mathbf{z}_O^{t_O}$, noisy state $\mathbf{z}_s^{t_a}$, noisy action $\mathbf{z}_a^{t_a}$, video timestep t_O, action timestep t_a, initial observation O_0, initial state s_0, language instruction c

Ensure: Predicted velocities $\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_a$; predicted inverse depth $\hat{D}$

$\mathbf{z}_{O_0} \leftarrow \mathrm{CausalVAE}(O_0)$; $\mathbf{z}_{s_0} \leftarrow \mathrm{MLP}_s(s_0)$ ▷ Encode conditions with $t = 0$

$\mathbf{Z} \leftarrow \mathrm{Concat}(\mathbf{z}_{O_0}, \mathbf{z}_O^{t_O}, \mathbf{z}_{s_0}, \mathbf{z}_s^{t_a}, \mathbf{z}_a^{t_a})$

Add learnable view embeddings to $\mathbf{Z}$

for $i = 1$ **to** $N - M$ **do** ▷ Shared trunk

  $\mathbf{Z} \leftarrow \mathrm{DiTBlock}_i(\mathbf{Z})$

end for

$\mathbf{Z}_\mathrm{m} \leftarrow \mathbf{Z}$; $\mathbf{Z}_D \leftarrow \mathbf{Z}$ ▷ Initialize main and depth branches

for $j = 1$ **to** M **do** ▷ Interleaved main-depth processing

  $\mathbf{Z}_D \leftarrow \mathrm{DepthBlock}_j(\mathbf{Z}_D \mid \mathbf{Z}_\mathrm{m})$ ▷ Depth attends to main branch's input

  $\mathbf{Z}_\mathrm{m} \leftarrow \mathrm{DiTBlock}_{N-M+j}(\mathbf{Z}_\mathrm{m})$ ▷ Main branch

end for

$\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_a \leftarrow \mathrm{Head}_{\text{main}}(\mathbf{Z}_\mathrm{m})$

$\hat{D} \leftarrow \mathrm{Head}_{\text{depth}}(\mathbf{Z}_D)$ ▷ Regress inverse depth

return $\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_a, \hat{D}$

B Training Details

B.1 Pretraining Data

Table 4 summarizes the pretraining datasets used in X-WAM. The data spans both real-robot and simulated environments, totaling over 1.49 million episodes and approximately 5,874 hours. All datasets undergo careful preprocessing and filtering: we remove episodes containing base locomotion, dexterous manipulation, and failed executions. Following Black et al. (2024), we additionally filter out stationary frames from the DROID dataset. All videos are uniformly downsampled to 3.75 FPS and resized to a resolution of 320×256. Since most pretraining datasets lack depth annotations, we extract depth maps from all training videos using Video Depth Anything (Chen et al., 2025a).

Table 4: Summary of pretraining datasets used in X-WAM.

Dataset	Source	Episodes	Duration (h)
AgibotWorld-Beta (Bu et al., 2025)	Real	866,562	2,221.5
DROID (Khazatsky et al., 2024)	Real	74,734	280.3
InternA1-Aloha (Tian et al., 2025)	Sim	184,803	1,337.3
InternA1-Genie1 (Tian et al., 2025)	Sim	50,638	174.0
InternA1-Lift2 (Tian et al., 2025)	Sim	231,018	1,464.7
RoboCasa MimicGen (Nasiriany et al., 2024; Mandlekar et al., 2023)	Sim	56,771	282.4
RoboTwin 2.0 (Chen et al., 2025b)	Sim	27,500	113.7
Total	–	1,492,026	5,873.9

B.2 Implementation Details

State and action representation. To unify heterogeneous single-arm and dual-arm robots across datasets, we define a universal state and action interface based on end-effector poses. The state is

Algorithm 2 Asynchronous Noise Sampling (ANS): Training and Inference

— Training: Coupled Noise Sampling —

Require: Clean video latent $\mathbf{z}_O^0$, clean state $\mathbf{z}_s^0$, clean action $\mathbf{z}_a^0$, probability p

