ΔWAM: DISTILLING ACTION TANGENT FIELDS INTO WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

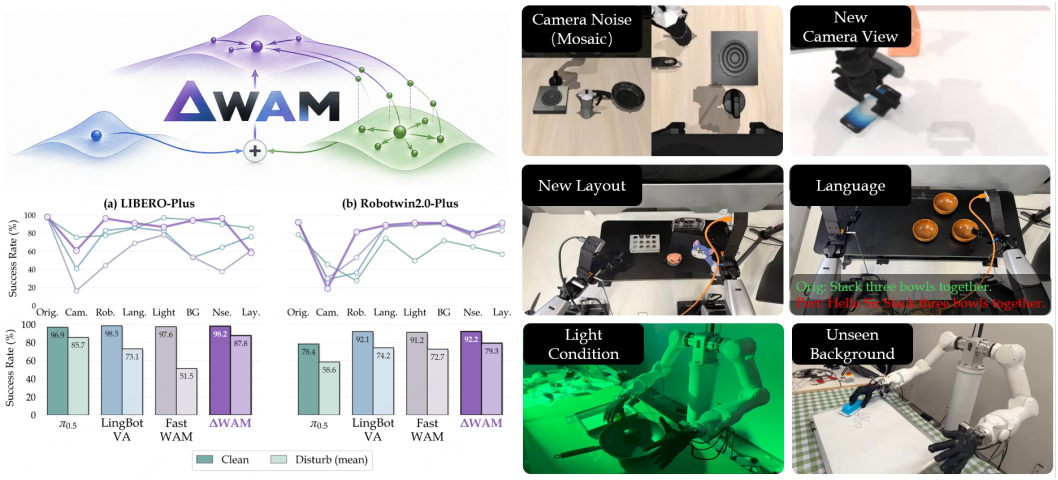


Figure 1: **ΔWAM under distribution shift**. Left: success rate by perturbation type on LIBERO-Plus and RoboTwin 2.0-Plus, with the average over perturbations shown for each method. Right: the corresponding shifts, camera noise, novel viewpoints and layouts, language, lighting, and unseen backgrounds.

ABSTRACT

World Action Models (WAM) improve robot policies by augmenting sparse action supervision with dense future prediction. However, much of the predictable future is dominated by appearance and scene persistence rather than action-dependent dynamics. We observe that several recent WAM designs, including optical flow, motion-centric representations, and latent actions, can be understood from a common perspective in which world supervision becomes more efficient as it contains a higher proportion of action-relevant variation. Based on this insight, we introduce Action Tangent Fields, which reformulate world supervision through a local Taylor expansion of how actions induce changes in future dynamics. We represent future dynamics in Residual-VAE space, where the future latent remains recoverable from the current latent and its residual, and use a strong action-conditioned world model (ACWM) to probe the local correspondence between action variations and residual-world variations. This local first-order structure is distilled into the WAM to guide its denoising supervision toward dynamics that are more tightly coupled to action, rather than merely predictable from appearance. Across LIBERO-Plus, RoboTwin, and RoboTwin2.0-Plus, our method consistently improves robustness to lighting, background, camera, layout, and other environmental perturbations. Despite using no large-scale embodied pretraining, it achieves stronger robustness under several distribution shifts than pretrained policies. We further distill multi-step VideoDiT denoising into a single step for efficient inference. Our results suggest that effective WAM supervision should remain information-rich while concentrating its predictive capacity on the directions along which actions change the future.

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.

1 INTRODUCTION

Behavior cloning labels each demonstrated observation with an action. World Action Models (WAMs) also predict the future video, giving dense supervision for object motion, contact, and state changes that a single action label cannot describe. A video target, however, rewards predictability rather than control relevance. Backgrounds, camera views, and untouched objects are already determined by the current frame and persist as the robot acts, so a model that reproduces them reduces its loss substantially while learning little about the consequences of its own actions. The value of world supervision therefore depends less on how much of the future it covers than on how much of that future is action-dependent.

Recent WAM designs change the target itself. Optical-flow and motion-centric targets keep only what moves, and latent-action targets compress the future into a compact code. Both reduce the supervision spent on static appearance, which is why they help, but both rely on motion as a proxy for action-dependence: a lighting change or a moving distractor alters the future with no contribution from the robot, and dropping appearance removes cues that contact-rich manipulation needs. A change of target also cannot reveal which directions of the future respond to the action, because each demonstration records one action and the single future that followed it. It is an isolated point of the action–dynamics relation, and its local neighborhood is never observed.

Our first step therefore keeps the target information-rich and changes its coordinate instead. Given the current frame, we subtract a broadcast reference derived from it, so the residual describes how the scene departs from the observed frame rather than restating it. Reference and residual recover the future VAE latent exactly, so no information is lost, and because the reference depends only on the current frame, subtracting it leaves the response to an action perturbation unchanged.

The second step supplies the missing neighborhood through a probe. For a fixed scene, an embodiment-specific action-conditioned world model (ACWM) is queried at small, temporally smooth perturbations of the expert action. The resulting residual futures differ mainly along directions the ACWM treats as action-sensitive, and their normalized differences span an Action Tangent Field (ATF), a low-dimensional estimate of those directions. Post-training keeps every element of the demonstrated residual in the loss and adds weight along the ATF, which makes the world loss anisotropic, ordinary world post-training is the isotropic special case. The probe sets the weighting, never the targets.

We first mid-train the Wan VideoDiT on a modest amount of robot manipulation video to adapt its prediction space to the residual latent space, then post-train a Fast-WAM-style policy with cached ATFs. Caching keeps the ACWM out of policy training and inference. Extensive experiments show that supervising dynamics-relevant world representations enables our method to generalize to visual perturbations substantially better than competing methods.

Our contributions are as follows.

- We introduce an information-preserving Residual-VAE target that shifts world prediction toward scene dynamics while retaining exact recovery of the future video.
- Motivated by a first-order Taylor expansion of the action-to-future dynamics map, we introduce Action Tangent Fields to estimate locally action-responsive directions and emphasize them during denoising supervision of the real demonstrated future.
- We obtain efficient world features with one VideoDiT evaluation at inference. Aggregate success improves from 51.5% to 87.8% over Fast-WAM on LIBERO-Plus and from 72.7% to 79.3% on RoboTwin 2.0-Plus.

2 RELATED WORK

2.1 WORLD ACTION MODELS

World Action Models (WAMs) complement sparse action labels with future prediction, encouraging policies to retain object states, interactions, and temporal dynamics. Predict-then-act methods such as GE-Act generate compact visual futures and recover controls through inverse dynamics, while GE-Act 2.0 scales this design with heterogeneous interaction data (Liao et al., 2025). Joint models

instead couple world and action generation: LingBot-VA autoregressively interleaves both outputs, whereas Cosmos Policy represents future observations, actions, and values as latent frames of a video diffusion model (Li et al., 2026c; Kim et al., 2026).

A complementary line treats prediction primarily as representation supervision. Fast-WAM and Light-WAM retain video co-training but omit explicit generation at deployment (Yuan et al., 2026; Li et al., 2026d), whereas Efficient-WAM and Faster-WAM preserve future latents while reducing generation or video–action interaction costs (Li et al., 2026b; Zhao et al., 2026). The prediction target nevertheless determines how model capacity is allocated: full video latents preserve rich details but are dominated by static appearance and other variations weakly related to control, whereas flow, residual, and motion-centric targets emphasize change but may discard useful cues. In contrast, Action Tangent Fields retain an information-rich residual future space and selectively emphasize locally action-responsive directions.

