PICA-WAM: VISUAL PERTURBATION INVARIANCE VIA CROSS-APPEARANCE SELF-ALIGNMENT IN WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) jointly learn to predict future video and to generate actions, yet they remain brittle under visual perturbations: task-irrelevant changes in how a scene is observed, which alter what it *looks like* but not what the robot *should do*. We study a gap in conventional WAM training: the video prediction objective requires faithful reconstruction of appearance details, while consistent action generation under task-irrelevant appearance changes is not explicitly enforced. Training on augmented observations exposes a WAM to such changes but never ties the representation of a perturbed observation to that of its nominal counterpart, and aligning to external pretrained encoders imports features that are not themselves invariant to task-irrelevant appearance. To address this gap, we propose PICA-WAM, a self-supervised training framework that casts perturbation invariance as an auxiliary cross-appearance self-alignment task for WAMs. An exponential-moving-average copy of the WAM reads the nominal observation at a slightly lower noise level and provides a later-layer target; the online WAM reads a perturbed view of the same state and aligns its earlier-layer action representation to this target through a lightweight projection head. Across three WAMs on LIBERO-Plus, RoboTwin 2.0, and a real robot, PICA-WAM consistently improves robustness under visual perturbations, preserves nominal performance in simulation, and raises the real-robot success rate from 33.1% to 67.5%. PICA-WAM further reduces the sensitivity of video features to task-irrelevant appearance and its propagation into actions, as shown by paired layer-wise diagnostics.

1 INTRODUCTION

World Action Models (WAMs) build robot policies on top of pretrained video generators, coupling predictions of how a scene will evolve with the actions that drive it (Yuan et al., 2026b; Li et al., 2026b; Ye et al., 2026b; Wang et al., 2026b; Cai et al., 2026b). Video–action co-training transfers the dynamics prior of large-scale video pretraining into control, and WAMs now approach saturation on standard manipulation benchmarks such as LIBERO (Liu et al., 2023) and RoboTwin (Mu et al., 2024; 2025; Chen et al., 2025). Yet these models remain brittle when observations depart from the training distribution (Hong et al., 2026). On LIBERO-Plus (Fei et al., 2025) and RoboTwin 2.0 Clean-to-Random (Yuan et al., 2026a), shifts in appearance, viewpoint, sensing, and scene configuration can substantially reduce success. The task goal remains the same, but the policy must act under unfamiliar observations and starting conditions.

We view this brittleness as a gap in the training objectives of WAMs. The video prediction objective requires faithful reconstruction of appearance details such as lighting and texture, while reliable control calls for consistent actions when the same task state is viewed under different appearances, and joint video–action training does not explicitly enforce this consistency. As the action expert reads video-stream features, task-irrelevant appearance variation leaks into the action pathway.

Existing approaches address this gap only partially. *(i) Data augmentation* (Tobin et al., 2017; Kostrikov et al., 2020) exposes the model to perturbed observations but never requires the representation of a perturbed observation to agree with that of its nominal counterpart. The resulting invariance is incidental rather than enforced: in our layer-wise analysis, augmentation lowers appearance

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.

sensitivity in intermediate video layers, but sensitivity rises above the unaugmented baseline in layers 21–25 (Figure 2, left). *(ii) External representation alignment* regularizes WAM features toward priors from outside the model, such as semantic vision encoders (Qiu et al., 2026; Yan et al., 2026; Wang et al., 2026a), 3D trajectory fields (Yang et al., 2026b), or rendered 3D Gaussian fields (Zhang et al., 2026b). These priors provide semantic and geometric guidance, but alignment to an external teacher does not by itself enforce agreement between different appearances of the same task state. This leaves a need for explicit cross-appearance supervision. *(iii) Self-alignment* constructs representation targets from the generative model itself, without an external pretrained encoder. In image and video generation, Jiang et al. (2025a) and Chefer et al. (2026) align shallow, noisier features to deeper, cleaner features of an exponential-moving-average (EMA) copy of the model (Tarvainen & Valpola, 2017; Grill et al., 2020), matching the gains of alignment to external encoders (Yu et al., 2024). These methods align differently noised versions of the same sample, without explicitly matching nominal and perturbed appearances of the same task state. Self-alignment also remains underexplored in WAMs, where joint video–action prediction raises the further question of whether to align the video stream, the action stream, or both.

