



PRISMWAM: DECOUPLING STATE REPRESENTATIONS IN WORLD ACTION MODELS

Tianyu Xin^{1,*}, Wanqi Zhong^{1,*}, Bowen Hu¹, Zhangchi Guo¹, Lijie Wang¹,
Yiming Wang¹, Haoran Liao¹, Heyang Yu¹, Xiaoyu Tian¹, Zhiding Yu², Yiming Li^{1,†}

¹Tsinghua University ²NVIDIA

*Equal contribution [†]Corresponding author

Project Page: <https://prism-wam.github.io/>

ABSTRACT

World Action Models (WAMs) typically represent the world through latent embeddings from pretrained video models, which entangle heterogeneous aspects of the world, from *how it looks* (e.g., texture and color) to *how it is physically configured* (e.g., occupancy and pose). This raises an underexplored question: ***what should constitute the state of a WAM?*** We argue that the world state should separate *how the world appears* from *how the world evolves under action*: the former captures *semantic context*, while the latter is governed by *physical state*. To this end, we propose **PrismWAM**, which decouples *physics* from *semantics* while jointly modeling state and action. PrismWAM factorizes an embodied task into a Semantic Planning $\rightarrow$ Physical Rollout $\rightarrow$ Executable Action pipeline: given only the initial RGB observation, semantic planning learns to predict the final goal image, while physical rollout predicts the fine-grained evolution toward this goal in depth space; executable actions are then generated from the depth rollout. Without any embodied action pretraining or additional data augmentation, PrismWAM achieves strong manipulation performance in both simulation and the real world. Notably, PrismWAM even surpasses a broad range of models with large-scale embodied action pretraining (e.g., OpenWAM- α). More broadly, we reframe World Action Models from modeling *how the world looks* to *how it responds to action*, paving the way for action-centric world models grounded in occupancy, flow, tactile sensing, proprioception, and beyond.

1 INTRODUCTION

World models have emerged as a promising paradigm for enabling autonomous agents to predict the consequences of their actions (Ha & Schmidhuber, 2018; LeCun, 2022). By learning how the world evolves over time, these models aim to provide a foundation for planning and action (Hafner et al., 2019). Recent World Action Models (WAMs) extend this paradigm by jointly modeling video and action, often relying on pretrained video backbones to capture the rich visual dynamics of the physical world (Ye et al., 2026b; Li et al., 2026b; Ye et al., 2026a). Yet, a key question remains underexplored: ***are video latents truly the most appropriate representations for world states?***

Our key observation is that video latents capture a mixture of factors that are not equally relevant to action generation (NVIDIA et al., 2025a). While appearance and texture provide rich semantic context for understanding the scene, they are often invariant to the actions being modeled. For example, when an agent moves or grasps an object, its geometry and pose may change, while its color, texture, and intrinsic appearance typically remain unchanged. Therefore, video latents which entangles the **physical state** with the **semantic context** can make the representation substantially richer than what is necessary for modeling how the world evolves under action.

This motivates a simple principle: a good state representation in WAMs should distinguish between physical state and semantic context. The physical state refers to action-relevant properties such as geometry, pose, occupancy, contact, and potentially dynamics, whereas semantic context provides

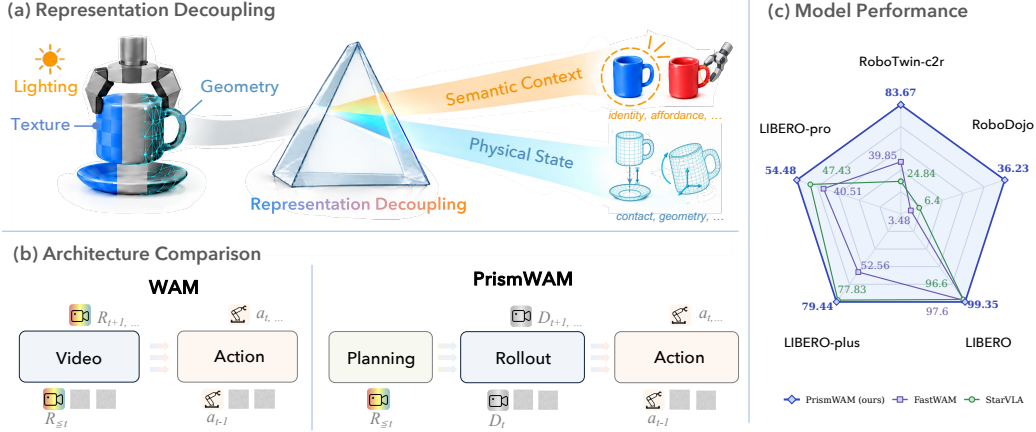


Figure 1: **Overview of PrismWAM.** (a) Our key insight is to decouple physical state from semantic context, yielding state representations that are better aligned with action-induced dynamics. (b) We follow a causal workflow of *what to do* $\rightarrow$ *how the world responds* $\rightarrow$ *how to act*, corresponding to *Semantic Planning* $\rightarrow$ *Physical Rollout* $\rightarrow$ *Executable Action*. (c) Comparison with starVLA and FastWAM across five benchmarks.

the appearance information necessary to interpret that state, such as object identity, category, and texture; see Fig. 1 (a). The distinction is not between information that is useful and useless; rather, it decouples *what actions can change* from *what helps us understand what is being acted upon*.

Therefore, we propose **PrismWAM** to separate semantic context from physical dynamics through a Semantic Planning $\rightarrow$ Physical Rollout $\rightarrow$ Executable Action workflow (see Fig. 1 (b)), rather than treating a single video latent as a monolithic state. Semantic planning infers *what should happen in the end* from RGB observations, physical rollout models *how the world physically evolves* toward the goal, and executable action determines *how to make it happen*. We instantiate this principle with a Mixture-of-Transformers (MoT) architecture built upon a pretrained video backbone (Team Wan et al., 2025), as shown in Fig. 2. During training, the semantic planner predicts the target from the initial RGB observation and transfers its task-level knowledge to the physical rollout via attention layers, while the action branch learns to map the predicted physical evolution to continuous actions. We use depth as the physical representation for its accessibility, physical grounding, and spatial alignment with RGB. Notably, the semantic planner serves only as a training-time teacher, with action inference relying solely on the initial RGB observation and learned depth rollout.

Extensive experiments across diverse simulation and real-world benchmarks validate the effectiveness and efficiency of PrismWAM. **(1)** Without action pretraining, PrismWAM substantially outperforms representative VLAs and WAMs under the same setting, improving the success rate over FastWAM by **43.82** percentage points (83.67 vs. 39.85) on RoboTwin-C2R and the score over starVLA by **29.83** points (36.23 vs. 6.40) on RoboDojo; see Fig. 1(c). **(2)** PrismWAM outperforms several WAMs, including models with action pretraining, with particularly strong gains in precision manipulation and generalization: on RoboDojo Precision, it surpasses OpenWAM- α by **21.37** points (39.82 vs. 18.45), while improving the success rate on RoboTwin-C2R by **14.62** percentage points (83.67 vs. 69.05), showing a higher ceiling for complex tasks and novel scenes. **(3)** PrismWAM even matches or surpasses large-scale pretrained models on several benchmarks despite being trained with only $\sim 2,300$ hours of depth data. On RoboDojo, it improves the score by **16.00** points over G0.5 (36.23 vs. 20.23) and **16.16** points over Xiaomi-Robotics-1 (36.23 vs. 20.07), showing that scaling action pretraining is not the only path to stronger embodied foundation models; principled architectural design offers a more data-efficient alternative. Our key contributions are listed as follows:

- **A decoupling principle for World Action Models.** We identify the entanglement of semantics and physics in video latents as a fundamental limitation of existing WAMs, forcing action generation to disentangle high-level task planning from fine-grained physical state transitions.

- **An effective and efficient WAM architecture.** We instantiate the principle with a Mixture-of-Transformers (MoT), where an RGB semantic planner guides depth-based physical rollout, and develop a learning pipeline combining dense depth supervision with sparse RGB goal supervision, enabling fast inference without explicit RGB goal prediction.
- **Strong performance and data efficiency.** PrismWAM outperforms existing VLAs and WAMs, with particularly strong gains in precision manipulation and generalization. It also surpasses a number of models with large-scale action pretraining, demonstrating that principled architectural design offers a more data-efficient path toward stronger embodied foundation models.

2 RELATED WORKS

Vision-Language-Action Policies. The general and embodied knowledge encoded in pretrained vision-language models (VLMs) (Comanici et al., 2025; Bai et al., 2025; Li et al., 2026a; Wang et al., 2026b; Xiaomi Embodied Intelligence Team, 2025) provides a foundation for vision-language-action (VLA) models that map visual observations and language instructions to robot actions (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Li et al., 2024; Physical Intelligence et al., 2025; NVIDIA et al., 2025b; Shukor et al., 2025). Internet-scale image-text pretraining provides semantic priors for complex instruction understanding and long-horizon manipulation (Brohan et al., 2023; Black et al., 2024; Physical Intelligence et al., 2025; NVIDIA et al., 2025b; Shi et al., 2025). However, such pretraining emphasizes recognition and semantic alignment, with limited explicit supervision for continuous state transitions and how actions change the environment.