2.2 Action-Conditioned World Models

Action-conditioned world models (ACWMs) predict visual consequences under specified controls. Recent systems broaden the control interface: Cosmos 3 jointly models language, images, video, audio, and actions (NVIDIA Cosmos Team, 2026), Hydra-0 represents actions as pixel-space flow for cross-embodiment transfer (Li et al., 2026a), and Pelican-Sim combines a unified numerical action space with rendered action videos (Zou et al., 2026). These methods condition generation through native action tokens, visual motion, and action-rendered observations, respectively.

Another line improves simulation reliability. ABot-PhysWorld uses physics-aware data and preference optimization to suppress implausible interactions (Chen et al., 2026b). Ctrl-World combines multi-view prediction, frame-level action conditioning, and pose-aware memory for policy-in-the-loop rollouts (Guo et al., 2026), while WALL-SS uses coarse-to-fine autoregression and compressed memory for long streaming generation (Zhang et al., 2026a). WorldSync addresses failures on off-expert commands through broader action coverage, dynamics grounding, and intervention-effect alignment (Chen et al., 2026a). Unlike these approaches, we neither plan with generated rollouts nor use synthetic futures as supervision targets. Instead, we perturb demonstrated actions and use the ACWM as a local differential probe in Residual-VAE space, transferring the resulting first-order action–world geometry while retaining real demonstrations as prediction targets.

3 Method

We introduce ΔWAM to train a world-action policy on real future video while transferring local action sensitivity from an action-conditioned world model (ACWM). Figure 2 outlines the method. We first adapt Wan to predict future video latents relative to the current frame (Wan Team et al., 2025). We then perturb demonstrated actions and query an embodiment-specific ACWM (NVIDIA Cosmos Team, 2026) to extract action tangent fields (ATFs), which identify locally action-responsive directions. Finally, we post-train a Fast-WAM-style VideoDiT–ActionDiT policy (Yuan et al., 2026) on real action–future pairs, giving additional weight to prediction errors along those directions. At inference, one VideoDiT pass provides the world features used to generate an action chunk.

3.1 Residual Video Mid-Training

The pretrained Wan VideoDiT predicts absolute future VAE latents, whereas our policy predicts future latents relative to the observed frame. Although the two representations contain the same information given that frame, they are different prediction targets. We therefore mid-train Wan before policy post-training to adapt its prediction space from absolute VAE latents to Residual-VAE latents. The known first frame carries persistent scene content, leaving the VideoDiT to predict changes such as robot motion, object displacement, and contact.

Let E and D denote the frozen Wan VAE encoder and decoder. We encode the observed first frame and the real future clip, then broadcast the first-frame code across the future latent positions:

$$\begin{aligned} Z_t &= E(I_t), & Z_{t+1:t+H} &= E(I_{t+1:t+H}), \\ B_t &= \mathrm{broadcast}(Z_t), & R_{t+1:t+H} &= Z_{t+1:t+H} - B_t. \end{aligned} \tag{1}$$

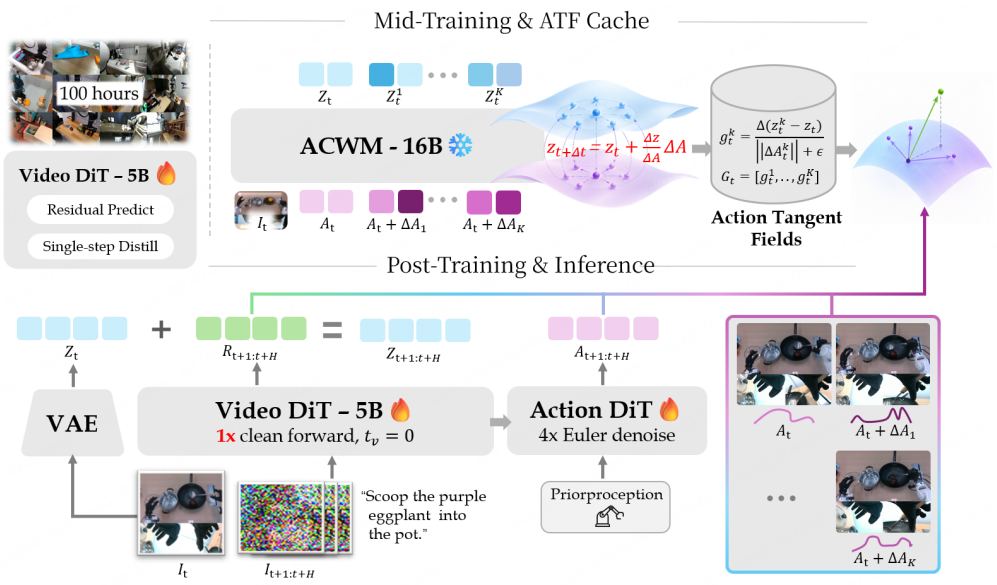


Figure 2: **Overview of ΔWAM**. A Wan VideoDiT-5B is mid-trained to predict the Residual-VAE future $R_{t+1:t+H}$. An embodiment-specific ACWM is then queried at K perturbed actions, and its normalized response differences are orthonormalized into a cached basis G_t of locally action-sensitive directions. Post-training keeps the real residual as the target and penalizes its projection onto G_t, one VideoDiT pass at $t_v = 0$ conditions four ActionDiT Euler steps at inference.

Here broadcast aligns the first-frame code with the shape of $Z_{t+1:t+H}$. The anchor B_t is known at prediction time, only the residual $R_{t+1:t+H}$ must be predicted. This change of coordinates loses no information relative to the original Wan latent:

$$Z_{t+1:t+H} = B_t + R_{t+1:t+H}, \qquad D(B_t + R_{t+1:t+H}) = D(Z_{t+1:t+H}). \tag{2}$$

Thus the first-frame anchor plus the future residual recovers exactly the absolute latent, and the Wan decoder produces the same video it would have produced from that latent. The claim is about preserving the VAE representation, not exact recovery of the original RGB pixels. The subtraction centers prediction on departures from the observed scene while retaining changes in robot pose, object state, and contact. Because B_t is fixed when I_t is fixed, subtracting it does not change the local response to an action perturbation: $\Delta R_{t+1:t+H} = \Delta Z_{t+1:t+H}$. This lets ATF identify action-related directions in the same information-rich latent space used by the VideoDiT.

For this prediction-space adaptation, we select 100 hours of robotics video from the DROID Khazatsky et al. (2024) and BridgeV2 Walke et al. (2023) portions of the Cosmos training corpus. We initialize the VideoDiT from pretrained Wan and train it to predict residuals using the same flow-matching parameterization. For video flow time t_v, Gaussian noise ϵ_R, and the Wan interpolation coefficients α_{t_v} and σ_{t_v}, the noisy residual and its velocity target are

$$x^R_{t,t_v} = \alpha_{t_v} R_{t+1:t+H} + \sigma_{t_v} \epsilon_R, \qquad u^{R,\star}_{t,t_v} = \dot{\alpha}_{t_v} R_{t+1:t+H} + \dot{\sigma}_{t_v} \epsilon_R. \tag{3}$$

The residual VideoDiT $F_{\theta_{\text{mid}}}$ minimizes

$$\mathcal{L}_{\text{mid}} = \mathbb{E}_{t,t_v,\epsilon_R}\left[\left\| F_{\theta_{\text{mid}}}(x^R_{t,t_v}, t_v, I_t, l_t) - u^{R,\star}_{t,t_v} \right\|_2^2\right]. \tag{4}$$

This stage aligns the pretrained VideoDiT with the Residual-VAE target used in policy post-training, it requires video but no action labels. The residual captures what changed, while the next stage uses a local action neighborhood to identify which changes respond to action.