To this end, we propose **PICA-WAM**, a self-supervised training framework that promotes **P**erturbation **I**nvariance via an auxiliary **C**ross-**A**ppearance self-alignment task in **WAM**s. An EMA copy of the WAM reads the nominal observation at a slightly lower noise level and provides a later-layer target; the online WAM reads a perturbed view of the same state and aligns its earlier-layer action representation to this target through a lightweight projection head (Figure 1). Each pair thus spans three gaps. The *appearance gap* supplies the invariance signal; the *noise-level* and *layer* gaps, which Jiang et al. (2025a) identified as key ingredients of self-alignment, make the target stronger than the student it supervises. We apply the objective to the action stream, which directly encourages consistency in the representation used for control while leaving the video prediction objective unchanged; in our ablations, this choice outperforms aligning the video stream or both streams. The teacher and the head are discarded after training, so PICA-WAM adds no parameters and no computation at inference and applies to any WAM with a layered backbone.

We evaluate PICA-WAM on three WAM backbones across simulation and real-world settings: Fast-WAM (Yuan et al., 2026b) on LIBERO-Plus, LingBot-VA (Li et al., 2026b) on RoboTwin 2.0 Clean-to-Random, and OpenWAM-α (Wang et al., 2026b) on a real robot. PICA-WAM consistently improves robustness over the corresponding baselines without sacrificing in-distribution performance, and, without embodied pretraining, achieves success rates on LIBERO-Plus comparable to those of models with large-scale embodied pretraining. Paired analyses further show that it reduces both the appearance sensitivity of video features and its propagation into actions, including under perturbation families unseen during training.

In summary, our main contributions are as follows:

- We propose PICA-WAM, a self-supervised training framework for WAMs in which the action representation from a perturbed observation is aligned to a cleaner, deeper target computed from the nominal observation by an EMA copy of the model, needs no external encoder, no additional data, and no inference-time computation.
- We design a diagnostic protocol for WAMs that compares nominal and perturbed views of the same state at three levels: layer-wise feature sensitivity normalized by distances between other states of the same episode, layer-band interventions that swap perturbed video features into a nominal prediction to trace their effect on the action, and the drift of executed trajectories. With this protocol, we show that augmentation alone leaves late video layers sensitive to task-irrelevant appearance and lets this sensitivity reach the action, whereas PICA-WAM reduces both.
- We demonstrate consistent robustness gains on LIBERO-Plus, RoboTwin 2.0 Clean-to-Random, and real-world manipulation under visual perturbation across three WAMs with and without embodied pretraining, while preserving in-distribution performance.

2 Related Work

World action models. Early work synthesizes future frames and infers actions from them (Du et al., 2023), or conditions a policy on video pretraining and predictive representations (Wu et al., 2024; Hu et al., 2024). Later models jointly denoise video and actions (Zhu et al., 2025; Liao et al.,

2026; Kim et al., 2026), and current WAMs scale this recipe with stronger video backbones, action experts, and embodied pretraining (Yuan et al., 2026b; Li et al., 2026b; Wang et al., 2026b; Ye et al., 2026b; Cai et al., 2026b; Chen et al., 2026b;a; Guo et al., 2026). Yet WAMs remain brittle under visual perturbations (Fei et al., 2025; Yuan et al., 2026a), and their predicted futures can hallucinate training-domain content (Wang et al., 2026a). Remedies include inference-time activation steering (Hong et al., 2026) and training-time priors from outside the model: semantic features from pretrained encoders such as DINOv3 and V-JEPA or from text (Qiu et al., 2026; Yan et al., 2026; Wang et al., 2026a; Liu et al., 2026a; He et al., 2026), geometric supervision such as 3D trajectory and Gaussian fields (Ma et al., 2026; Yang et al., 2026b; Zhang et al., 2026b), and distractor-invariant object slots (Liu et al., 2026b). Data augmentation and domain randomization (Tobin et al., 2017; Kostrikov et al., 2020) expose a policy to appearance variation but supervise each observation independently. Self-Guided WAM (Zhao et al., 2026) uses an EMA teacher, but to supervise future dynamics rather than cross-appearance agreement. None of these methods constrains how WAM representations change across appearances of the same task state; PICA-WAM instead casts perturbation invariance as an cross-appearance self-alignment task, without any external encoder.