World Action Models. Advances in video generative models (Team Wan et al., 2025; NVIDIA et al., 2025a; NVIDIA, 2026) have paved the way for world action models (WAMs), a promising paradigm that jointly models future states and robot actions for generalist robotic manipulation. Most jointly model future RGB observations and action sequences (Zhu et al., 2025; Kim et al., 2026). Some methods guide actions based on predicted future RGB observations (Shen et al., 2025; Bi et al., 2025; Li et al., 2026b; Ye et al., 2026b; Xu et al., 2026; Zhang et al., 2026b; Wang et al., 2026a). Others use future prediction as training supervision and generate actions directly from observed context (Yuan et al., 2026; Ye et al., 2026a; Fang et al., 2026; Gu et al., 2026; Zhang et al., 2026a). Both approaches primarily learn state evolution in RGB appearance space, which entangles action-relevant geometry and pose with texture, illumination, and background. This requires modeling appearance details that may be irrelevant to control and inferring physical changes from visual variations that may not reflect motion (Qiu et al., 2026; Zhang et al., 2026a).

Physical Representations for Manipulation. Physical or spatial representations support precise manipulation and spatial generalization (Shridhar et al., 2022; Goyal et al., 2023; Ze et al., 2024; Ke et al., 2024; Qu et al., 2025). Such information can directly guide action generation through depth or point-cloud features extracted from current observations (Yuan et al., 2025; Li et al., 2025a; Ke et al., 2024). Complementary approaches introduce physical or spatial structure through supervision, including representation alignment (Li et al., 2025b; Wu et al., 2026; Shi et al., 2026; Yang et al., 2026) and auxiliary prediction (Guo et al., 2026; Li et al., 2026c; Zhang et al., 2026a; 2025).

Recent world action models take RGB and geometry as joint inputs and predict their future states together (Zhao et al., 2026; Lopetegui Gonzalez et al., 2026). This couples semantic and physical information, retaining the RGB-based modeling issues discussed above. PrismWAM separates their roles along a Semantic Planning $\rightarrow$ Physical Rollout $\rightarrow$ Executable Action chain, distinguishing semantic task understanding from the physical state evolution that guides action execution.

3 METHOD

PrismWAM follows a Semantic Planning $\rightarrow$ Physical Rollout $\rightarrow$ Executable Action chain for prediction and control. The RGB stream learns semantic planning representations from a sparse future-frame target, while the depth stream predicts a dense physical rollout that guides action generation. We first present the problem formulation (§3.1), then describe PrismWAM’s architecture, training, and inference (§3.2).

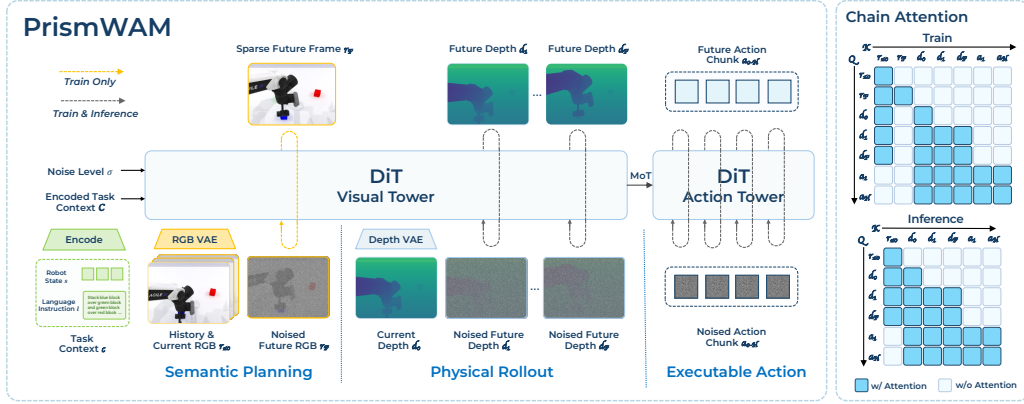


Figure 2: **PrismWAM decouples semantic planning from physical rollout.** **Left:** the RGB stream learns task semantics through sparse supervision from a single intended-future frame, while the depth stream predicts a dense geometric trajectory jointly with actions. **Right:** chain attention routes observed-RGB features to the depth stream and physical rollout features to the action tower.

3.1 PROBLEM FORMULATION

We formulate a world-action model (WAM) that jointly predicts world-state evolution and the corresponding actions. Let x_0 denote the current world state, $x_{1:F}$ its future world state trajectory, and $a_{1:H}$ the corresponding action sequence. Given task context $c = (\ell, s)$, comprising the language instruction ℓ and proprioceptive state s , the policy can be expressed as

$$\pi_\theta(a_{1:H} | x_0, c) = \int \underbrace{p_\theta(a_{1:H} | x_{1:F}, x_0, c)}_{\text{action generation}} \underbrace{p_\theta(x_{1:F} | x_0, c)}_{\text{world-state evolution}} dx_{1:F}. \quad (1)$$

Here, $p_\theta(x_{1:F} | x_0, c)$ models how the current state may evolve over the prediction horizon, while $p_\theta(a_{1:H} | x_{1:F}, x_0, c)$ predicts actions conditioned on the current state and a future state trajectory. Marginalizing over possible future world state trajectories yields the policy’s action distribution. The future world state trajectory $x_{1:F}$ and action sequence $a_{1:H}$ cover the same time interval.

Semantic Planning and Physical Rollout. We use historical and current RGB observations $r_{\leq 0}$ for semantic planning and the current depth observation d_0 as the initial physical state. Semantic planning captures task intent and the intended scene configuration, while physical rollout models the intermediate geometric changes needed for action execution. Guided by RGB representations, the depth stream predicts the sequence $d_{1:F}$, and the action tower jointly generates executable actions from the evolving rollout features.

Let z_R denote semantic planning representations extracted from $r_{\leq 0}$ and z_D the physical rollout features produced during depth prediction. The resulting policy is

$$\pi_\theta(a_{1:H} | r_{\leq 0}, d_0, c) = \int \int \underbrace{p_\theta(a_{1:H} | z_D, c)}_{\text{Executable Action}} \underbrace{p_\theta(z_D | d_0, z_R, c)}_{\text{Physical Rollout}} \underbrace{p_\theta(z_R | r_{\leq 0}, c)}_{\text{Semantic Planning}} dz_R dz_D. \quad (2)$$

3.2 PRISMWAM

Architecture with Sparse-Dense Temporal Supervision. Building on the pretrained video backbone WAN (Team Wan et al., 2025), PrismWAM uses RGB and depth streams with asymmetric temporal supervision, coupled to an action tower through a mixture-of-transformers (MoT) architecture (Fig. 2, left). The RGB stream predicts a single intended-future frame r_F , a sparse target for learning task semantics. The depth stream predicts the denser sequence $d_{1:F}$, providing intermediate spatial states that guide action evolution. This assigns semantic planning to sparse RGB prediction and physical rollout to dense geometric prediction, whose representation is less entangled with texture and illumination.

The model receives RGB history $r_{\leq 0}$, current depth d_0 , instruction ℓ , and proprioception s . Camera views are resized into matching RGB and depth mosaics, and RGB values and clipped metric depth are normalized. The frozen RGB and depth VAEs encode observations and future training targets into latent representations, which modality-specific patch projections convert into visual tokens. Frozen T5 embeddings and projected, normalized proprioception form the context C , which all branches access through cross-attention. Appendix A.1 specifies input layouts and temporal sampling.

During joint training, the RGB token sequence combines clean history latents with noisy latents of the single future target, while the depth sequence combines clean current-depth latents with noisy future-depth latents. The action branch linearly projects each action vector in the noisy sequence into an action token. Gaussian corruption is applied only to prediction targets; observed RGB and depth remain fixed. Each branch receives its own noise-level embedding.

Chain Attention. A structured attention mask controls information flow between these token groups (Fig. 2, right). Within each visual stream, future tokens attend to observed tokens and one another, while observed tokens cannot read future tokens. Depth tokens additionally read only the observed-RGB tokens. Action tokens attend to one another and to both observed and future depth tokens, with no direct access to RGB. These cross-modal connections are directed: RGB does not read depth or actions, and depth does not read actions. Thus, action prediction uses evolving depth features without accessing the future-RGB target.

Specifically, let $h_m^{(j)}$, $m \in \{R, D, A\}$, denote the RGB, depth, and action features after j layers, with $h_m^{(0)}$ their input embeddings and $h_{R,\leq 0}^{(j)}$ the observed-RGB subset. Both visual streams and the action tower receive the encoded task context C through cross-attention. With modality-specific noise levels $\sigma_R, \sigma_D, \sigma_A$, the three feature sequences update synchronously as

$$\begin{aligned} h_R^{(j+1)} &= \mathcal{B}_V^{(j)}(h_R^{(j)} \mid C, \sigma_R), \\ h_D^{(j+1)} &= \mathcal{B}_V^{(j)}(h_D^{(j)} \mid h_{R,\leq 0}^{(j)}; C, \sigma_D), \\ h_A^{(j+1)} &= \mathcal{B}_A^{(j)}(h_A^{(j)} \mid h_D^{(j)}; C, \sigma_D, \sigma_A). \end{aligned} \quad (3)$$

Here $\mathcal{B}_V^{(j)}$ is the visual block shared by the RGB and depth streams, and $\mathcal{B}_A^{(j)}$ is the independently parameterized action block. The visual streams retain modality-specific input, conditioning, and output projections. After the final layer, visual heads project tokens back to latent grids to predict RGB and depth flow velocities, while the action head predicts a velocity vector for each step in normalized action space. These outputs are supervised by the joint flow-matching objective below.