Ensure: Noisy samples $\mathbf{z}_O^{t_O}, \mathbf{z}_s^{t_a}, \mathbf{z}_a^{t_a}$ with coupled timesteps t_O, t_a

Draw $u \sim \mathrm{U}(0,1)$

if $u < p$ **then** ▷ Action-conditioned video generation

  $t_a \leftarrow 0;\ t_O \sim \mathrm{U}(0,1)$

else ▷ Asynchronous joint generation

  $t_a \sim \mathrm{U}(0,1);\ b \sim \mathrm{Beta}(1.5,\ 1)$

  $t_O \leftarrow t_a + (1 - t_a) \cdot b$ ▷ Rescale to $[t_a, 1]$, ensuring $t_O \geq t_a$

end if

$\boldsymbol{\epsilon}_O, \boldsymbol{\epsilon}_s, \boldsymbol{\epsilon}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathbf{z}_O^{t_O} \leftarrow (1 - t_O)\,\mathbf{z}_O^0 + t_O\,\boldsymbol{\epsilon}_O;\ \mathbf{z}_s^{t_a} \leftarrow (1 - t_a)\,\mathbf{z}_s^0 + t_a\,\boldsymbol{\epsilon}_s;\ \mathbf{z}_a^{t_a} \leftarrow (1 - t_a)\,\mathbf{z}_a^0 + t_a\,\boldsymbol{\epsilon}_a$ ▷ Flow matching interpolation

Compute $\mathcal{L}_{\text{total}}$ via DENOISE (Algorithm 1) and backpropagate

— Inference: Asynchronous Denoising —

Require: Conditions (O_0, s_0, c), video steps T_O, action steps T_a ($T_a < T_O$)

Ensure: Denoised video $\mathbf{z}_O$, state $\mathbf{z}_s$, action $\mathbf{z}_a$

$\mathbf{z}_O, \mathbf{z}_s, \mathbf{z}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Initialize from pure noise

Initialize schedulers: $\mathcal{S}_O$ with T_O steps, $\mathcal{S}_a$ with T_a steps

for $k = 1$ **to** T_O **do**

  Get current timestep t_O from $\mathcal{S}_O$

  if $k \leq T_a$ **then** ▷ Joint denoising phase

    Get current timestep t_a from $\mathcal{S}_a$

    $\hat{\mathbf{v}}_O, \hat{\mathbf{v}}_s, \hat{\mathbf{v}}_a, _ \leftarrow \text{DENOISE}(\mathbf{z}_O, \mathbf{z}_s, \mathbf{z}_a, t_O, t_a, O_0, s_0, c)$

    $\mathbf{z}_O \leftarrow \mathcal{S}_O.\text{step}(\mathbf{z}_O, \hat{\mathbf{v}}_O);\ \mathbf{z}_s \leftarrow \mathcal{S}_a.\text{step}(\mathbf{z}_s, \hat{\mathbf{v}}_s);\ \mathbf{z}_a \leftarrow \mathcal{S}_a.\text{step}(\mathbf{z}_a, \hat{\mathbf{v}}_a)$

  else ▷ Video-only denoising phase (action-conditioned)

    $\hat{\mathbf{v}}_O, _, _, _ \leftarrow \text{DENOISE}(\mathbf{z}_O, \mathbf{z}_s, \mathbf{z}_a, t_O, 0, O_0, s_0, c)$

    $\mathbf{z}_O \leftarrow \mathcal{S}_O.\text{step}(\mathbf{z}_O, \hat{\mathbf{v}}_O)$

  end if

end for

return $\mathbf{z}_O,\ \mathbf{z}_s,\ \mathbf{z}_a$ ▷ Actions available after step T_a; video after step T_O

represented as a 16-dimensional absolute vector: (position$_3$ + quaternion$_4$ + gripper$_1$) $\times$ 2 arms. The action is represented as a 14-dimensional relative vector: (Δposition$_3$ + Δaxis-angle$_3$ + gripper action$_1$) $\times$ 2 arms. For single-arm robots, only the first 8 dimensions of the state and the first 7 dimensions of the action are supervised. We compute per-dataset quantile statistics ($q_{0.01}$, $q_{0.99}$) for normalization. Notably, the action normalization applies only scaling without bias, preserving the semantics that a zero action corresponds to no movement across all datasets.