Representation alignment in generative models. REPA (Yu et al., 2024) aligns intermediate diffusion-transformer features with a pretrained visual encoder to accelerate training and improve generation, and follow-up work extends it to end-to-end VAE tuning (Leng et al., 2025) and video generation (Zhang et al., 2026a); several of the semantic WAM methods above follow the same recipe. Such targets inherit the properties of the external encoder: frozen pretrained encoders, including DINO- and SigLIP-style encoders, are strongly appearance-coupled, and the action information decodable from their features collapses under noise and blur (Yeom et al., 2026). Self-alignment instead constructs targets within the model. SRA (Jiang et al., 2025a) aligns earlier-layer features of a student at a higher noise level with later-layer features of an EMA teacher at a lower noise level; Self-Flow (Chefer et al., 2026) noises the student's tokens at heterogeneous levels while the EMA teacher observes the cleaner input, scaling self-alignment to image, video, and audio generation. These approaches train a student to agree with a slowly updated teacher on a different view of the same input, but their views differ in noise level or network depth, not in task-irrelevant appearance, so the agreement does not encode appearance invariance. PICA-WAM makes appearance a view difference: the teacher reads the nominal view and the student a perturbed view of the same state.

3 METHOD

3.1 PRELIMINARIES: WORLD ACTION MODELS

A WAM takes the current observation and a language instruction c and jointly predicts future video and a chunk of actions. We consider the common two-stream design (Yuan et al., 2026b; Li et al., 2026b; Wang et al., 2026b) (Figure 1a): a video stream initialized from a pretrained video diffusion transformer and an action stream are coupled layer by layer through joint attention, in the style of a mixture of transformers. Let $\boldsymbol{z}$ be the latent video clip, whose first frame encodes the current observation, and let $\boldsymbol{a}$ be the action chunk. The two streams are trained jointly under a flow-matching objective (Lipman et al., 2022; Liu et al., 2022). The future frames are noised as $\boldsymbol{z}_\sigma = (1-\sigma)\boldsymbol{z}+\sigma\boldsymbol{\epsilon}$ while the first frame is kept clean, and the actions as $\boldsymbol{a}_\tau = (1-\tau)\boldsymbol{a} + \tau\boldsymbol{\eta}$, with $\boldsymbol{\epsilon}, \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The model θ predicts the velocities v of both streams, and the generative loss is the sum of a video term and an action term:

$$\begin{aligned}\mathcal{L}_{\text{gen}} &= \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}} \\ &= \mathbb{E}\Big[\big\|v_\theta^{\text{vid}}(\boldsymbol{z}_\sigma, \boldsymbol{a}_\tau, c) - (\boldsymbol{\epsilon} - \boldsymbol{z})\big\|^2 + \big\|v_\theta^{\text{act}}(\boldsymbol{z}_\sigma, \boldsymbol{a}_\tau, c) - (\boldsymbol{\eta} - \boldsymbol{a})\big\|^2\Big],\end{aligned} \tag{1}$$

where the video term is taken over the future frames. We write $\boldsymbol{h}^{(l)}(\boldsymbol{z}_\sigma, \boldsymbol{a}_\tau; \theta) \in \mathbb{R}^{K\times d}$ for the hidden states of the K action tokens at layer l of the action stream. Because action tokens attend to video tokens at every layer, $\boldsymbol{h}^{(l)}$ depends on the observation and predicted future through the video stream, and so does any task-irrelevant appearance variation in it. Unless stated otherwise, "appearance" below refers to this task-irrelevant variation.