Training Details. We initialize the depth VAE from the pretrained Wan VAE and pretrain it on 2,300 hours of depth data (see Appendix A.2 for the data mixture, losses, and optimization settings); both VAEs and T5 remain frozen during policy training. Policy training follows two stages. First, we adapt the pretrained Wan backbone to depth space by predicting depth sequences conditioned on current depth and language, using depth-prediction loss alone. We then jointly optimize the RGB and depth streams, action tower, and conditioning projections with sparse RGB goal supervision, dense intermediate depth supervision, and action supervision, connecting semantic planning, physical rollout, and executable action generation.

For joint training, let $R_{\leq 0}, R_F$ and $D_0, D_{1:F}$ denote the encoded RGB and depth observations and targets, and $A_{1:H}$ the normalized action chunk; visual subscripts identify the source frames rather than the number of temporally compressed latent frames. For targets $(X_R, X_D, X_A) = (R_F, D_{1:F}, A_{1:H})$, we independently sample Gaussian noise ϵ_m and noise levels $\sigma_m \in [0, 1]$ across modalities:

$$\tilde{X}_m = (1 - \sigma_m)X_m + \sigma_m\epsilon_m, \quad v_m = \epsilon_m - X_m, \quad \epsilon_m \sim \mathcal{N}(0, I). \quad (4)$$

Conditioned on $R_{\leq 0}, D_0, C$, the model predicts velocities $\hat{v}_m$ and minimizes the joint flow-matching objective (Lipman et al., 2023):

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_D + \mathcal{L}_A, \quad \mathcal{L}_m = \mathbb{E}[w(\sigma_m) \text{MSE}(\hat{v}_m, v_m)]. \quad (5)$$

Here w weights noise levels, and the expectation is over training examples, noise, and noise levels. MSE is averaged over valid prediction targets, excluding observations and padding. The noise

schedule and weighting function are specified in Appendix A.1; benchmark-specific budgets, data mixtures, and optimizer settings are collected in table 5.

Inference. At inference (Figure 2, lower right), the RGB stream encodes $r_{\leq 0}$ once and caches its features for semantic guidance, without generating future RGB frames. Future depth latents and the normalized action chunk are initialized from Gaussian noise and jointly denoised, with the observed RGB, current depth, and task context held fixed. The resulting depth latents can be decoded into a depth sequence for visualization, but action generation uses intermediate rollout features and does not require decoding depth maps. We denormalize the predicted action chunk, execute a prefix, and replan from the next observation.

4 EXPERIMENT

Without embodied action pretraining, we evaluate PrismWAM across five simulation benchmarks spanning both in-distribution manipulation and out-of-distribution transfer (§4.2). We further examine how our role-decoupled representations support control through robustness analyses, controlled ablations, and discussion (§4.3), and validate the approach in real-world manipulation (§4.4). Training settings and detailed benchmark results are provided in Appendices A.1 and B, respectively.

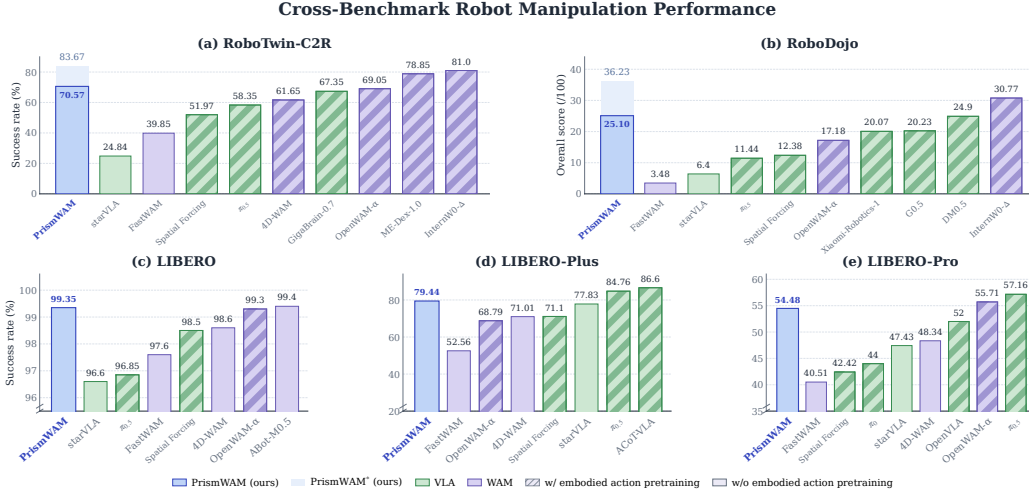


Figure 3: **Cross-benchmark performance of PrismWAM and PrismWAM***. PrismWAM* denotes our exploratory scaled-up variant. Across five manipulation benchmarks, our methods outperform the cross-benchmark baselines starVLA and FastWAM and remain competitive with leading benchmark-specific methods and models using large-scale embodied action pretraining.

4.1 EXPERIMENT SETUP

Benchmarks. We evaluate in-distribution manipulation and out-of-distribution transfer across five simulation benchmarks: RoboTwin clean-to-random (Chen et al., 2025), LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), LIBERO-Pro (Zhou et al., 2025), and RoboDojo (Chen et al., 2026b). LIBERO assesses in-distribution manipulation under its standard protocol. RoboTwin clean-to-random, LIBERO-Plus, and LIBERO-Pro probe transfer across diverse visual, spatial, semantic, task, and environmental shifts, while RoboDojo evaluates manipulation capabilities using both task scores and success rates.

Baselines. We compare PrismWAM with representative cross-benchmark VLAs and WAMs, including methods trained without embodied action pretraining (starVLA (StarVLA Community, 2026), FastWAM (Yuan et al., 2026)), methods that use embodied action pretraining ($\pi_{0.5}$ (Physical Intelligence et al., 2025), OpenWAM- α (Wang et al., 2026a)), and methods that incorporate spatial information during pretraining or fine-tuning (X-WAM (Guo et al., 2026), Spatial-Forcing (Li et al.,

Table 1: **RoboTwin-C2R and RoboDojo performance.** RoboTwin-C2R: success (%); RoboDojo: Score (/100) / SR (%). PrismWAM* denotes our exploratory scaled-up variant. RoboDojo results for both PrismWAM variants are means over seeds 0–2. **Bold**: best; underline: second best.

Method	RoboTwin-C2R			RoboDojo					
	Clean	Rand.	Avg.	Gen.	Precision	Long-H.	Memory	Open	Overall
<i>w/ Embodied Action Pretraining</i>									
$\pi_{0.5}$	70.70	46.00	58.35	13.38/8.17	12.40/5.50	23.54/14.67	5.89/4.67	<u>1.98/1.67</u>	11.44/6.93
OpenWAM- α	<u>89.40</u>	48.70	69.05	20.71/14.83	18.45/9.25	34.93/ <u>25.33</u>	10.41/9.11	1.41/1.08	17.18/11.92
<i>w/o Embodied Action Pretraining</i>									
starVLA	46.52	3.16	24.84	3.94/2.34	9.90/4.33	14.15/6.50	3.34/2.44	0.67/0.58	6.40/3.24
FastWAM	77.80	1.90	39.85	2.34/1.11	1.96/0.00	9.14/5.17	3.55/3.44	0.42/0.42	3.48/2.03
PrismWAM (ours)	87.98	<u>53.16</u>	<u>70.57</u>	<u>27.01/19.56</u>	<u>28.70/19.50</u>	<u>37.39/22.08</u>	<u>30.46/30.11</u>	1.92/1.50	<u>25.10/18.55</u>
PrismWAM* (ours)	91.84	75.50	83.67	36.71/29.33	39.82/31.08	48.67/31.00	50.96/50.67	4.98/4.42	36.23/29.30
<i>w/ Spatial Information</i>									
X-WAM	70.00	25.80	47.90	7.39/3.33	6.72/1.83	17.47/9.08	6.32/4.67	0.57/0.25	7.69/3.83
Spatial-Forcing	77.20	26.74	51.97	14.12/9.34	17.32/10.58	23.26/14.58	5.43/4.11	1.78/1.58	12.38/8.04
4D-WAM	81.50	41.80	61.65	13.35/8.83	18.08/10.00	23.30/14.00	7.38/6.00	1.44/1.00	12.71/7.97

Table 2: **LIBERO performance and transfer.** Success rates (%).