Large-scale pretraining. We pretrain X-WAM on 256 NVIDIA H20 GPUs with a per-GPU batch size of 8 (total batch size 2,048). We use the AdamW optimizer with a peak learning rate of 1×10^{-4}, 1,000 steps of linear warmup followed by cosine decay to 0, and train for 40,000 steps. The prediction horizon is set to $H = 8$. The loss weighting coefficients are $\lambda_s = 1.0$, $\lambda_a = 1.0$, and $\lambda_D = 1.0$. The number of replicated depth blocks is $M = 10$, and the ANS action-conditioned probability is $p = 0.5$.

Benchmark fine-tuning. For RoboCasa and RoboTwin 2.0, we further fine-tune the pretrained model on the respective benchmark data using 32 NVIDIA H20 GPUs with a per-GPU batch size of 4 (total batch size 128), a learning rate of 3×10^{-5}, and the same warmup and cosine decay schedule. Fine-tuning proceeds for 20,000 steps. To obtain ground-truth depth maps for fine-tuning, we replay the official demonstration data in the simulator, ensuring that the total data volume and initial configurations remain unchanged and that the replay random seeds do not overlap with those used at test time. For RoboCasa, we directly use the raw actions provided in the dataset as supervision signals. For RoboTwin 2.0, we use relative actions that are converted to absolute end-effector poses based on the state of the first frame in each action chunk before being sent to the simulator for execution.

Inference. We use asynchronous denoising with $T_a = 10$ action denoising steps and $T_O = 50$ video denoising steps, following the UniPC (Zhao et al., 2023) scheduler. The classifier-free guidance scale is set to 1.0, as we empirically find that larger guidance scales do not improve action quality but increase inference cost. For both benchmarks, each task is evaluated over 100 episodes and the success rate is averaged. All other evaluation settings follow the official benchmark protocols.

C Real-Robot Experiments

To validate the practical applicability of X-WAM, we deploy the model on a real-world dual-arm robotic platform. We first present qualitative rollouts on an earphone packing task (Appendix C.1), and then report quantitative results on three additional bimanual tasks under both in-domain and out-of-distribution conditions (Appendix C.2).

Setup. All experiments are conducted on an AC One dual-arm platform equipped with one main camera providing a global view and two wrist-mounted cameras for close-up observations, all operating at a resolution of 320×256. We collect approximately 20 hours of demonstration data for the earphone packing task. The model is fine-tuned on 64 NVIDIA H20 GPUs with a per-GPU batch size of 4 for 40,000 steps. We employ asynchronous inference with 8 denoising steps on an NVIDIA RTX 5090 D GPU, yielding a single-pass latency of approximately 300 ms per action chunk. We further adopt the Real-Time Chunking (RTC) method (Black et al., 2025) to overlap denoising computation with action execution. The robot operates at a control frequency of 15 Hz, executing 15 actions (1 second) per chunk with an RTC inference delay of 6 actions, enabling seamless real-time deployment.

C.1 Qualitative Results

We qualitatively demonstrate X-WAM on an earphone packing task. We select this task because it is a challenging long-horizon manipulation scenario that places stringent demands on 3D spatial reasoning: the robot must estimate the precise 6-DoF pose of the earphone case, localize the narrow slot openings whose geometry varies across orientations, and execute insertion trajectories that satisfy sub-centimeter clearance constraints in all three spatial dimensions. Beyond spatial precision, the task further requires robust bimanual coordination across four sequential stages, testing both long-horizon reliability and real-time execution efficiency.