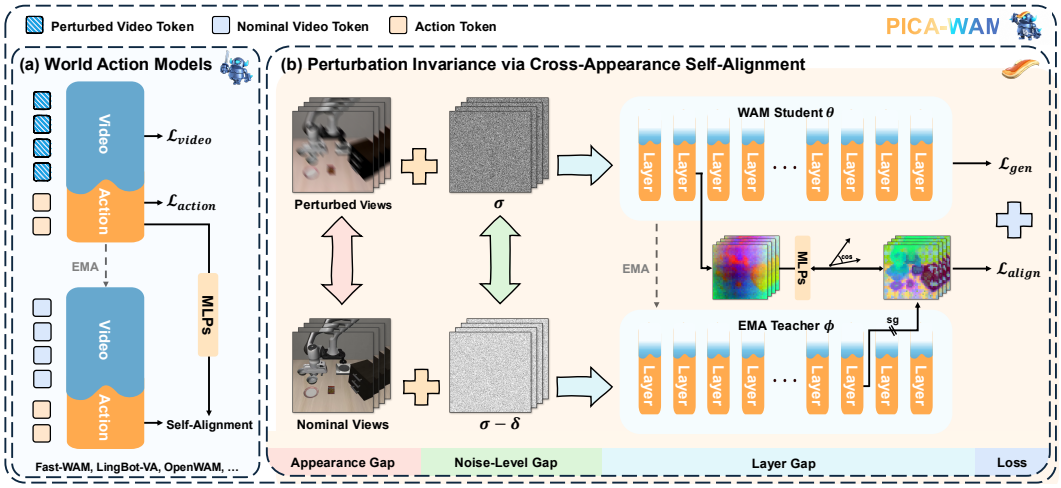

Figure 1: **Overview of PICA-WAM. (a)** A two-stream WAM is trained with $\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}}$; PICA-WAM adds an auxiliary self-alignment task between the action streams of the model and its EMA copy. **(b)** The student θ reads a perturbed view at video noise level σ and is trained with $\mathcal{L}_{\text{gen}}$; the EMA teacher ϕ reads the nominal view of the same task state at $\sigma - \delta$. Projected earlier-layer action features of the student are aligned to later-layer action features of the teacher under stop-gradient (sg), giving $\mathcal{L}_{\text{align}}$.

3.2 Action Invariance to Visual Perturbations

Let $\boldsymbol{z}$ be a nominal clip as recorded and $T \in \mathcal{T}$ a visual perturbation, such as a photometric, sensor, or mild geometric change, that alters how the scene is observed but not the task state. Applying T to every frame yields a perturbed clip $\boldsymbol{z}' = T(\boldsymbol{z})$ for which the demonstrated actions $\boldsymbol{a}$ remain correct. We seek action representations that are invariant to such perturbations:

$$\boldsymbol{h}^{(l)}(\boldsymbol{z}'_\sigma, \boldsymbol{a}_\tau; \theta) \approx \boldsymbol{h}^{(l)}(\boldsymbol{z}_\sigma, \boldsymbol{a}_\tau; \theta), \qquad \forall\, T \in \mathcal{T}. \tag{2}$$

We impose this property on the action stream only. The video stream must retain appearance to predict the future frames of $\boldsymbol{z}'$, so requiring invariance of the whole representation would conflict with the video objective. The action stream is the representation from which actions are decoded, and constraining it still shapes the video features it reads through joint attention (Section 4.2).

Training on perturbed clips, i.e., minimizing $\mathbb{E}_T\big[\mathcal{L}_{\text{gen}}(T(\boldsymbol{z}), \boldsymbol{a})\big]$, is equivalent to augmentation-only training, and it does not impose Eq. equation 2. Specifically, in this objective, each clip is a separate training example, and the loss scores only the predictions for its future frames and its actions. It never scores the intermediate representations, and it never considers a perturbed clip together with the nominal clip of the same state. Whether the two clips are represented alike is therefore not a training target, and any invariance that emerges is a by-product. Two properties of WAMs work against this by-product. First, the video objective requires faithful reconstruction of the perturbed future frames, so the video stream must encode the appearance introduced by T. Second, the action tokens read these video features through joint attention at every layer, so this appearance variation enters the action stream. The action objective penalizes this variation only when it changes the predicted actions on the training clips and never removes it from the intermediate features. Section 4.2 examines this layer by layer. Imposing Eq. equation 2 thus requires supervision on the representation itself that relates each perturbed clip to the nominal clip of the same state. We provide it by computing, from the nominal clip, a target for $\boldsymbol{h}^{(l)}(\boldsymbol{z}'_\sigma, \boldsymbol{a}_\tau; \theta)$, as described next.

3.3 Self-Alignment across Appearances

We obtain the target through *cross-appearance self-alignment* (Figure 1b): the WAM computes a target from the nominal clip, and its representation of the perturbed clip is aligned to that target.