Method	LIBERO					LIBERO-Plus								LIBERO-Pro					
	Spatial	Object	Goal	Long	Overall	Camera	Robot	Lang.	Light	Bg.	Noise	Layout	Overall	Object	Pos.	Sem.	Task	Env.	Overall
w/ Embodied Action Pretraining																			
$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.85	<u>71.23</u>	76.06	85.82	<u>97.72</u>	<u>96.19</u>	87.32	86.23	84.76	<u>85.40</u>	95.50	25.30	<u>51.45</u>		57.16
OpenWAM- α	99.6	99.6	99.8	<u>98.2</u>	<u>99.3</u>	33.21	<u>75.87</u>	87.57	97.02	85.59	39.85	77.38	68.79	84.45	<u>26.10</u>	95.95	<u>25.25</u>	46.80	<u>55.71</u>
w/o Embodied Action Pretraining																			
starVLA	97.8	98.6	96.2	93.8	96.6	64.04	57.16	85.75	93.78	96.00	<u>80.95</u>	77.25	77.83	76.65	2.80	94.85	17.80	45.05	47.43
FastWAM	98.2	100.0	97.0	95.2	97.6	16.01	46.26	72.93	81.96	55.48	39.41	66.49	52.56	73.00	0.05	82.90	14.90	31.70	40.51
PrismWAM (ours)	<u>99.4</u>	100.0	98.2	99.8	99.35	60.60	67.61	92.13	99.65	99.54	<u>97.46</u>	<u>81.70</u>	<u>79.44</u>	88.65	6.55	98.55	23.45	55.20	54.48
w/ Spatial Information																			
X-WAM	96.8	98.6	95.4	95.6	96.60	72.92	46.32	88.48	93.61	88.29	62.90	80.79	74.80	83.95	21.20	<u>97.25</u>	17.70	45.55	53.13
Spatial-Forcing	<u>99.4</u>	<u>99.6</u>	<u>98.8</u>	<u>96.0</u>	98.5	56.85	37.94	84.39	95.36	91.82	65.90	79.02	71.10	73.95	2.05	94.45	9.80	31.85	42.42
4D-WAM	98.8	99.5	98.6	97.4	98.6	45.15	64.26	<u>90.63</u>	94.29	57.71	69.08	79.21	71.01	80.75	3.90	95.85	21.10	40.10	48.34

2025b), 4D-WAM (Yang et al., 2026)). Hatched bars in fig. 3 denote pretrained models and methods fine-tuned from them. We additionally include top-performing benchmark-specific methods (Giga-Brain Team et al., 2026; Chen et al., 2026a; Zhong et al., 2026; Kim et al., 2024; Black et al., 2024; Liu et al., 2026; Xiaomi Robotics Team et al., 2026; Dexmal Team, 2026). For RoboTwin-C2R, fig. 3 also includes ME-Dex-1.0 (Zhang et al., 2026c), a leading entry on the official leaderboard as of October 19, 2026 (RoboTwin Team, 2026). ME-Dex-1.0 uses tactile supervision during training and zero tactile input at evaluation. We also include InternW0- Δ (Miao et al., 2026), with publicly available fine-tuning and inference code and benchmark checkpoints. We use its reported clean/random average for RoboTwin-C2R and the October 17, 2026 leaderboard score for RoboDojo (RoboDojo Team, 2026).

4.2 MAIN RESULTS

Across five benchmarks, PrismWAM *substantially outperforms the cross-benchmark baselines starVLA and FastWAM without embodied action pretraining*. On RoboTwin-C2R, PrismWAM achieves 53.16% success in randomized scenes, compared with starVLA at 3.16% (+50.00 pp) and FastWAM at 1.90% (+51.26 pp). These results demonstrate the strong *zero-shot generalization* capability of PrismWAM without embodied action pretraining. PrismWAM further outperforms the available geometry-aware baselines on LIBERO, reaching 99.35% versus 98.60% for 4D-WAM (table 2). These results suggest that *the key to effectively leveraging physical information is not only to incorporate it, but also to decouple it from semantic representations*. Moreover, PrismWAM *surpasses both pretrained and non-pretrained WAM baselines* in most settings, achieving 70.57% clean/random average success on RoboTwin-C2R compared with 69.05% for pretrained OpenWAM- α . On RoboDojo, PrismWAM* achieves an overall score of 36.23 (fig. 3). Beyond WAM comparisons, PrismWAM remains competitive with large-scale pretrained policies (fig. 5). Results on RoboDojo also suggest our *data-efficiency*, achieving strong control without additional embodied backbone pretraining.

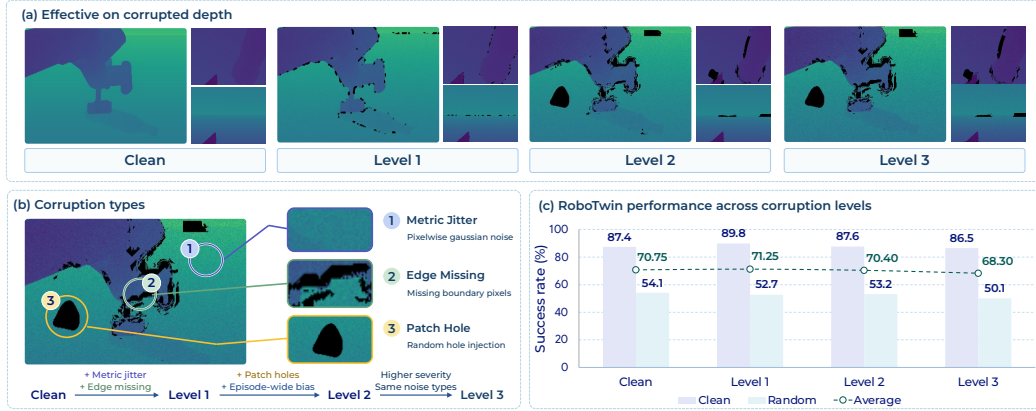


Figure 4: **Robustness to progressive depth corruption.** **Top:** Depth observations across corruption levels. **Bottom left:** Corruption types and their level-wise composition. **Bottom right:** RoboTwin performance under progressively increasing corruption of the original depth observations.

4.3 ANALYSIS AND DISCUSSION

RQ1: Does physical-state prediction offer more than appearance robustness? Depth encodes distance rather than texture or illumination, making it naturally insensitive to changes in color and lighting. This raises a key question: does physical-state prediction offer more than appearance robustness? Results across multiple benchmarks point to broader benefits for generalization and precise control. PrismWAM achieves 81.70% under layout perturbations on LIBERO-Plus, compared with 77.38% for OpenWAM- α and 66.49% for FastWAM; on LIBERO-Pro, it achieves 88.65% when object attributes such as size, shape, and color change, exceeding OpenWAM- α at 84.45% and FastWAM at 73.00% (table 2). On RoboDojo’s challenging Precision tasks, PrismWAM substantially outperforms pretrained OpenWAM- α , achieving 19.50% success versus 9.25% (table 1). These results suggest that physical-state prediction goes beyond appearance robustness: *it supports broader generalization, as illustrated by robustness to changes in scene layout and object shape, and exposes physical structure for precise control.*

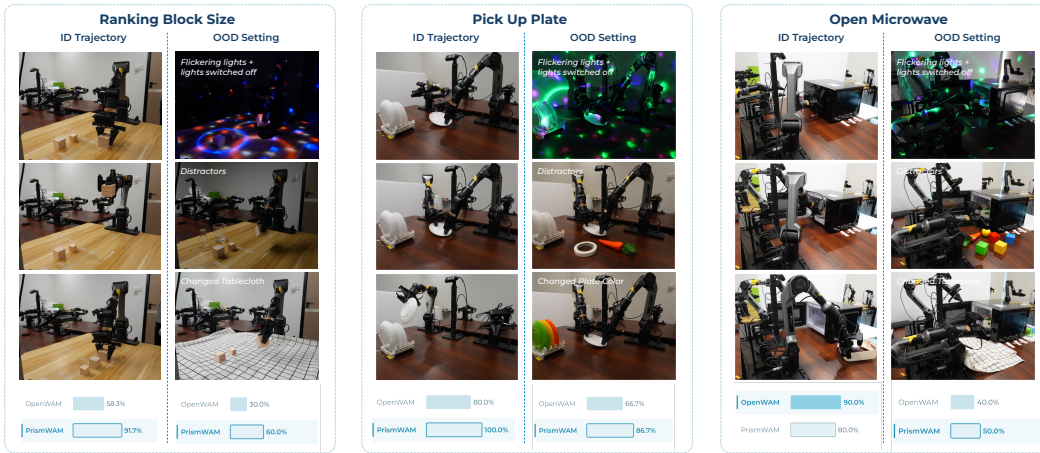
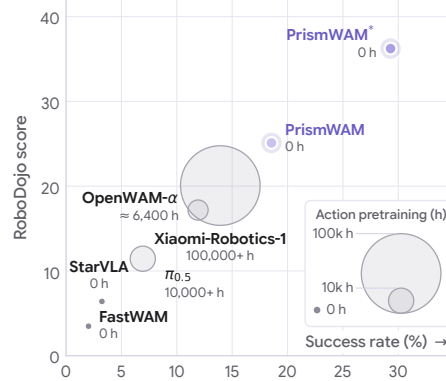
RQ2: Can PrismWAM remain reliable with real-world-like corruption? Real-world depth sensors may produce measurement noise and missing values, motivating evaluation with real-world-like depth corruption. The corruption profiles, which are informed by the manufacturers’ specifications for Orbbec Gemini 335 and RealSense D435, are controlled stress tests rather than calibrated models of either sensor; full settings are provided in section A.4. As shown in fig. 4, we train and evaluate PrismWAM on RoboTwin-C2R across three corruption levels constructed from *Metric Jitter*, *Edge Missing*, and *Patch Hole*. On randomized scenes, all three corruption levels reduce success relative to uncorrupted depth. In the clean setting, light corruption yields a small improvement before performance declines at higher severity. Nevertheless, even at the strongest corruption level, the average performance drops by only 2.45 pp. These results demonstrate that PrismWAM remains robust to diverse forms of depth corruption.