3.2 Action Tangent Fields from Local Dynamics

The demonstrated residual $R_{t+1:t+H}$ gives one realization of future dynamics under A_t. To ask which parts of these dynamics depend on the action, imagine the unknown map $f_t(A)$ from a nearby

action chunk to the vectorized future residual at the same current state. Where this map is differentiable, a small action change produces

$$f_t(A_t + \Delta A) - f_t(A_t) \approx J_t \,\mathrm{vec}(\Delta A), \qquad J_t = \left.\frac{\partial f_t}{\partial\, \mathrm{vec}(A)}\right|_{A_t}. \tag{5}$$

The directions in the output of J_t are the possible local changes in video dynamics induced by action. A single demonstrated future cannot reveal them. We therefore adapt a 16B ACWM NVIDIA Cosmos Team (2026) and posttrain them separately for each robot embodiment and use it to sample nearby futures. Let $b(t)$ identify the embodiment and $\Phi_{b(t)}$ its frozen ACWM. The ACWM provides a neighborhood of dynamics, not replacement targets for the policy.

We sample $K = 8$ small, temporally smooth perturbations ΔA_t^k in normalized action coordinates and set $A_t^k = A_t + \Delta A_t^k$. Normalization makes perturbation magnitudes comparable across action dimensions, while random directions probe the neighborhood without selecting a preferred coordinate axis. For the reference and perturbed actions we hold the observation, instruction, proprioception, and generation conditions ξ_t fixed:

$$\widetilde{I}_t^0 = \Phi_{b(t)}(I_t, l_t, p_t, A_t, \xi_t), \qquad \widetilde{I}_t^k = \Phi_{b(t)}(I_t, l_t, p_t, A_t^k, \xi_t). \tag{6}$$

We sample frames at the policy's prediction times and encode the generated futures with the same frozen VAE. In Figure 2, $z_t = E(\widetilde{I}_t^0)$ is the ACWM future under the expert action and $z_t^k = E(\widetilde{I}_t^k)$ is its future under perturbation k, lowercase z distinguishes these generated futures from the observed latent Z_t. Their difference removes the common first-frame anchor:

$$\begin{aligned} \Delta R_t^k &= (z_t^k - B_t) - (z_t - B_t) = z_t^k - z_t, \\ \mathrm{vec}(\Delta R_t^k) &\approx \widetilde{J}_t \,\mathrm{vec}(\Delta A_t^k). \end{aligned} \tag{7}$$

Here $\widetilde{J}_t$ is the ACWM's local action–world Jacobian. The finite difference therefore estimates a direction in which *future dynamics* change under a nearby action. It approximates the teacher's local response, not the unknown environment Jacobian J_t in equation 5. Matched generation conditions help isolate action effects from sampling variation.

We divide each response by perturbation magnitude so that its scale reflects a directional sensitivity, then retain the output directions rather than attempting to estimate all entries of $\widetilde{J}_t$:

$$g_t^k = \frac{\mathrm{vec}(\Delta R_t^k)}{\|\Delta A_t^k\|_2 + \varepsilon}, \qquad G_t = [g_t^1, \ldots, g_t^K] \in \mathbb{R}^{D_R \times K}, \qquad Q_t = \mathrm{orth}(G_t). \tag{8}$$

Here D_R is the number of elements in one future residual. If U_t collects the normalized perturbation directions, the finite-difference relation gives $G_t \approx \widetilde{J}_t U_t$. Thus the columns of $Q_t \in \mathbb{R}^{D_R \times r_t}$, where $r_t \le K$ and $Q_t^\top Q_t = I_{r_t}$, span only the ACWM output directions reached by the eight sampled actions. The point is to locate a small, action-responsive neighborhood of the dynamics target, not to recover the full Jacobian. We cache Q_t and its rank with the training window, the ACWM is not used during policy training or inference.

3.3 ATF-Guided World-Action Post-Training

The first stage gives the VideoDiT a dense target for real future dynamics, the second marks directions in that target that vary across a local action neighborhood. We combine them during Fast-WAM-style post-training. The policy VideoDiT is initialized from the residual-mid-trained Wan backbone and supplies world features to an ActionDiT. Its world branch is conditioned on the observed scene and instruction, not on a candidate action, so we use Q_t to weight errors in predicted dynamics rather than asking the policy to reproduce an action Jacobian.

During post-training, the VideoDiT uses the same single-pass computation as at inference. At fixed video time $t_v = 0$, it processes the known latent reference B_t and produces world tokens h_t^W. With the flow-output sign convention of our implementation, its output head O_θ gives a clean residual estimate:

$$h_t^W = F_\theta(B_t, l_t, t_v = 0), \qquad \widehat{R}_{t+1:t+H} = -O_\theta(h_t^W)_{\mathrm{future}}. \tag{9}$$

The target is always the residual of the *real* demonstrated future, and the ordinary loss covers every latent element:

$$\mathcal{L}_R = \mathbb{E}_t \left[\frac{1}{D_R} \| \operatorname{vec}(\widehat{R}_{t+1:t+H} - R_{t+1:t+H}) \|_2^2 \right]. \tag{10}$$

For action flow time τ and Gaussian noise ϵ_A, let $A_{t,\tau} = (1-\tau)A_t + \tau\epsilon_A$. The ActionDiT predicts the velocity of this interpolation from the noisy action, proprioception, instruction, and world tokens:

$$\mathcal{L}_A = \mathbb{E}_{t,\tau,\epsilon_A} \left[\left\| U_\phi(A_{t,\tau}, \tau, p_t, l_t, h_t^W) - (\epsilon_A - A_t) \right\|_2^2 \right]. \tag{11}$$

The action and residual losses learn from the observed action–future pair. To make the world loss sensitive to its local action neighborhood, define the real-future dynamics error $e_t = \operatorname{vec}(\widehat{R}_{t+1:t+H} - R_{t+1:t+H})$ and add a penalty on its projection onto Q_t:

$$\mathcal{L}_{\mathrm{ATF}} = \mathbb{E}_t \left[\frac{1}{r_t} \| Q_t^\top e_t \|_2^2 \right], \tag{12}$$

with a zero contribution when $r_t = 0$. Since Q_t has orthonormal columns, this is the squared error along the sampled action-responsive directions. For each window with $r_t > 0$, the two world losses can be written as

$$\frac{\lambda_R}{D_R} \|e_t\|_2^2 + \frac{\lambda_T}{r_t} \|Q_t^\top e_t\|_2^2 = e_t^\top \left(\frac{\lambda_R}{D_R} I_{D_R} + \frac{\lambda_T}{r_t} Q_t Q_t^\top \right) e_t. \tag{13}$$

Every direction retains the ordinary residual weight, while directions in $\operatorname{span}(Q_t)$ receive additional weight. The full future latent is therefore still learned, but errors aligned with local action effects matter more.

The complete post-training objective is

$$\mathcal{L}_{\mathrm{post}} = \lambda_A \mathcal{L}_A + \lambda_R \mathcal{L}_R + \lambda_T \mathcal{L}_{\mathrm{ATF}}. \tag{14}$$

The VideoDiT still learns the dynamics of the real demonstrated future. The ATF term uses nearby, ACWM-generated outcomes only to identify which residual errors should matter more for action. Thus pointwise action and video targets are enriched by local action-neighborhood structure without training on synthetic futures.