Student and teacher. The student is the online model θ. It reads the perturbed clip at video noise level σ, and its layer-m action features are mapped by a lightweight MLP projection head g_ψ. The

teacher is an EMA copy of the model, $\phi \leftarrow \mu\phi + (1-\mu)\theta$. It reads the nominal clip at a lower video noise level $\sigma - \delta$ with $\delta > 0$, and provides action features from a later layer $n > m$. Both branches differ only in the observed clip and its future video noise level. The alignment loss is the negative cosine similarity per action token, where $\mathrm{sg}[\cdot]$ denotes stop-gradient:

$$\mathcal{L}_{\text{align}} = -\frac{1}{K}\sum_{k=1}^{K} \cos\Big(g_\psi\big(\boldsymbol{h}_k^{(m)}(\boldsymbol{z}'_\sigma, \boldsymbol{a}_\tau; \theta)\big),\ \mathrm{sg}\big[\boldsymbol{h}_k^{(n)}(\boldsymbol{z}_{\sigma-\delta}, \boldsymbol{a}_\tau; \phi)\big] \Big), \tag{3}$$

Three gaps. Each pair spans three gaps between student and teacher (bottom of Figure 1b). The *appearance gap* ($\boldsymbol{z}'$ versus $\boldsymbol{z}$) supplies the invariance signal. Student and teacher share the task state, the noisy actions, and the instruction, and their inputs differ in appearance only through T. Matching the teacher target therefore requires the action features of the student to discard what T changes, which is the invariance of Eq. equation 2. The alignment thus provides the supervision relating a perturbed clip to its nominal counterpart that augmentation-only training lacks (Section 3.2). Since only the student receives gradients, the representation of the perturbed clip is pulled toward that of the nominal clip and not the reverse, which matches deployment, where the model receives perturbed observations. The *noise-level* and *layer gaps* make the target stronger than the student's own features: the teacher reads a less noisy input and provides features from a later layer, both of which Jiang et al. (2025a) found to yield better target for alignment. Setting $\boldsymbol{z}' = \boldsymbol{z}$ removes the appearance gap and reduces Eq. equation 3 to SRA-style self-alignment of the action stream.

Training and inference. The overall objective is:

$$\mathcal{L} = \mathcal{L}_{\text{gen}}(\boldsymbol{z}', \boldsymbol{a}) + \lambda\, \mathcal{L}_{\text{align}}, \tag{4}$$

optimized over θ and ψ, with the teacher updated by EMA after each step. The alignment term adds one gradient-free teacher forward pass per step, which can stop at layer n. At inference, the teacher and the projection head are discarded, so PICA-WAM leaves the architecture, the parameter count, and the sampling procedure of the WAM unchanged. The perturbation families, layer choices, and other hyperparameters are listed in Appendix A.

4 Experiments

4.1 Experimental Setup

Simulation Benchmarks. LIBERO-Plus (Fei et al., 2025) evaluates policies trained on the four LIBERO suites (Liu et al., 2023) on 10,030 perturbed tasks along seven dimensions: robot initial state, language instruction, lighting, background texture, sensor noise, object layout and camera viewpoint. RoboTwin 2.0 Clean-to-Random (Chen et al., 2025; Yuan et al., 2026a) trains on 50 clean demonstrations for each of 50 bimanual tasks, recorded with a fixed background, default lighting, no distractors, and a fixed table height. Each task is evaluated for 100 episodes in this clean setting and in a domain-randomized setting that randomizes background textures, lighting, distractor clutter, and table height and uses unseen language instructions. We report the success rates on all dimensions of both benchmarks, following prior work.

Backbones and variants. We apply PICA-WAM to three WAMs: Fast-WAM (Yuan et al., 2026b), which has no embodied pretraining, and post-trained on LIBERO; LingBot-VA (Li et al., 2026b) on RoboTwin 2.0; and OpenWAM-α (Wang et al., 2026b) on a real robot (Section 4.5). We use Fast-WAM in its joint mode, in which the video stream takes a single denoising step before the action is generated from the resulting video features. On LIBERO, we further train three controlled variants of Fast-WAM with the same data, schedule, and inference procedure: *Baseline*, trained on nominal clips; *+Aug*, trained on perturbed clips, and PICA-WAM, which adds the alignment term to +Aug. More details are provided in Appendix A.