RQ3: Do PrismWAM’s design choices work together effectively? We examine the contributions of *representation composition*, *attention routing*, and *supervision and training* through ablations on RoboTwin-C2R (table 3). Full settings and results for all six ablations are reported in Appendix A.3. Removing either the RGB or depth stream reduces both clean-scene success and transfer, supporting the complementary roles of semantic context and physical state. Direct RGB-to-action access improves clean-scene success but weakens transfer, while exposing depth to future RGB also reduces transfer, favoring the proposed chain from observed semantic context through physical rollout to action. Together, these results support *representation decoupling through complementary streams*, with the full design achieving the best clean-random average.

Table 3: Component ablations on 50 RoboTwin tasks, with 20 episodes per task and condition.

Variant	Clean	Random	Avg.
PrismWAM	87.40	54.10	70.75
<i>Representation composition</i>			
w/o RGB stream	79.10	40.10	59.60
w/o depth stream	83.00	2.40	42.70
<i>Attention routing</i>			
Observed RGB $\rightarrow$ Action	89.10	40.30	64.70
Future RGB $\rightarrow$ Depth	87.00	50.00	68.50
<i>Supervision & training</i>			
w/o future-RGB supervision	81.60	51.40	66.50
w/o depth-only stage	89.30	24.70	57.00

Figure 5: RoboDojo score and success rate.

Figure 6: **Real-world evaluation.** ID trajectories (left), representative OOD scenes (right), and OpenWAM versus PrismWAM success rates (bottom) across three tasks.

4.4 REAL-WORLD VALIDATION

Tasks and evaluation conditions. We compare PrismWAM with OpenWAM (Wang et al., 2026a) on three real-robot tasks: *Pick Up Plate*, *Ranking Block Size*, and *Open Microwave*. These tasks cover bimanual plate handling, single-arm size-dependent ordering, and multistep bimanual retrieval, respectively. In-distribution (ID) trials use the original setups and task-specific initial arrangements. Out-of-distribution (OOD) trials vary illumination, distractors, and object appearance or placement while preserving each task’s goal. Figure 6 and table 4 summarize the settings and results. Task instructions, training recipes, are detailed in Appendix A.5.

PrismWAM achieves pooled success of 88.9% in ID and 68.6% in OOD, compared with 74.1% and 48.6% for OpenWAM, improving success by 14.8 and 20.0 percentage points, respectively. OOD gains hold across all three tasks: 86.7% versus 66.7% on *Pick Up Plate*, 60.0% versus 30.0% on *Ranking Block Size*, and 50.0% versus 40.0% on *Open Microwave*. In ID, PrismWAM improves plate handling and size ordering, while OpenWAM performs better on microwave retrieval (90.0% versus 80.0%). These results extend the scene-change robustness observed in simulation to physical manipulation, supporting the practical value of combining RGB semantic guidance with depth-based physical rollouts.

Table 4: **Real-world success under ID and OOD conditions.** Entries are successes/trials (%). Both models are evaluated on the same three tasks and trial counts. Pooled rates divide total successes by total trials across tasks. Bold marks the higher success rate within each task and condition.

Task	OpenWAM		PrismWAM	
	ID	OOD	ID	OOD
Pick Up Plate	4/5 (80.0)	10/15 (66.7)	5/5 (100.0)	13/15 (86.7)
Ranking Block Size	7/12 (58.3)	3/10 (30.0)	11/12 (91.7)	6/10 (60.0)
Open Microwave	9/10 (90.0)	4/10 (40.0)	8/10 (80.0)	5/10 (50.0)
Pooled (3 tasks)	20/27 (74.1)	17/35 (48.6)	24/27 (88.9)	24/35 (68.6)

5 CONCLUSION

We introduce **PrismWAM**, a World Action Model built on the principle of decoupling semantic context from physical state. Rather than treating video latents as a monolithic world representation, PrismWAM decomposes world-action modeling into Semantic Planning $\rightarrow$ Physical Rollout $\rightarrow$ Executable Action: RGB future-frame prediction captures task-level intent, depth models physical state transitions, and the resulting rollout is mapped directly to continuous actions without RGB goal prediction at inference. Across diverse simulation and real-world manipulation benchmarks, PrismWAM achieves strong performance and generalization with only 2,300 hours of depth data and without embodied action pretraining, while matching or surpassing large-scale action-pretrained models on challenging settings. These results suggest that scaling action data is not the only path to stronger embodied foundation models; principled state representations and architectural design offer a more data-efficient alternative. More broadly, PrismWAM paves the way for a new generation of WAMs that shifts the focus from modeling *how the world looks* to *how the world responds to action*. While depth is one practical realization of this principle, it can extend to occupancy, flow, tactile sensing, proprioception, and other representations capturing action-induced state transitions.

Limitations and future work. At present, large-scale real-world manipulation data with synchronized depth observations remains relatively limited, which constrains the scale of depth pretraining compared with existing action-pretrained models. We view this primarily as a reflection of the current data landscape: as depth sensing becomes more accessible and embodied datasets incorporate richer physical modalities, such data is likely to become increasingly available. Physical-state pretraining may therefore emerge as a natural scaling direction for World Action Models, with PrismWAM taking an early step toward establishing this new paradigm.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting, language polishing, and structural editing, as well as with literature search and code development. AI-generated suggestions were used only as assistance and were reviewed, revised, and verified by the authors. All technical claims, experimental results, interpretations, and final written content were validated by the authors, who take full responsibility for the work.

ACKNOWLEDGMENTS

We thank Chong Kar Hing, Junwei Li, and Zexuan Wen for their assistance with the experiments, and Shihao Wang and Yu-Cheng Chou for their insightful ideas.

REFERENCES

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report, 2025. URL <https://arxiv.org/abs/2511.21631>.
- 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, 2025. URL <https://arxiv.org/abs/2512.13030>.
- Kevin Black et al. π_0 : A vision-language-action flow model for general robot control, 2024. URL <https://arxiv.org/abs/2410.24164>.
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Danny Driess, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL <https://arxiv.org/abs/2307.15818>.
- Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. URL <https://arxiv.org/abs/2503.06669>.
- Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified mobility-and-manipulation world action model, 2026a. URL <https://arxiv.org/abs/2607.00678>.
- Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies, 2026b. URL <https://arxiv.org/abs/2607.04434>.
- Tianxing Chen et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation, 2025. URL <https://arxiv.org/abs/2506.18088>.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL <https://arxiv.org/abs/2507.06261>.
- Dexmal Team. DM0.5: An open-world foundation model for general-purpose embodied intelligence, July 2026. URL https://www.dexmal.com/blog/dm0.5/index_en.html.
- Pengcheng Fang, Hongli Chen, and Xiaohao Cai. Privileged foresight distillation: Zero-cost future correction for world action models, 2026. URL <https://arxiv.org/abs/2604.25859>.
- Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models, 2025. URL <https://arxiv.org/abs/2510.13626>.
- GigaBrain Team et al. GigaBrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture, 2026. URL <https://arxiv.org/abs/2608.15875>.
- Ankit Goyal et al. RVT: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, 2023. URL <https://arxiv.org/abs/2306.14896>.
- Qiwen Gu, Jifan Li, Bingjie Gao, Rui Chen, Jing Tang, Xiangxiang Chu, and Junqiao Zhao. The right future for action: Learning action-relevant predictive states in world action models, 2026. URL <https://arxiv.org/abs/2609.23369>.
- Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D world action modeling from video priors with asynchronous denoising, 2026. URL <https://arxiv.org/abs/2604.26694>.
- David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL <https://arxiv.org/abs/1803.10122>.

- Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.
- Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3D Diffuser Actor: Policy diffusion with 3D scene representations, 2024. URL <https://arxiv.org/abs/2402.10885>.
- Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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, 2024. URL <https://arxiv.org/abs/2406.09246>.
- 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, 2026. URL <https://arxiv.org/abs/2601.16163>.
- Yann LeCun. A path towards autonomous machine intelligence, 2022. URL <https://openreview.net/pdf?id=BZ5alr-kVsfi>. Position paper, version 0.9.2, June 27, 2022.
- Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. PointVLA: Injecting the 3D world into vision-language-action models, 2025a. URL <https://arxiv.org/abs/2503.07511>.
- 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, 2025b. URL <https://arxiv.org/abs/2510.12276>.
- Kehan Li, Bohan Hou, Minghao Zhu, et al. RynnBrain 1.1: Towards more capable and generalizable embodied foundation model, 2026a. URL <https://arxiv.org/abs/2607.17977>.
- 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, 2026b. URL <https://arxiv.org/abs/2601.21998>.
- Qixiu Li et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation, 2024. URL <https://arxiv.org/abs/2411.19650>.
- Ying Li, Xiaobao Wei, Jiajun Cao, Hao Wang, Xiaowei Chi, Chengyu Bai, Qianpu Sun, Jiajun Li, Xiaojie Zhang, Peidong Jia, Jian Tang, Sirui Han, and Shanghang Zhang. WAM4D: Fast 4D world action model via spatial register tokens, 2026c. URL <https://arxiv.org/abs/2606.14048>.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL <https://arxiv.org/abs/2210.02747>.
- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL <https://arxiv.org/abs/2306.03310>.
- Yicheng Liu et al. G0.5: One autoregressive stream for robot reasoning and action, 2026. URL <https://arxiv.org/abs/2608.11739>.
- Javier Alejandro Lopetegui Gonzalez, Paul Pacaud, and Cordelia Schmid. Spatially aware world action model via geometric latent diffusion, 2026. URL <https://arxiv.org/abs/2609.02531>.
- Xingyu Miao, Zizun Li, Baole Fang, Kaiwen Song, Tenghui Wang, Hanxue Zhang, Yating Wang, Xudong Li, Yuping He, Xueyuan Wei, Chao Gao, Xijie Yang, Yingxiang Xu, Kerui Ren, Wenqi Guo, Jianjun Zhou, Xinzhe Wang, Weiguang Zhao, Ni Yang, Zetao Cai, Yufei Xue, Hengjie Li, Zeyu He, Yuanzhen Zhou, Rong Fu, Jianyang Zhang, Siwei Cui, Fuxian Huang, Yunsong Zhou, Xing Gao, Yifei Yao, Qiaojun Yu, Kailin Li, Ming Zhou, Mu Huang, Xinyue Li, Wenze Cui, Bingqi Jiang, Xueyue Zhu, Junting Dong, Haoyu Guo, Tao Lu, Mulin Yu, Bowen Zhou, Bin