3.4 Single-Pass World Conditioning at Inference

At a replanning step, the policy receives the latest image I_t, instruction l_t, and proprioceptive state p_t. It encodes I_t to Z_t, forms the broadcast reference B_t, and evaluates the VideoDiT once at $t_v = 0$, exactly as in equation 9. The resulting world tokens h_t^W summarize the observed scene and the dynamics learned during residual and ATF-guided post-training. Action generation consumes these tokens directly: it does not need to decode $\widehat{R}_{t+1:t+H}$ into video. Because the world branch is independent of the candidate action chunk, h_t^W is computed once and reused throughout action sampling.

We sample an initial action chunk $a_t^{(0)} \sim \mathcal{N}(0, I)$ and integrate the learned action flow from noise time $\tau_0 = 1$ to data time $\tau_4 = 0$ using four descending Euler steps:

$$a_t^{(j+1)} = a_t^{(j)} + (\tau_{j+1} - \tau_j) U_\phi\left(a_t^{(j)}, \tau_j, p_t, l_t, h_t^W\right), \qquad j = 0, \ldots, 3. \tag{15}$$

The negative time increment follows the action-flow parameterization in equation 11, where $\tau = 1$ denotes noise and $\tau = 0$ denotes the expert action. Only the noisy action chunk and its flow time change between Euler steps, the image encoding and VideoDiT features remain fixed. The final state $\widehat{A}_{t+1:t+H} = a_t^{(4)}$ is the predicted action chunk.

4 Experiments

We evaluate ΔWAM on three simulation benchmarks and two real robot platforms. LIBERO-Plus and RoboTwin 2.0-Plus measure each perturbation dimension separately, which shows where action-tangent supervision helps and where it does not. RoboTwin, RoboTwin 2.0-Plus, and Tianji involve bimanual manipulation, and the real-world tasks add sustained contact and a 20-DoF dexterous hand action space. The real-world platform detail is provided in the Appendix. We then ablate the number of VideoDiT denoising steps and the number of action perturbations.

Table 1: **LIBERO-Plus evaluation results.** We report success rates (%) under the original setting and seven perturbation dimensions Zhang et al. (2026b). The best result in each column is shown in **bold**, while the second-best result is underlined.

Model	Original	Camera	Robot	Lang.	Light	BG	Noise	Layout	Total
VLAs									
π_0 (2025b)	94.2	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0 rerun (2025b)	91.3	61.0	40.8	63.5	89.3	84.1	80.1	76.4	69.4
π_0-FAST (2025)	85.5	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
$\pi_{0.5}$ (2025a)	96.9	75.4	77.5	85.6	**96.9**	94.6	89.7	**85.7**	85.7
OpenVLA-OFT$_m$ (2025)	97.6	55.6	21.7	81.0	92.7	91.0	78.6	68.7	67.9
UniVLA (2025)	95.2	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
RIPT-VLA (2025)	97.5	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
X-VLA (2025)	98.1	23.4	89.7	75.7	88.2	**96.0**	62.7	71.8	71.4
HoloBrain0-GD (2026)	96.7	65.5	58.2	78.7	88.1	90.3	66.9	79.5	74.0
ABot-M0 (2026)	**98.6**	60.4	67.9	86.4	96.2	91.6	86.4	82.6	80.5
VLA + WM									
VLA-JEPA (2026)	97.2	64.2	67.7	88.1	91.8	93.4	65.8	83.9	77.9
WAMs									
GE-Act (2025)	94.4	60.7	77.0	77.4	95.8	86.0	90.9	80.2	80.3
Cosmos Policy (2026)	98.5	75.8	63.3	81.7	96.5	88.9	92.7	82.2	82.2
Fast-WAM (2026)	97.6	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
Ours	98.2	**60.9**	**96.2**	**91.2**	86.8	94.1	**96.4**	58.4	**87.8**

4.1 LIBERO-PLUS

LIBERO-Plus Fei et al. (2025) extends LIBERO Liu et al. (2023) with controlled shifts along seven dimensions: camera viewpoint, robot initialization, language, lighting, background, observation noise, and object layout. We fine-tune on the official demonstrations for 10 epochs and evaluate one checkpoint under every condition against vision-language-action policies (VLAs), VLA-plus-world-model methods, and WAMs. We obtain the best aggregate success rate of **87.8%**, 2.1 points above the strongest baseline $\pi_{0.5}$. Our 98.2% original-setting rate is also close to the best reported result of 98.6%, indicating that the aggregate robustness gain does not come at a substantial cost to in-distribution performance, and does not require the large-scale embodied pretraining that several competing VLAs inherit. Total is the pooled success rate across all perturbation instances (total successes divided by total evaluated instances, see Appendix A.2 and Table 4).

The clearest gains occur under robot-initialization, language, and observation-noise shifts. We achieve 96.2% under robot-initialization shifts, 6.5 points above the best competitor, and rank first on language (91.2%), respectively. Background performance is also strong at 94.1%, although $\pi_{0.5}$ and X-VLA remain ahead. Camera shifts present a different picture: our 60.9% success rate trails the best competitor by 14.9 points. Lighting (86.8%) and especially layout (58.4%) also remain challenging. These results suggest that weighting residual errors along action-responsive directions helps with some changes in embodiment and observation, but does not by itself resolve viewpoint shifts or large geometric rearrangements that require scene-level reasoning.

4.2 ROBOTWIN AND ROBOTWIN 2.0-PLUS

RoboTwin Chen et al. (2025) and RoboTwin 2.0-Plus Zhang et al. (2026b) stress bimanual coordination across diverse dual-arm tasks: RoboTwin reports the Clean and Randomized settings, and RoboTwin 2.0-Plus perturbs the same seven dimensions as LIBERO-Plus. Each benchmark uses one unified multi-task model trained for 5 epochs and evaluated with the same checkpoint without large-scale embodied pretraining, which lets us read the data efficiency of ATF supervision directly.

We reach 92.8% on Clean and 92.1% on Randomized, the best average of **92.4%**: within 0.1 points of pretrained LingBot-VA on Clean, ahead of it by 0.6 on Randomized. The 0.7-point Clean–

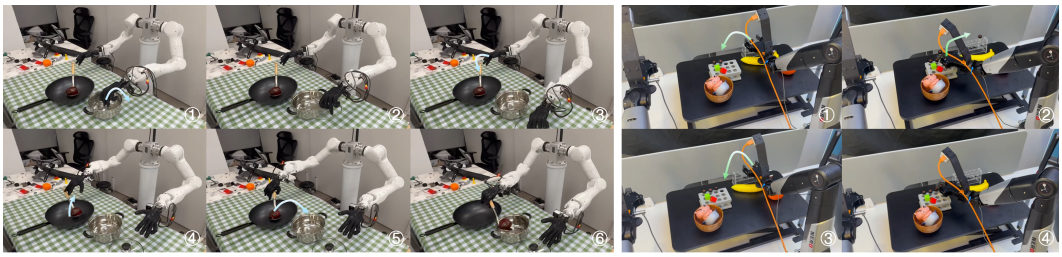
Table 2: **Results on RoboTwin and RoboTwin 2.0-Plus.** We report success rates (%) on RoboTwin under the Clean and Randomized settings, and on RoboTwin 2.0-Plus under the original setting and seven perturbation dimensions. For RoboTwin 2.0-Plus, all methods use a single unified model across all tasks. The best result in each column is highlighted in **bold**, while the second-best result is underlined. Embodied PT.: embodied pretraining. Lang.: Language. BG: Background.