Task design. Since earphone packing is a multi-step long-horizon task, we decompose it into four sequential stages:

1. Grasp the empty earphone case and open the lid.
2. Pick up one earphone and correctly place it into the case.
3. Pick up the other earphone and correctly place it into the case.
4. Close the lid and return the case to the table.

We roll out X-WAM under two groups of settings, shown in Figure 3. The *scalability* settings consecutively pack 1, 2, or 3 earphones in a single episode, testing the model's ability to handle increasing task length. The *generalization* settings pack 1 earphone under three out-of-distribution conditions not seen during training: novel object placements, unseen tablecloth colors, and unseen distractor objects.

Qualitative results. Figure 4 presents a representative rollout sequence of X-WAM on the earphone packing task. For more qualitative results, please visit our project website or supplementary material.

C.2 Quantitative Results

To provide quantitative real-robot evidence, we evaluate X-WAM on three additional bimanual tasks that require distinct manipulation skills rather than variants of one skill, and compare against two baselines under a larger trial budget.



(a) Scalability Settings (b) Generalization Settings

Figure 3: The two groups of earphone packing settings, shown on the AC One dual-arm platform. (a) The scalability settings pack 1, 2, or 3 earphones consecutively in a single episode. (b) The generalization settings pack 1 earphone under novel placements, unseen tablecloth colors, and unseen distractors. The task requires precise bimanual coordination and robust insertion under tight geometric tolerances.

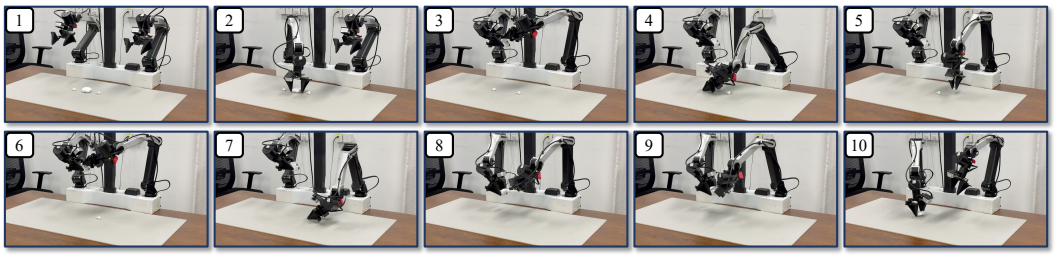


Figure 4: Qualitative results of X-WAM deployed on a real AC One dual-arm robot for the earphone packing task. Each image shows keyframes from a representative execution rollout.

Tasks. The three tasks are chosen to exercise different demands. *Phone packing* is a long-horizon sequential task: the robot must first place the phone into the box, then place the phone accessories, and finally put on the lid, so that each stage depends on the previous one having been completed correctly. *Box packing* is a standard pick-and-place task in which objects are placed into a box, and serves as the least demanding of the three. *Printer refilling* requires interaction with deformable sheets: the robot must separate an appropriate quantity of paper from a stack and load it into the printer tray. Following the protocol of Appendix C.1, each task is decomposed into four sequential stages, each contributing 25% of total progress, so that partial completion is measured rather than binary success. Data collection, fine-tuning, and deployment follow the same setup as the earphone packing task.

Conditions and protocol. Each task is evaluated under three conditions: *in-domain*, an *OOD tablecloth* of a color never seen during training, and *OOD lighting* in which the illumination of the workspace is changed. Both OOD conditions perturb appearance while leaving task semantics and object geometry unchanged, so differences under them isolate visual generalization. We run 20 trials per task-condition-method combination, giving 180 rollouts per method and 540 in total. We compare X-WAM against $\pi_{0.5}$ (Intelligence et al., 2025), a strong VLA on physical robots, and Fast-WAM (Yuan et al., 2026), the world action model with the lowest action latency in Table 6. All three methods use the same RTC (Black et al., 2025) asynchronous execution protocol, control frequency, and action horizon, so no method benefits from a different execution mode. We report average progress over all 20 trials and average completion time over trials reaching 100% progress. Figure 5 shows the workspace of each task under all three conditions, and results are given in Table 5.