- Zhao, Tianfan Xue, Weinan Zhang, and Chunhua Shen. InternW0- Δ : A world action model bridging predictive dynamics and actions with 20k+ hours of open data, 2026. URL <https://arxiv.org/abs/2609.31394>.
- NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026. URL <https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf>.
- NVIDIA et al. World simulation with video foundation models for physical AI, 2025a. URL <https://arxiv.org/abs/2511.00062>.
- NVIDIA et al. GR00T N1: An open foundation model for generalist humanoid robots, 2025b. URL <https://arxiv.org/abs/2503.14734>.
- Physical Intelligence, Kevin Black, Noah Brown, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization, 2025. URL <https://arxiv.org/abs/2504.16054>.
- Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models, 2026. URL <https://arxiv.org/abs/2606.12217>.
- Delin Qu et al. SpatialVLA: Exploring spatial representations for visual-language-action model, 2025. URL <https://arxiv.org/abs/2501.15830>.
- RoboDojo Team. RoboDojo Official Leaderboard, 2026. URL <https://robodojo-benchmark.com/LeaderBoard>. Updated October 17, 2026; accessed October 19, 2026.
- RoboTwin Team. RoboTwin 2.0 Official Leaderboard, 2026. URL <https://robotwin-platform.github.io/leaderboard>. Accessed October 19, 2026; ranked by mean clean and randomized success.
- 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. In *Advances in Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=UPHlqbZFZB>.
- Jiixin Shi, Xidong Zhang, Fucai Zhu, Zhe Li, Siyu Zhu, and Weihao Yuan. 3DThinkVLA: Endowing vision-language-action models with latent 3D priors via 3D-thinking-guided co-training, 2026. URL <https://arxiv.org/abs/2606.04436>.
- Lucy Shi et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL <https://arxiv.org/abs/2502.19417>.
- Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, 2022. URL <https://arxiv.org/abs/2209.05451>.
- Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmoVLA: A vision-language-action model for affordable and efficient robotics, 2025. URL <https://arxiv.org/abs/2506.01844>.
- StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model development, 2026. URL <https://arxiv.org/abs/2604.05014>.
- Team Wan et al. Wan: Open and advanced large-scale video generative models, 2025. URL <https://arxiv.org/abs/2503.20314>.
- Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhao Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao Dong, Lin Shao, and Hang Zhao. OpenWAM: An open, modular exploration towards systematic world-action model pretraining, 2026a. URL <https://arxiv.org/abs/2609.07398>.

- Ziyi Wang, Xumin Yu, Yongming Rao, et al. Hy-Embodied-VLM-1.0: Efficient physical-world agents, 2026b. URL <https://arxiv.org/abs/2607.12894>.
- Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving VLA models in practice, 2026. URL <https://arxiv.org/abs/2607.06403>.
- Xiaomi Embodied Intelligence Team. MiMo-Embodied: X-Embodied foundation model technical report, 2025. URL <https://arxiv.org/abs/2511.16518>.
- Xiaomi Robotics Team et al. Xiaomi-Robotics-1: Scaling vision-language-action models with over 100k hours of real-world trajectories, 2026. URL <https://arxiv.org/abs/2607.15330>.
- Gangwei Xu, Qihang Zhang, Jiaming Zhou, Xing Zhu, Yujun Shen, Xin Yang, and Yinghao Xu. Next Forcing: Causal world modeling with multi-chunk prediction, 2026. URL <https://arxiv.org/abs/2606.11187>.
- Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Pengwei Wang, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing spatiotemporal awareness into world action models through trajectory fields, 2026. URL <https://arxiv.org/abs/2608.08023>.
- 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, 2026a. URL <https://arxiv.org/abs/2603.17240>.
- Seonghyeon Ye et al. World action models are zero-shot policies, 2026b. URL <https://arxiv.org/abs/2602.15922>.
- Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. DepthVLA: Enhancing vision-language-action models with depth-aware spatial reasoning, 2025. URL <https://arxiv.org/abs/2510.13375>.
- 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. URL <https://arxiv.org/abs/2603.16666>.
- Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable visuomotor policy learning via simple 3D representations. In *Robotics: Science and Systems*, 2024. URL <https://arxiv.org/abs/2403.03954>.
- Jianjun Zhang, Jian Zhu, Taiyi Su, Chong Ma, Zitai Huang, Yi Xu, and Hanli Wang. Learning 4D geometric priors for inference-efficient world action models, 2026a. URL <https://arxiv.org/abs/2607.05468>.
- Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native video-action pretraining for generalizable robot control, 2026b. URL <https://arxiv.org/abs/2607.08639>.
- Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge, 2025. URL <https://arxiv.org/abs/2507.04447>.

- Xuancheng Zhang, Xuetao Liu, Qianying Tang, Jizhe Wang, Runsheng Wang, Zhijing Cheng, Bochen Lin, Haoran Wen, Ming Li, Kun Zhan, and Yu Liu. ME-Dex 1.0: Bringing Heterogeneous Tactile Sensing into World Action Modeling, 2026c. URL <https://arxiv.org/abs/2609.21449>.
- Haoyu Zhao, Xingyue Zhao, Siteng Huang, Xin Li, Deli Zhao, and Zhongyu Li. RynnWorld-4D: 4D embodied world models for robotic manipulation, 2026. URL <https://arxiv.org/abs/2607.06559>.
- Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. ACoT-VLA: Action chain-of-thought for vision-language-action models, 2026. URL <https://arxiv.org/abs/2601.11404>.
- Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization, 2025. URL <https://arxiv.org/abs/2510.03827>.
- 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, 2025. URL <https://arxiv.org/abs/2504.02792>.

A TRAINING AND ADDITIONAL EXPERIMENTS

A.1 PRISMWAM TRAINING RECIPE

The main PrismWAM model has approximately 6B parameters and uses Wan2.2-5B (Team Wan et al., 2025); PrismWAM* denotes the exploratory 15B variant, which is initialized from WAN-2.1-14B with additional video-only pretraining. Training comprises depth-only adaptation conditioned on current depth and language, followed by joint RGB/depth/action training; both VAEs and T5 remain frozen.

Table 5: PrismWAM training and deployment settings. C2R: clean-to-random transfer; EE: end-effector; H/K : action horizon/executed prefix.

Setting	RoboTwin C2R	LIBERO	RoboDojo
Training data	Clean	Four suites	ARX X5 sim.
Tasks / total demos	50 / 2,500	40 / 2,000	35 / 3,500
Demos per task	50 clean	50	100
Optimizer updates: depth / joint	15k / 15k	15k / 15k	15k / 40k
Global batch size	256	256	1024
Views; mosaic ($H \times W$)	3; 384×320	2; 224×448	3; 384×320
Depth frames	9	5	9
RGB history frames	5	5	9
Future-RGB target	$t + 32$	$t + 16$	$t + 32$
Action representation	14-D absolute joints	7-D EE deltas	14-D abs. joints
Proprioceptive dimension	14	8	14
Target action indices	$[t + 1, t + 32]$	$[t, t + 15]$	$[t + 1, t + 32]$
Action / state statistics	Shared mean/std	Separate min-max	Shared mean/std
Horizon / replan (H/K)	32 / 24	16 / 16	32 / 24

Optimization and evaluation. AdamW uses $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.01, gradient clipping 1.0, and bf16 without gradient accumulation. Each stage resets the optimizer; all trainable modules share 5% linear warmup and cosine decay from 10^{-4} to 10^{-6} . We use seed 42, uniform task/window sampling, and checkpoints every 1000 updates. Main results use the final checkpoint and no sensor-depth corruption. LIBERO trains on Spatial/Object/Goal/Long; Plus/Pro reuse its checkpoint. Inference uses ten shift-5 Euler steps with cached observed RGB.