	RoboTwin				RoboTwin 2.0-Plus								
Method	Embodied PT.	Clean	Rand.	Avg.	Original	Camera	Robot	Lang.	Light	BG	Noise	Layout	Total
π_0 (2025b)	✓	65.92	58.40	62.2	–	–	–	–	–	–	–	–	–
$\pi_{0.5}$ (2025a)	✓	82.74	76.76	79.8	78.4	**45.6**	27.6	74.4	49.6	71.7	64.9	56.8	58.6
X-VLA (2025)	–	–	–	–	65.6	23.2	65.2	64.4	63.1	58.6	49.7	34.8	53.1
MOTUS (2025)	✓	88.66	87.02	87.8	87.0	21.6	**85.0**	83.2	84.6	84.4	43.1	82.8	71.5
LingBot-VA (2026c)	✓	**92.90**	91.50	92.2	92.1	28.9	36.2	87.3	89.0	**91.3**	**80.9**	87.9	74.2
Fast-WAM (2026)	×	91.88	91.78	91.8	91.2	30.4	53.2	86.7	88.8	90.0	76.4	83.2	72.7
Ours	×	92.80	**92.10**	**92.4**	**92.2**	19.1	81.4	**88.8**	**91.4**	91.2	78.8	**91.3**	**79.3**

Randomized gap indicates little reliance on training-scene appearance, so ATF supervision matches WAMs that use substantial embodied pretraining from task-specific data alone.

Under the perturbation-wise protocol we obtain the best original-setting score (92.2%) and raise the aggregate to **79.3%**, 5.1 points above LingBot-VA. We rank first under language (88.8%), lighting (91.4%), and layout (91.3%), ahead of the respective runners-up by 1.5, 2.4, and 3.4 points, and reach 81.4% under robot perturbations, second only to MOTUS. Gains are not uniform: background and observation noise trail the best results by 0.1 and 2.1 points, and camera perturbation is the clearest weakness at 19.1%. Together with the LIBERO-Plus layout result, this indicates that a large change in observation geometry moves the input outside the local action–world neighborhood the tangent estimate covers, while within that neighborhood the objective follows action-induced rather than merely visually salient variation.

4.3 Real-World Experiments across Embodiments



(a) "Cook Meal" in New Background. (b) "Insert tubes" in New Layout.

Figure 3: **Real-world rollouts across embodiments.** Left: Tianji performs bimanual cooking with dexterous hands under a new background. Right: NERO inserts test tubes with a parallel gripper under new objects layout, numbered frames show the progression of each task.

We evaluate on two real-world platforms, NERO and Tianji, covering parallel-gripper and dexterous-hand manipulation over five tasks: test-tube insertion, bowl stacking, language-conditioned ordered object transfer, whiteboard wiping, and bimanual cooking. Each policy is tested under four conditions (standard, red light, green light, and tabletop-background replacement), with three trials per task–condition pair, we report the average progress score. Models are trained and evaluated separately per task for 10 epochs. Task descriptions and platform setups are in Appendix A.3.

Table 3 compares $\pi_{0.5}$, Fast-WAM, and our method under identical task-specific data and evaluation conditions. The NERO tasks require precise insertion and rigid-object placement with parallel grippers, whereas the Tianji tasks require language-conditioned sequencing, sustained contact, bimanual coordination, and tool use with dexterous hands. ATF supervision transfers across both regimes: on NERO, ΔWAM averages 0.79 on test-tube insertion against 0.58 for $\pi_{0.5}$ and 0.54 on bowl stacking against 0.33 for both baselines, on Tianji it leads every task average, with the largest margin on bimanual cooking, where the baselines score at most 0.02.

Table 3: **Real-world evaluation on NERO and Tianji**. Scores are averaged over three trials for each task and condition. The Average column is the mean across the four conditions.

7-DoF NERO Arm & Gripper

Task	Method	Standard	Layout 1	Layout 2	Red & Green Light	Average
Stack Bowls	$\pi_{0.5}$ (2025a)	0.33	0.33	0.17	**0.50**	0.33
	FastWAM (2026)	0.50	**0.50**	0.17	0.17	0.33
	ΔWAM (Ours)	**0.67**	**0.50**	**0.50**	**0.50**	**0.54**
Insert Tubes	$\pi_{0.5}$ (2025a)	0.83	0.50	0.50	0.50	0.58
	FastWAM (2026)	0.50	0.50	0.33	0.50	0.46
	ΔWAM (Ours)	**1.00**	**0.83**	**0.67**	**0.67**	**0.79**

7-DoF Tianji Arm & 20-DoF Wuji Dexterous Hand

Task	Method	Standard	Red Light	Green Light	New Background	Average
Move Ingredients	$\pi_{0.5}$ (2025a)	**0.73**	0.53	0.10	0.10	0.37
	FastWAM (2026)	0.40	0.20	0.10	0.10	0.20
	ΔWAM (Ours)	0.63	**0.73**	**0.60**	**0.60**	**0.64**
Wipe Board	$\pi_{0.5}$ (2025a)	0.57	0.50	0.27	0.10	0.36
	FastWAM (2026)	0.43	0.37	**0.47**	**0.30**	0.39
	ΔWAM (Ours)	**0.80**	**0.53**	0.33	0.20	**0.47**
Cook Meal	$\pi_{0.5}$ (2025a)	0.08	0.00	0.00	0.00	0.02
	FastWAM (2026)	–	–	–	–	–
	ΔWAM (Ours)	**0.42**	**0.42**	**0.25**	**0.42**	**0.38**

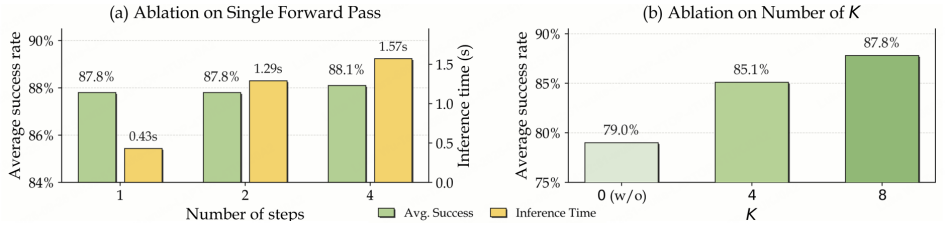


Figure 4: **Ablations on LIBERO-Plus.** (a) Average success and inference time for different VideoDiT denoising steps. (b) Average success for $K = 0, 4, 8$ local action perturbations.

4.4 Ablation Study

We ablate two design choices on LIBERO-Plus: the number of VideoDiT denoising steps at inference and the number K of local action perturbations used to construct the ATF. Figure 4 reports average success, and panel (a) also reports inference time.

VideoDiT denoising steps. One and two steps both achieve 87.8% average success and four steps reach 88.1%, while inference time rises from 0.43 s to 1.29 s and 1.57 s. Three extra steps thus buy 0.3 points at 1.14 s of additional latency, so we use one step, which keeps nearly the same success rate without iterative refinement.

Local action neighborhood. At $K = 0$, the policy retains the real action and residual targets but receives no ATF loss, achieving 79.0% success. Success rises to 85.1% at $K = 4$ and 87.8% at $K = 8$, an 8.8-point gain that motivates our use of $K = 8$ throughout.