Comparison with a specialized VLA. Averaged over the nine task-condition cells, X-WAM reaches 70.7% progress against 69.2% for $\pi_{0.5}$, leading in 5 cells, tied in 3, and behind only on in-domain printer refilling by 2.5 points. It is also faster in every one of the six cells where completion time is defined. X-WAM therefore matches a strong specialized VLA on real-robot task performance while additionally producing the future RGB-D video, an output $\pi_{0.5}$ does not generate at all. We read this as evidence that adding 4D world modeling to a policy need not cost task performance, not as evidence that X-WAM dominates $\pi_{0.5}$: on the nine cells reported here the gap between the two is small relative to what 20 trials per cell can resolve.

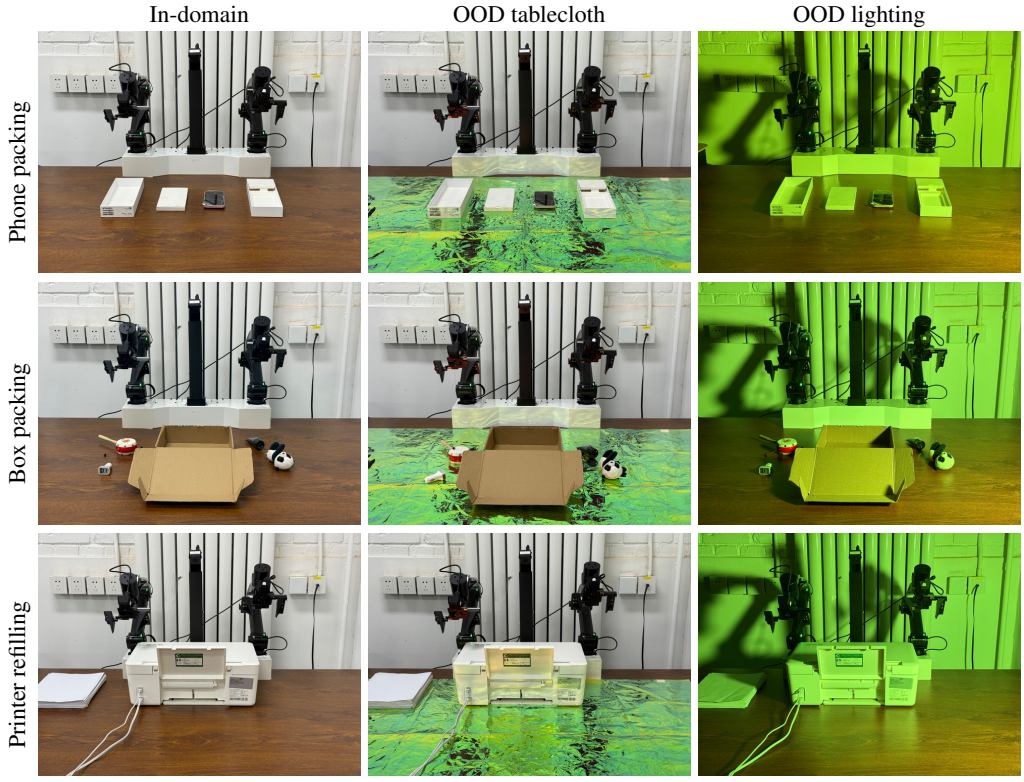


Figure 5: Experimental setup for the three additional real-robot tasks (rows) under the three evaluation conditions (columns). The OOD tablecloth condition replaces the work surface with an unseen color, and the OOD lighting condition changes the workspace illumination. Both perturb appearance while leaving task semantics, object geometry, and the required motion unchanged.

Table 5: Extended real-robot evaluation on three bimanual tasks. Progress (%) is averaged over 20 trials per cell; completion time (s) is averaged over trials reaching 100% progress. All methods run under the same RTC asynchronous execution protocol. Completion time is omitted (–) for printer refilling, where too few trials reach 100% progress for the average to be meaningful. **Bold**: best progress in each row.