Inputs. Three-view mosaics place a 256×320 head image above two 128×160 wrist images; LIBERO places 224×224 external/wrist images left/right. RGB/depth layouts are shared across training and evaluation. RoboTwin/LIBERO encode RGB history as one causal clip (two latent frames) and the future target separately (one latent frame); RoboDojo uses history-noise probability 0.25. Depth clips include the current frame; actions use consecutive steps. Out-of-range frames repeat boundary observations, with padded targets masked.

Flow matching. Independently per predicted modality, sample $u \sim \mathcal{U}(0, 1)$, set $\sigma = 5u/(1 + 4u)$, and use timestep 1000σ . For the targets in eqs. (4) and (5),

$$w(\sigma) = \frac{y(\sigma) - y_{\min}}{\bar{y} - y_{\min} + 10^{-10}}, \quad y(\sigma) = \exp[-2(\sigma - 1/2)^2], \quad (6)$$

where $y_{\min}$ and $\bar{y}$ are the minimum and mean on the 1000-point shifted grid $u \in \{1, 0.999, \dots, 0.001\}$. Modality losses have unit coefficients and exclude conditioning frames and padded targets.

A.2 DEPTH-VAE TRAINING RECIPE

We train Wan2.2-compatible and Wan2.1-compatible single-channel depth VAEs with 48 and 16 latent channels, respectively. Each is initialized from its corresponding pretrained RGB VAE (Team Wan et al., 2025), averaging the RGB input and output boundary weights across channels while retaining matching internal weights. Their spatial compression factors are 16 and 8, respectively. We additionally train a head-view-only depth VAE for real-world setups without wrist-view depth. The depth VAEs remain frozen throughout downstream policy training.

Training data and mixture. We use processed AgiBotWorld-Beta depth (Bu et al., 2025) and simulated RoboTwin depth from clean and randomized demonstrations (Chen et al., 2025). After additional filtering of AgiBotWorld-Beta, the combined corpus contains approximately 2,300 hours of depth sequences. We sample AgiBotWorld-Beta and RoboTwin with probabilities 0.75 and 0.25, respectively, then sample a valid temporal clip with replacement. Training augmentation comprises random temporal cropping, in-plane rotation within $\pm 5^\circ$, and horizontal flipping with probability 0.5, applied consistently across frames and views.

Table 6 lists the shared training settings. The training objective is

$$\mathcal{L}_{\text{VAE}} = \|\hat{d} - d\|_1 + \|\hat{d} - d\|_2^2 + 10^{-6} D_{\text{KL}}(q(z | d) \| \mathcal{N}(0, I)), \quad (7)$$

with mean-reduced reconstruction terms, stochastic posterior sampling, and no additional valid-depth mask. Per-channel latent means and standard deviations are computed from encoder posterior means and stored with each checkpoint for downstream normalization.

Table 6: Shared depth-VAE training settings.

Setting	Value
Clip length / frame stride	9 frames / 4
Temporal compression factor	4
Depth clipping / normalization	0–1500 mm / $[-1, 1]$
Global batch / gradient accumulation	256 / 1
Optimizer	AdamW, $\beta = (0.9, 0.95)$
Weight decay / gradient clipping	0.01 / 1.0
Precision	bf16
Warmup updates	5000
Learning-rate schedule	Linear warmup, then cosine decay
Selected checkpoint	50k updates

A.3 COMPONENT ABLATIONS

We ablate PrismWAM initialized from Wan2.2-5B, using clean-only RoboTwin demonstrations, global batch 256, and 30k updates (15k depth adaptation plus 15k joint training; 30k joint updates without the depth-only stage). Inference uses ten denoising steps and replans every 24 actions; the future-RGB-to-depth variant jointly denoises RGB, depth, and actions. We evaluate final checkpoints on 50 tasks with 20 episodes per task and Clean/Random condition (1000 each), compared with 100 episodes per task and condition in the main PrismWAM results. Evaluation uses unseen instructions, policy seed 0, instruction seed 42, and environment seeds starting at 100000.

Table 7: RoboTwin success (%); Δ : Mean change from the full model (pp).

Variant	Clean	Random	Mean	Δ
PrismWAM (full)	87.40	54.10	70.75	–
<i>Representation composition</i>				
w/o RGB stream	79.10	40.10	59.60	–11.15
w/o depth stream	83.00	2.40	42.70	–28.05
<i>Attention routing</i>				
Observed RGB $\rightarrow$ Action	89.10	40.30	64.70	–6.05
Future RGB $\rightarrow$ Depth	87.00	50.00	68.50	–2.25
<i>Supervision & training</i>				
w/o future-RGB supervision	81.60	51.40	66.50	–4.25
w/o depth-only stage	89.30	24.70	57.00	–13.75

A.4 DEPTH CORRUPTION

Perturbation Setting. We corrupt metric depth at three severity levels (fig. 4 and table 8). *Metric Jitter* adds Gaussian noise whose standard deviation increases quadratically with depth, ranging from $\sigma_0 = 12\text{--}36$ mm at 0 m to $\sigma_1 = 20\text{--}60$ mm at 1 m, with uniform clip-/episode-wise offsets in $[-25, 25]$ and $[-50, 50]$ mm at Levels 2 and 3, respectively. *Edge Missing* removes pixels around depth discontinuities exceeding 30 mm, using dilation radii of 1, 5, or 7 pixels and nominal removal fractions of 24–25%, modulated by depth at Levels 2–3. *Patch Hole* is enabled at Levels 2–3: each camera receives at most one connected hole targeting 1.5% of valid pixels per event, with a duration of 4–8 frames and onset offsets of 16–32 frames within each 80-frame cycle.

Table 8: Depth corruption parameters and success rates (%). Edge radius is the native-resolution four-neighbor dilation count; edge fraction is measured within the edge band. Patch area is relative to valid pixels during active events.

Level	σ_0/σ_1 (mm)	Shift (mm)	Edge radius	Edge fraction	Patch area	Clean	Random	Mean
Clean depth	0/0	0	–	0	0	87.40	54.10	70.75
1	12/20	0	1	25%	0	89.80	52.70	71.25
2	24/40	$[-25, 25]$	5	25%	1.5%	87.60	53.20	70.40
3	36/60	$[-50, 50]$	7	24%	1.5%	86.50	50.10	68.30

Training and evaluation. Each severity level uses 15k depth-only updates followed by 15k joint updates. Evaluation uses the matching corruption on 50 RoboTwin tasks, with 20 rollouts per task in each Clean/Random condition (1000 per condition; 2000 per level).

A.5 REAL-WORLD EVALUATION

Training. Both policies train on the same demonstrations for 30k total updates with a global batch size of 256 and seed 42. PrismWAM is initialized from Wan2.2-5B and uses 15k depth-only adaptation updates followed by 15k joint RGB/depth/action updates; OpenWAM- α uses 30k RGB/action updates. For setups without wrist-view depth, PrismWAM uses the separately pretrained head-view-only depth VAE with 256×320 inputs (Appendix A.2). The VAEs and T5 remain frozen during policy training. PrismWAM predicts absolute joint commands, and OpenWAM- α predicts absolute end-effector targets.

Embodiment and data. We use single-arm and dual-arm Piper-X configurations, each equipped with one head-view RGB-D camera (Orbbec Gemini 335). *Ranking Block Size* uses a single arm, while *Pick Up Plate* and *Open Microwave* use both arms. For these three tasks, we collect 100 demonstration episodes for *Open Microwave*, 200 for *Pick Up Plate*, and 180 for *Ranking Block Size*, with 30 episodes for each of the six size permutations. Each task has 50 goal-preserving training paraphrases.

Tasks and evaluation. Success requires (i) *Pick Up Plate*: pick up the plate with both arms and place it on the rack; (ii) *Ranking Block Size*: place the largest cube farthest from the robot, the medium cube centrally, and the smallest nearest; (iii) *Open Microwave*: open the door, move the inside basket to the tabletop on the microwave’s right, and leave the door open. ID uses the original setup, including all six size permutations for *Ranking Block Size*. As illustrated in fig. 6, OOD settings for all three tasks include flickering lights with ambient lights switched off and added distractor objects. Additional settings change plate colors for *Pick Up Plate* and tablecloths for *Ranking Block Size* and *Open Microwave*. Table 9 reports successes/trials; overall rates pool trials.

Table 9: Real-world successes/trials.

Task	OpenWAM- α		PrismWAM	
	ID	OOD	ID	OOD
Pick Up Plate	4/5	10/15	5/5	13/15
Ranking Block Size	7/12	3/10	11/12	6/10
Open Microwave	9/10	4/10	8/10	5/10

B DETAILED RESULTS ACROSS FIVE BENCHMARKS

B.1 LIBERO, LIBERO-PLUS, AND LIBERO-PRO

We evaluate PrismWAM on LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and LIBERO-Pro (Zhou et al., 2025) with an action horizon of 16 and replanning every 16 actions. LIBERO-Plus and LIBERO-Pro reuse the checkpoint trained on standard LIBERO without additional fine-tuning. For LIBERO-Plus, we apply the RGB sensor-noise settings to depth observations using the same corruption parameters. LIBERO-Pro evaluates five perturbation types with 2000 episodes per type.

Table 10: PrismWAM on standard LIBERO and LIBERO-Plus by base suite.