5 Conclusion

ΔWAM learns future video dynamics and how they vary within a local neighborhood of the demonstrated action. Residual-VAE prediction keeps the full future latent while making temporal change explicit, and the ATF loss upweights errors along directions that respond to nearby actions, keeping the real future as the target. One VideoDiT pass then suffices at inference. Gains hold on LIBERO-Plus, RoboTwin, and RoboTwin 2.0-Plus, though large camera and layout changes remain challenging.

AI USE STATEMENT

Large language models were used to assist with programming and figure preparation, as well as grammar and expression polishing. They did not contribute to research ideation, methodology design, or experiments.

ETHICS STATEMENT

This work studies robot manipulation policies learned from video and action demonstrations. The real-world evaluation involves no human subjects, and no personally identifiable information is collected or released. Experiments on the NERO and Tianji platforms were conducted on table-top manipulation tasks with a two-minute per-trial limit, under the standard operational procedures of the respective hardware. All simulation benchmarks and video corpora used for mid-training are publicly available, and the real-world demonstrations were collected by the authors. We do not foresee direct ethical concerns arising from this work, while noting that general-purpose manipulation policies carry the usual dual-use considerations of the robotics field.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of our work. The proposed model, implementation details, and evaluation protocols are described in detail in the main paper and appendix. The source code and data will be made publicly available.

REFERENCES

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. *arXiv preprint arXiv:2512.13030*, 2025.

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, 2025a. URL https://arxiv.org/abs/2504.16054.

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. In *Robotics: Science and Systems*, 2025b. URL https://arxiv.org/abs/2410.24164.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2505.06111.

Sixiang Chen, Jiaming Liu, Jixian Wu, Yichen Guo, Tinghao Wang, Siyuan Qian, Hao Chen, Jiajun Cao, Jian Tang, and Shanghang Zhang. Do robotic world models really follow actions? diagnosing and aligning action-conditioned generation for policy learning. *arXiv preprint arXiv:2608.24885*, 2026a.

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

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

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

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, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2502.19645.

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

Hongyu Li, Bowen Wen, Xinghao Zhu, Yixuan Wang, Yilun Du, Yunzhu Li, George Konidaris, Stan Birchfield, Soha Pouya, Chenran Li, et al. Hydra-0: Action flow for generalist world modeling and control. *arXiv preprint arXiv:2608.18077*, 2026a.

Jiajun Li, Tiecheng Guo, Yifan Ye, Rongyu Zhang, Xiaowei Chi, Qianpu Sun, Ying Li, Yunfan Lou, Yan Huang, Zhihe Lu, et al. Efficient-wam: A 1b-parameter world-action model with low-cost future imagination. *arXiv preprint arXiv:2606.10040*, 2026b.

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

Ziang Li, Dongzhou Cheng, Yibin Wang, Shiyue Wang, Xiaoyang Xu, Lingxuan Weng, Juan Wang, and Jiaqi Wang. Light-wam: Efficient world action models with state-fusion action decoding. *arXiv preprint arXiv:2606.08242*, 2026d.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, 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. *arXiv preprint arXiv:2508.05635*, 2025. URL https://arxiv.org/abs/2508.05635.

Xuewu Lin, Tianwei Lin, Yun Du, Hongyu Xie, Yiwei Jin, Jiawei Li, Shijie Wu, Qingze Wang, Mengdi Li, Mengao Zhao, Ziang Li, Chaodong Huang, Hongzhe Bi, Lichao Huang, and Zhizhong Su. HoloBrain-0 technical report, 2026. URL https://arxiv.org/abs/2602.12062.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

NVIDIA Cosmos Team. Cosmos 3: Omnimodal world models for physical ai. Technical report, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

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, 2026. URL https://arxiv.org/abs/2602.10098.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models, 2025. URL https://arxiv.org/abs/2505.17016.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Wan Team, 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, 2025. URL https://arxiv.org/abs/2503.20314.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: Vla foundation model for robotic manipulation with action manifold learning, 2026. URL https://arxiv.org/abs/2602.11236.

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.

Maeve Zhang, Rain Sun, Xiang Wang, Cyril Zhang, Shalfun Li, Meng Cao, Howard Lu, Ethan Chen, Harry Jhou, KZ Zheng, et al. Wall-ss: Scaling long-horizon world models via next-scale autoregression. *arXiv preprint arXiv:2608.26239*, 2026a.

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. *arXiv preprint arXiv:2603.22078*, 2026b.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model, 2025. URL https://arxiv.org/abs/2510.10274.

Shilong Zou, Shilin Zhang, Yingji Zhang, Yuhang Huang, Yi Zhang, Zeyuan Ding, Han Dong, Junwei Liao, Yong Dai, Jian Tang, et al. Pelican-sim 1.0: A general world model simulator for embodied intelligence. *arXiv preprint arXiv:2609.12036*, 2026.

A Appendix

A.1 Implementation Details

Model and residual mid-training. The world backbone is a Wan VideoDiT-5B initialized from its pretrained checkpoint. We first adapt its prediction space from absolute VAE latents to Residual-VAE latents on 100 hours of robotics video drawn from the DROID and BridgeData V2 portions of the Cosmos training corpus. This stage uses the same flow-matching parameterization as the pre-trained model and requires video without action labels. The frozen Wan VAE encoder and decoder are shared by mid-training, ACWM probing, and policy post-training, so all three stages operate in the same latent space, and the first-frame anchor plus the predicted residual recovers the absolute latent used by the decoder.

Action Tangent Field construction. For each embodiment we adapt a 16B action-conditioned world model (NVIDIA Cosmos Team, 2026) following the released post-training recipe, and use it only to sample local futures. For every training window we draw $K = 8$ temporally smooth perturbations in normalized action coordinates and hold the observation, instruction, proprioception, and generation conditions fixed across the reference and perturbed rollouts, so that the measured differences are dominated by the action change rather than by sampling variation. Generated futures are encoded with the same frozen VAE, differenced against the reference future, and normalized by the perturbation magnitude, yielding the response matrix G_t. We orthonormalize its columns to obtain the local basis Q_t and cache Q_t and its rank together with each training window, so the ACWM is never needed during policy training or at inference.

Policy post-training. The policy follows the Fast-WAM architecture: a VideoDiT world branch initialized from the residual-mid-trained backbone supplies world tokens h_t^W to an ActionDiT that predicts action chunks. The world branch is evaluated at $t_v = 0$ during training, matching the single-pass computation used at inference, and receives the ordinary residual loss on every latent element together with the ATF projection loss on the cached tangent directions. We use AdamW with learning rate = 1e-4, and an action chunk horizon of H = 32. LIBERO-Plus models are fine-tuned for 10 epochs on the official demonstration data, RoboTwin and RoboTwin 2.0-Plus share a single unified multi-task model trained for 5 epochs, real-world policies are trained separately per task for 10 epochs on 8 NVIDIA A100 GPUs with a per-GPU batch size of 1.

Inference. Each replanning step performs one VideoDiT evaluation and four ActionDiT Euler steps, so the world predictor is evaluated exactly once and no video denoising or VAE decoding is required at deployment. The ACWM and the cached tangent basis contribute only through post-training and are not loaded online.

A.2 Detail Results on LIBERO-Plus

Table 4 reports success for each of the 40 original tasks after aggregating its LIBERO-Plus perturbation instances. The overall rate is 87.78% across 10,030 evaluated instances. LIBERO-Object and LIBERO-Spatial reach 92.45% and 93.26%, respectively, while LIBERO-Goal is lower at 78.66%. The largest task-level drop is opening the middle cabinet drawer, with 29 successes in 268 instances. This unevenness shows that the strong aggregate result does not extend equally to every goal and scene configuration.