Task	Condition	$\pi_{0.5}$ (Intelligence et al., 2025)		Fast-WAM (Yuan et al., 2026)		X-WAM (Ours)	
		Prog. (%)	Time (s)	Prog. (%)	Time (s)	Prog. (%)	Time (s)
Phone packing	In-domain	81.3	62.25	65.0	82.11	**85.0**	**61.73**
	OOD tablecloth	75.0	86.36	62.5	95.52	**77.5**	**85.23**
	OOD lighting	70.0	88.94	50.0	102.13	**75.0**	**85.74**
Box packing	In-domain	**100.0**	40.45	97.5	48.84	**100.0**	**40.23**
	OOD tablecloth	**100.0**	45.74	92.5	54.43	**100.0**	**42.52**
	OOD lighting	98.8	47.23	75.0	58.91	**100.0**	**44.39**
Printer refilling	In-domain	**37.5**	–	12.5	–	35.0	–
	OOD tablecloth	**32.5**	–	10.0	–	**32.5**	–
	OOD lighting	27.5	–	2.5	–	**31.3**	–
Average		69.2		51.9		**70.7**	

Behavior under appearance shift. The two methods separate more clearly out of domain. Going from the in-domain condition to the average of the two OOD conditions, progress drops 4.0 points for X-WAM, 5.6 for $\pi_{0.5}$, and 9.6 for Fast-WAM. The gap also widens with the severity of the shift: under OOD lighting, the harder of the two conditions, X-WAM leads $\pi_{0.5}$ on all three tasks and by up to 5.0 points, whereas in-domain the two are tied on one task and X-WAM trails on another. Since the OOD conditions perturb appearance while holding task semantics and geometry fixed, this difference is attributable to visual generalization, and it is consistent with our claim that predicting future depth supplies geometric structure that is more stable than pixel-level appearance under visual shift. We

note that this is an observational rather than a controlled attribution: the three methods differ in more than their spatial supervision, so the ablation in Table 3(a), which removes only the depth branch, remains the more direct evidence for the role of depth.

Behavior of the lowest-latency world action model. Fast-WAM trails both methods in every cell and nearly collapses on printer refilling, reaching 12.5% progress in-domain and 2.5% under OOD lighting. Two properties of its design plausibly account for this rather than our setup. First, its attention mask makes video and action fully independent, so the action branch receives no predictive signal from the generated future and the joint modeling contributes nothing to control. Second, it is fine-tuned without large-scale robotic pretraining. Read together with Table 6, where Fast-WAM has the lowest action latency of any world action model, this indicates that minimizing action latency by severing the video-action dependency is not the right objective: the method that is fastest to emit an action is also the weakest on every physical task we evaluate.

Deformable-object manipulation remains difficult. All three methods perform poorly on printer refilling, with no method exceeding 37.5% progress in any condition and too few trials reaching completion for completion time to be meaningful. Separating an appropriate quantity of paper from a stack requires fine-grained contact with deformable sheets, whose response to the gripper is governed by friction and compliance rather than by the geometry of the visible surface. This is a limitation of our current model rather than a property of the comparison, and we report it as such: explicit future-depth prediction supplies geometric structure for rigid objects, but does not by itself confer a usable model of deformable dynamics.

D End-to-End Latency Analysis

Measurement protocol. We report the complete observation-to-action closed-loop latency: timing starts when input images are issued and ends when the executable action chunk reaches the controller, including preprocessing, the T_a denoising passes, and action decoding. During online execution the depth branch is disabled and inference terminates once the action is available, so the remaining video steps are never launched and no background GPU work competes with the next action inference. To isolate video-action dependency design from backbone capacity, all world action models in Table 6 use the Wan2.2-TI2V-5B (Wan et al., 2025) backbone, identical input resolution, 40 video and 8 action denoising steps, and one RTX 5090 D, reported as the mean of 50 runs. We disable the multi-frame history of DreamZero and LingBot-VA so the measurement reflects modality coupling rather than context length, and include $\pi_{0.5}$ (Intelligence et al., 2025) as a feed-forward policy lower bound rather than a matched-backbone baseline. The latency column of Table 3 uses an RTX 3090 with 5 action and 25 video steps and is not comparable to this table.