Suite	LIBERO Standard		LIBERO-Plus	
	Success / Total	SR (%)	Success / Total	SR (%)
Spatial	497/500	99.40	2075/2402	86.39
Object	500/500	100.00	2127/2518	84.47
Goal	491/500	98.20	1830/2591	70.63
Long	499/500	99.80	1936/2519	76.86
Overall	1987/2000	99.35	7968/10030	79.44

Table 11: PrismWAM on LIBERO-Plus by perturbation. Overall pools episodes rather than averaging rounded category percentages.

Perturbation	Success / Total	SR (%)
Camera viewpoints	969/1599	60.60
Robot initial states	1048/1550	67.61
Language instructions	1416/1537	92.13
Light conditions	1138/1142	99.65
Background textures	1071/1076	99.54
Sensor noise (matched RGB/depth)	1080/1601	67.46
Object layout	1246/1525	81.70
Overall	7968/10030	79.44

B.2 ROBOTWIN-C2R: PER-TASK RESULTS

RoboTwin-C2R trains the policy on clean demonstrations only and evaluates 50 tasks in Clean and Random conditions. PrismWAM and PrismWAM* results in table 12 use 100 episodes per task and condition.

Table 12: RoboTwin-C2R per-task SR (%) after clean-only training.

Task	PrismWAM		PrismWAM*	
	Clean	Rand.	Clean	Rand.
Adjust Bottle	98	54	100	97
Beat Block Hammer	81	48	98	58
Blocks Ranking RGB	87	32	93	72
Blocks Ranking Size	56	27	70	55
Click Alarmclock	100	86	100	98
Click Bell	100	78	100	93
Dump Bin Bigbin	84	80	92	82
Grab Roller	100	89	100	99
Handover Block	99	37	98	66
Handover Mic	95	24	100	80
Hanging Mug	38	17	59	41
Lift Pot	99	67	100	86
Move Can Pot	99	19	100	47
Move Pillbottle Pad	100	19	100	78
Move Playingcard Away	99	65	100	92
Move Stapler Pad	33	8	64	33
Open Laptop	91	50	94	83
Open Microwave	57	19	44	3
Pick Diverse Bottles	93	78	92	84
Pick Dual Bottles	100	96	100	96
Place A2B Left	83	39	92	74
Place A2B Right	80	41	88	70
Place Bread Basket	94	77	95	78
Place Bread Skillet	90	66	91	74
Place Burger Fries	98	90	96	100
Place Can Basket	81	31	79	61
Place Cans Plasticbox	100	95	100	99
Place Container Plate	100	68	100	93
Place Dual Shoes	73	43	94	70
Place Empty Cup	99	70	100	84
Place Fan	88	42	97	68
Place Mouse Pad	89	29	91	43
Place Object Basket	86	44	88	78
Place Object Scale	88	37	97	81
Place Object Stand	97	76	99	90
Place Phone Stand	88	47	94	75
Place Shoe	92	55	99	95
Press Stapler	92	50	85	80
Put Bottles Dustbin	90	82	92	93
Put Object Cabinet	89	28	90	52
Rotate QRcode	83	26	89	49
Scan Object	94	72	97	81
Shake Bottle	100	79	100	98
Shake Bottle Horizontally	100	80	100	97
Stack Blocks Three	90	33	95	87
Stack Blocks Two	98	60	100	98
Stack Bowls Three	84	59	82	79
Stack Bowls Two	94	74	96	87
Stamp Seal	80	23	95	33
Turn Switch	70	49	67	65
Mean over 50 tasks	87.98	53.16	91.84	75.50

B.3 ROBODOJO: CATEGORY, SEED, AND PER-TASK RESULTS

The simulation suite (Chen et al., 2026b) has 42 canonical tasks in five categories, represented by 54 standard/randomized entries per seed. Generalization uses 25 episodes/entry/seed; other categories use 50 (2100 episodes/seed, 6300 over seeds 0–2). Both policies use simulator depth, ten denoising steps, and replanning every 24 actions. We evaluate both policies with three seeds (0–2). Table 13 reports category and overall Score/SR as mean $\pm$ population standard deviation across seeds, together with each seed’s overall result; overall metrics average the five categories equally. Table 14 reports per-entry Score/SR averaged over the same three seeds.

Table 13: RoboDojo category and seed results. Score (/100) and SR (%) are mean $\pm$ population standard deviation over seeds 0–2; seed rows give each complete evaluation’s category-balanced metrics.

Category / seed	PrismWAM		PrismWAM*	
	Score	SR (%)	Score	SR (%)
Generalization	27.01 $\pm$ 1.12	19.56 $\pm$ 0.75	36.71 $\pm$ 0.93	29.33 $\pm$ 0.98
Precision	28.70 $\pm$ 0.98	19.50 $\pm$ 1.77	39.82 $\pm$ 2.21	31.08 $\pm$ 2.66
Long-horizon	37.39 $\pm$ 1.15	22.08 $\pm$ 0.77	48.67 $\pm$ 0.26	31.00 $\pm$ 0.74
Memory	30.46 $\pm$ 1.80	30.11 $\pm$ 1.81	50.96 $\pm$ 0.20	50.67 $\pm$ 0.27
Open	1.92 $\pm$ 0.46	1.50 $\pm$ 0.54	4.98 $\pm$ 0.52	4.42 $\pm$ 0.66
Overall	25.10 $\pm$ 0.68	18.55 $\pm$ 0.74	36.23 $\pm$ 0.55	29.30 $\pm$ 0.61
Seed 0	25.21	18.58	35.58	28.70
Seed 1	25.86	19.43	36.92	30.13
Seed 2	24.21	17.63	36.17	29.07

Table 14: All 54 RoboDojo evaluation entries: three-seed mean Score (/100) and SR (%). Rand. marks randomized Generalization entries.

Task	PrismWAM		PrismWAM*	
	Score	SR	Score	SR
<i>Generalization – standard</i>				
stack bowls	80.53	77.33	88.67	86.67
pour liquid into cup	54.67	54.67	60.00	60.00
push T	0.00	0.00	1.33	1.33
make toast	40.67	22.67	68.67	60.00
pack objects into box	25.73	4.00	31.33	5.33
arrange largest number	7.80	2.67	12.47	5.33
fold clothes	62.13	58.67	78.67	77.33
sort nesting dolls by size	14.67	14.67	32.00	32.00
hang mugs	42.53	20.00	73.60	61.33
store laptop and headphones	38.67	22.67	58.40	46.67
sweep blocks	2.67	2.67	8.00	8.00
stack blocks	74.60	72.00	93.20	92.00
<i>Generalization – randomized</i>				
stack bowls (Rand.)	31.87	22.67	21.20	8.00
pour liquid into cup (Rand.)	45.33	45.33	36.00	36.00
push T (Rand.)	0.00	0.00	0.00	0.00
make toast (Rand.)	27.00	10.67	22.67	10.67
pack objects into box (Rand.)	23.73	4.00	24.87	2.67
arrange largest number (Rand.)	2.73	0.00	8.93	1.33
fold clothes (Rand.)	13.07	8.00	86.40	84.00
sort nesting dolls by size (Rand.)	1.33	1.33	1.33	1.33
hang mugs (Rand.)	11.27	0.00	20.33	0.00
store laptop and headphones (Rand.)	40.27	22.67	32.53	17.33
sweep blocks (Rand.)	0.00	0.00	5.33	5.33
stack blocks (Rand.)	7.07	2.67	15.13	1.33

Continued on next page

Table 14 (continued)

Task	PrismWAM		PrismWAM*	
	Score	SR	Score	SR
<i>Precision</i>				
fasten screws	32.93	8.00	58.60	30.00
play Xylophone	4.00	4.00	0.00	0.00
insert tubes	74.67	60.67	82.53	74.00
deposit coin	15.33	9.33	40.00	34.67
plug in charger	1.33	1.33	10.67	10.67
insert key	14.40	0.00	14.40	0.00
pour balls into vase	38.67	38.67	58.00	58.00
build tower	48.27	34.00	54.33	41.33
<i>Long-horizon</i>				
put bottles into dustbin	95.60	92.67	99.50	99.33
fill egg holder	7.17	0.00	9.00	0.00
play tic tac toe	40.43	10.67	31.30	4.67
organize table	59.83	7.33	55.00	4.00
classify objects	5.70	1.33	53.20	42.67
play stacking toy	5.00	0.00	42.93	19.33
fill pen holder	62.03	41.33	65.73	45.33
make kong	23.33	23.33	32.67	32.67
<i>Memory</i>				
cover blocks	100.00	100.00	100.00	100.00
press by number	20.00	20.00	81.33	81.33
match and pick from conveyor	51.33	51.33	89.33	89.33
imitate sorting sequence	2.77	0.67	2.40	0.67
swap T	8.67	8.67	32.67	32.67
swap blocks	0.00	0.00	0.00	0.00
<i>Open</i>				
align blocks	0.00	0.00	0.00	0.00
classify objects by language	1.07	0.00	6.00	2.67
solve equation	0.00	0.00	0.00	0.00
pick from conveyor by image	0.00	0.00	0.00	0.00
stack blocks by language	7.60	5.33	11.33	10.67
store tools in toolbox	0.00	0.00	0.50	0.00
general pickup	6.67	6.67	22.00	22.00
pour by language	0.00	0.00	0.00	0.00
Overall (five categories)	25.10	18.55	36.23	29.30