A.3 Realworld Tasks

Platforms. We conduct real-world experiments on two complementary robotic platforms. NERO uses parallel grippers and represents a conventional low-dimensional manipulation setting, whereas Tianji is equipped with Wuji dexterous hands, each with 20 actively controlled degrees of freedom. Both platforms use three RGB views, including one head-mounted camera and two hand-mounted cameras. The two platforms jointly cover single-arm and bimanual manipulation with substantially different end effectors and action spaces. We collect 200 demonstrations for each task.

Two tasks for the Nero platform.

- **Test-tube insertion.** The robot picks up two test tubes one by one and inserts them into designated slots of a test-tube rack. Each correctly inserted test tube contributes 50 points to the task score.
- **Bowl stacking.** The robot sequentially picks up three bowls and stacks them into a stable vertical pile. The task contains two stacking steps, each contributing 50 points to the total score.

Three tasks for the Tianji platform.

- **Language-conditioned ordered object transfer.** Three objects are initially placed in a container. Given a language instruction specifying their order, the robot must pick up the objects one by one and transfer them from the container into the pot in the instructed sequence. Each correctly ordered transfer contributes one third of the total score, while an object transferred out of order receives no credit.
- **Whiteboard wiping.** The robot picks up an eraser, establishes contact with the whiteboard, and wipes the written notes from the board. The task score assigns 20 points for successfully picking up the eraser, up to 60 points according to the erased area, and 20 points for lifting the hand away after wiping. The task is considered successfully completed when the notes on the whiteboard have been erased.
- **Bimanual cooking.** The robot performs a coordinated bimanual manipulation task in which one hand opens and holds the pot lid while the other hand picks up a spoon, scoops objects, and transfers them into the pot. The task is divided into four stages—lifting the lid, lifting the spoon, successfully scooping the objects, and depositing them into the pot—with each stage contributing 25 points.

Evaluation protocol. Each policy is evaluated under four conditions, three of which perturb the nominal setup, and every task–condition pair is repeated three times. Each trial has a two-minute time limit. Because several tasks contain multiple meaningful stages, we report a progress score in $[0, 100]$ instead of a binary success rate. With three trials per cell, one unlucky rollout moves a cell average by 33 points, so Tables 6 and 7 list the individual trials rather than only their mean.

The two platforms are perturbed along different axes, because their failure modes differ. On Tianji we use the standard setup, red illumination, green illumination, and a replaced tabletop background, which target appearance and colour changes while the scene layout is held fixed. On NERO we use the standard setup, two rearrangements of the object and rack placement (*Layout 1* and *Layout 2*), and a simultaneous red and green illumination change, which target geometric variation together with colour shift. Fast-WAM was not evaluated on the spoon task, and its entries for that task are omitted.

Table 4: Success rates for the 40 original tasks under LIBERO-Plus perturbations. Task descriptions are abbreviated. Success/total counts one trial per perturbation instance; for the 264 repeated instances in LIBERO-Goal, we retain the result with the latest start time. Suite aggregates are weighted by the number of evaluated instances.

Original task	Success / total	Success rate (%)
LIBERO-10		
Stove on; moka pot on stove	243 / 281	86.48
Black bowl in bottom drawer; close drawer	277 / 304	91.12
Yellow-and-white mug in microwave; close it	232 / 278	83.45
Both moka pots on stove	194 / 233	83.26
Alphabet soup and cream cheese box in basket	186 / 194	95.88
Alphabet soup and tomato sauce in basket	204 / 212	96.23
Cream cheese box and butter in basket	201 / 210	95.71
White mug on left plate; yellow-and-white mug on right plate	236 / 287	82.23
White mug on plate; chocolate pudding to its right	201 / 283	71.02
Book in back compartment of caddy	224 / 237	94.51
Suite aggregate	2198 / 2519	87.26
LIBERO-Goal		
Open middle drawer of cabinet	29 / 268	10.82
Open top drawer; put bowl inside	264 / 324	81.48
Push plate to front of stove	247 / 270	91.48
Put bowl on plate	208 / 235	88.51
Put bowl on stove	133 / 156	85.26
Put bowl on top of cabinet	209 / 243	86.01
Put cream cheese in bowl	247 / 291	84.88
Put wine bottle on rack	263 / 302	87.09
Put wine bottle on top of cabinet	225 / 270	83.33
Turn on stove	213 / 232	91.81
Suite aggregate	2038 / 2591	78.66
LIBERO-Object: place the named object in a basket		
Alphabet soup	213 / 242	88.02
BBQ sauce	256 / 292	87.67
Butter	258 / 273	94.51
Chocolate pudding	237 / 257	92.22
Cream cheese	163 / 173	94.22
Ketchup	205 / 214	95.79
Milk	236 / 272	86.76
Orange juice	265 / 280	94.64
Salad dressing	254 / 264	96.21
Tomato sauce	241 / 251	96.02
Suite aggregate	2328 / 2518	92.45
LIBERO-Spatial: place the black bowl on a plate		
Between plate and ramekin	220 / 231	95.24
Table center	260 / 267	97.38
Top drawer of wooden cabinet	250 / 280	89.29
Next to cookie box	226 / 232	97.41
Next to plate	177 / 180	98.33
Next to ramekin	183 / 198	92.42
On cookie box	240 / 246	97.56
On ramekin	237 / 256	92.58
On stove	251 / 269	93.31
On wooden cabinet	196 / 243	80.66
Suite aggregate	2240 / 2402	93.26
Overall	**8804 / 10030**	**87.78**