Table 6: End-to-end latency comparison under a matched protocol: Wan2.2-TI2V-5B backbone, identical input resolution, 40 video and 8 action denoising steps, single NVIDIA RTX 5090 D, mean of 50 runs. *Action ready* is the observation-to-action closed-loop latency; *Video ready* is the time until the fully denoised video is available. $\pi_{0.5}$ is a specialized feed-forward policy included as a lower-bound reference, not a matched-backbone baseline.

Method	Action ready (ms)↓	Video ready (ms)↓	Video-action dependency
$\pi_{0.5}$ (Intelligence et al., 2025)	96	n/a	No video generation
Fast-WAM (Yuan et al., 2026)	**112**	**1554**	Video and action generated independently
DreamZero (Ye et al., 2026b)	1610	1610	Video and action share one noise level
LingBot-VA (Li et al., 2026b)	1696	1581	Video generated before action
Motus (Bi et al., 2025)	380	1681	Joint denoising, independent noise levels
X-WAM (Ours)	<u>304</u>	<u>1606</u>	Joint denoising, action first via ANS

Analysis. Action latency orders as Fast-WAM $<$ X-WAM $<$ Motus $<$ DreamZero $<$ LingBot-VA. X-WAM is 20.0% faster than Motus, the closest jointly denoising method, and over 81% faster than DreamZero and LingBot-VA, whose designs tie the action to a fully denoised video by sharing one noise level or by generating video first. Video latency stays within 3.4% of the fastest method. Relative to the feed-forward $\pi_{0.5}$, X-WAM adds 208 ms, which buys a complete future RGB-D video that $\pi_{0.5}$ does not produce.

Fast-WAM is fastest, at 112 ms, because its attention mask makes video and action fully independent, reducing the action path to a small policy head. This is a trade-off rather than a strict improvement: its own paper reports that removing the dependency costs policy performance, and on physical robots it trails both X-WAM and $\pi_{0.5}$ in every cell of Appendix C.2, collapsing almost entirely on printer refilling. Latency is therefore only interpretable jointly with policy performance, and our target is the lowest latency compatible with the joint modeling that policy quality depends on.

E Limitations and Future Work

While X-WAM demonstrates strong performance across simulation benchmarks and real-robot deployment, two primary limitations remain.

First, the current framework processes only a fixed-length context window of observations without incorporating historical information or autoregressive rollout, unlike approaches such as DreamZero (Ye et al., 2026b) that leverage KV caching for extended temporal context. This limited context horizon may hinder the model's ability to fully comprehend task progress in long-horizon manipulation scenarios, potentially leading to suboptimal decisions when the current observation alone is insufficient to disambiguate the task stage.

Second, as a unified model that jointly generates high-dimensional videos and low-dimensional actions, X-WAM incurs higher inference latency compared to dedicated policy models. Specialized VLAs and lightweight WAMs such as Fast-WAM (Yuan et al., 2026) achieve substantially lower per-step latency, whereas X-WAM requires approximately 300 ms per action chunk with 8 denoising steps. Although real-time chunking (Black et al., 2025) enables seamless deployment on physical robots by overlapping computation with execution, the additional inference delay can degrade policy performance, as the robot must act on predictions computed several frames in the past.

Both limitations point to promising directions for future work. Our proposed architecture and noise scheduling strategy are orthogonal to long-context mechanisms, and X-WAM can be readily extended with history conditioning, KV caching, or autoregressive inference to support longer temporal horizons. Similarly, advances in inference acceleration, such as model distillation, consistency models, and more aggressive asynchronous scheduling, could further narrow the latency gap with dedicated policy models while preserving the benefits of unified 4D modeling.