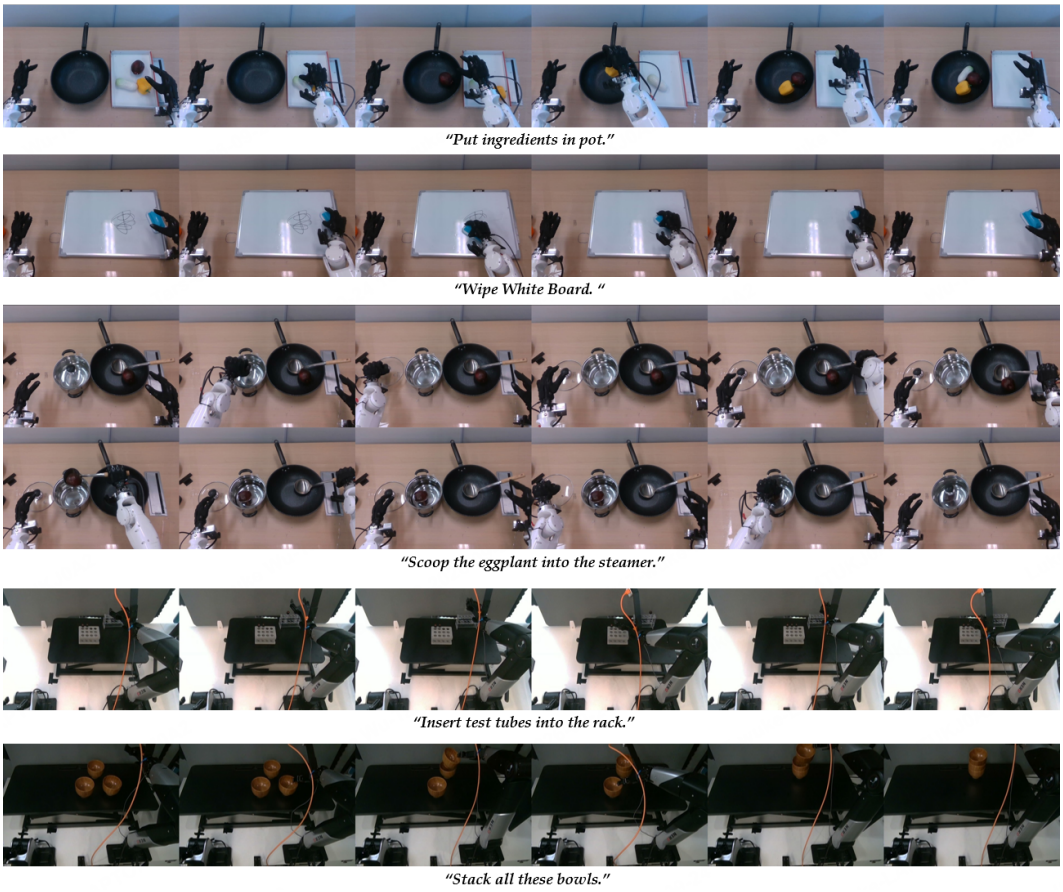


Figure 5: **Real-world task suite.** Top row: the three Tianji tasks, language-conditioned ordered object transfer, whiteboard wiping, and bimanual cooking with lid holding and spoon scooping. Bottom row: the two NERO tasks, test-tube insertion and bowl stacking. Each task is shown in its initial configuration together with the target configuration.

Table 5: Detailed results under different perturbations on LIBERO-Plus.

Model	Original	Camera	Robot	Lang.	Light	BG	Noise	Layout	Total
Ours	**98.15%** 1535/1564	**60.97%** 414/679	**96.24%** 1458/1515	**91.15%** 1401/1537	**86.78%** 991/1142	**94.09%** 780/829	**96.44%** 1544/1601	**58.38%** 679/1163	**87.76%** 8802/10030

Table 6: **Per-trial real-world scores on the three Tianji tasks.** *Mult-PnP* is the language-conditioned ordered object transfer, *Wipe Board* is whiteboard wiping, and *Use Spoon* is the bimanual cooking task. Each entry is the task progress score in $[0, 1]$ obtained from a single rollout. Conditions denote the visual perturbation applied, with *New Background* replacing the tabletop surface. Fast-WAM was not evaluated on the spoon task.

Method	Condition	Move Ingredients			Wipe Board			Cook Meal		
		Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3
$\pi_{0.5}$ (2025a)	Vanilla	0.6	0.6	1	0.7	0.7	0.3	0	0.25	0
	Red Light	0	0.6	1	0.5	0.6	0.4	0	0	0
	Green Light	0	0	0.3	0	0.2	0.6	0	0	0
	New Background	0	0	0.3	0	0.3	0	0	0	0
Fast-WAM (2026)	Vanilla	0.6	0.3	0.3	0.5	0.4	0.4	–	–	–
	Red Light	0.3	0	0.3	0.3	0.4	0.4	–	–	–
	Green Light	0.3	0	0	0.6	0.5	0.3	–	–	–
	New Background	0	0	0.3	0.2	0.2	0.5	–	–	–
ΔWAM	Vanilla	0.6	0.3	1	0.7	1	0.7	0.25	1	0
	Red Light	0.6	1	0.6	0.4	0.6	0.6	0.25	0.75	0.25
	Green Light	0.6	0.6	0.6	0.2	0.6	0.2	0.25	0.25	0.25
	New Background	0.6	0.6	0.6	0.2	0.2	0.2	0.25	0.75	0.25

Figure 5 shows the initial and target configurations of the five tasks. The two NERO tasks are single-arm and gripper-based: test-tube insertion requires inserting both tubes into their slots, and bowl stacking requires placing all three bowls into a stable pile. The three Tianji tasks use both arms and the Wuji hands. Ordered object transfer is the only task whose instruction changes the required behaviour, since the target sequence is specified in language rather than fixed by the scene. Whiteboard wiping and cooking both require sustained contact, either between the eraser and the board surface or between the spoon and the contents of the container, so success depends on force regulation rather than on reaching a target pose alone.

The task suite is designed so that the two platforms are not simply two instances of the same manipulation problem. NERO provides geometrically precise but low-dimensional manipulation with a rigid gripper, where the main difficulty is alignment under tight tolerances. Tianji provides a high-dimensional hand action space with bimanual coordination, where the main difficulty is contact regulation and sequencing. A supervision signal that only helps on one of these regimes would be of limited interest, so we report both platforms separately rather than pooling them into a single average.

Tables 6 and 7 report the raw per-trial scores behind the real-world comparison. We separate the Tianji tasks from the two single-arm NERO tasks because they use different perturbation sets, as described above. Within each block, rows group trials by condition and columns separate the individual trials, so that a consistently high score can be distinguished from one driven by a single successful attempt. The variation within a condition is substantial for all three methods: identical setups differ by up to a full task stage between trials, which is why the analysis below is based on the per-condition averages rather than on individual cells.

Three patterns are visible across both tables, and we report them as they appear rather than summarising them into a single claim.

First, the perturbed conditions are consistently harder than the standard setup, and no method is unaffected by them. On the Tianji tasks all three methods lose accuracy under green illumination and under background replacement, which are the conditions that most strongly change the appearance of the object or the surface it rests on. On NERO the effect is visible but less uniform, since the illumination change is applied jointly with the layout variation in one of the four conditions.

Table 7: **Per-trial real-world scores on the two NERO tasks.** *Plug Tubes* is test-tube insertion and *Stack Bowls* is bowl stacking. Each entry is the task progress score in $[0, 1]$ from a single rollout. *Layout 1* and *Layout 2* rearrange the object and rack placement, and *Red & Green Light* applies both illumination changes simultaneously.

Method	Condition	Stack Bowls			Plug Tubes		
		Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3
$\pi_{0.5}$ (2025a)	Vanilla	0	0.5	0.5	1	1	0.5
	Layout 1	0.5	0	0.5	1	0	0.5
	Layout 2	0	0	0.5	0	1	0.5
	Red & Green Light	1	0	0.5	0	1	0.5
Fast-WAM (2026)	Vanilla	0	0.5	1	0.5	0.5	0.5
	Layout 1	1	0.5	0	1.0	0.5	0
	Layout 2	0.5	0	0	0.5	0.5	0
	Red & Green Light	0	0	0.5	1	0.5	0
ΔWAM	Vanilla	0.5	0.5	1	1	1	1
	Layout 1	0.5	0.5	0.5	1	0.5	1
	Layout 2	1	0.5	0	0.5	1	0.5
	Red & Green Light	0	0.5	1	0.5	0.5	1

Second, the variation between trials within a single condition is comparable to the difference between methods on several tasks. This is most visible in the bowl stacking and test-tube insertion columns, where individual trials of the same method span the full range from zero to complete success. The per-condition averages are therefore the appropriate quantity to compare, and the individual trials are reported mainly to make that variance explicit rather than to suggest that any single cell is reliable on its own.

Third, the relative ordering of the three methods is not identical on the two platforms. On the Tianji tasks, which involve tool use and sustained contact with a 20-DoF hand action space, the ordering across conditions is more stable than on the NERO tasks, which involve precise placement with a gripper and are dominated by a small number of discrete stages. We therefore do not claim a uniform improvement on every task, and instead read these results together with the simulation evidence, where the effect of ATF supervision can be measured over many more trials and where the perturbation axes are controlled individually.