4.2 Does Augmentation Enforce Invariance?

To test the argument of Section 3.2 that augmentation-only training leaves intermediate representations unconstrained, we design a paired diagnostic protocol. It compares the three controlled Fast-WAM variants (Section 4.1), which differ only in whether they are trained on perturbed clips and

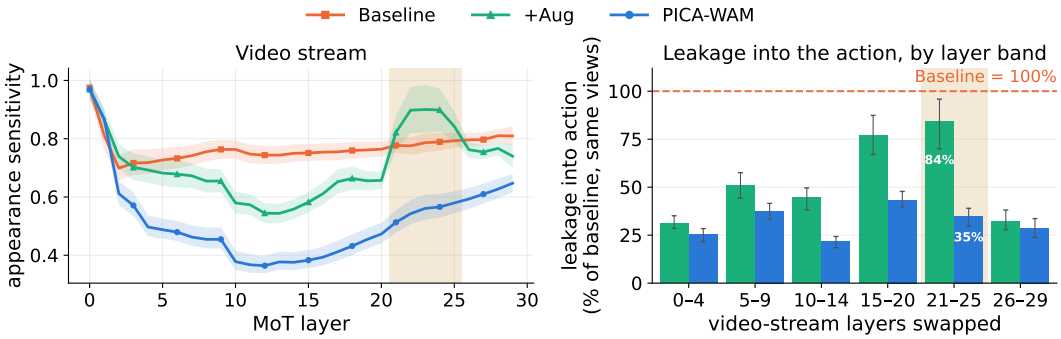

Figure 2: **Video-stream appearance sensitivity and its transfer into actions.** Left: layer-wise sensitivity of video features on unseen perturbation families (Section 4.2). Right: action deviation caused by swapping the perturbed video features of one layer band into a nominal prediction, as a median ratio to that of the Baseline on the same views. In the highlighted layers 21–25, the sensitivity of +Aug rebounds, and these layers retain 84% of the action leakage of the Baseline under +Aug, compared with 35% under PICA-WAM.

whether they are aligned. Each perturbed LIBERO-Plus view is paired with the nominal view of the same simulator state under the same noise seed, so that the two inputs differ only in how the scene is observed. The protocol traces the perturbation from features to executed actions at three levels. (i) *Layer-wise sensitivity* is the feature distance between the two views divided by the distance between the nominal views of two different states in the same episode. Raw feature distances differ in scale across layers and models; normalizing by a change in task state removes this difference, so that a value of 1 means that the perturbation changes the features as much as a change in task state does. (ii) A *layer-band intervention* replaces the video features of one layer band in a nominal prediction with those from the perturbed view, which tests whether the sensitivity remaining in that band shifts the action. (iii) *Trajectory drift* is the distance between the end-effector positions reached by executing the action chunks predicted from the two views. We report separately the *seen* perturbation families, whose type is contained in T (Gaussian, motion, and zoom blur, and fog), and the *unseen* ones (glass blur, lighting, background, and camera viewpoint on LIBERO-Plus). All diagnostics use the single-step video denoising of deployment. Appendix B gives the details.

Augmentation leaves late video layers appearance-sensitive. Figure 2 (left) shows the video-stream sensitivity on unseen families. Relative to the Baseline, +Aug lowers the sensitivity in the intermediate layers, but the sensitivity rebounds in layers 21–25 and exceeds that of the Baseline, so augmentation does not remove appearance variation from the video features that the action stream reads in these layers. PICA-WAM is the least sensitive from layer 2, although it aligns only the action stream: constraining the action stream also shapes the video features through joint attention.

Residual sensitivity reaches the action. Figure 2 (right) reports the layer-band intervention, normalized by the action deviation of the Baseline on the same views. Under +Aug, bands 15–20 and 21–25 carry the largest leakage, and layers 21–25, where the sensitivity rebounds, retain 84% of the leakage of the Baseline. The appearance variation remaining in these features therefore still shifts the action. PICA-WAM lowers the leakage in every band and reduces it to 35% in layers 21–25.

Alignment reduces sensitivity on seen and unseen families. Table 1 reports the sensitivity of both streams at earlier-layer. On both groups, +Aug lowers the video sensitivity of the Baseline only moderately, from 0.610 to 0.530 on seen and from 0.755 to 0.655 on unseen families, whereas PICA-WAM lowers it to 0.369 and 0.455. In the action stream, where the action objective penalizes appearance variation that changes the predicted actions, +Aug lowers the sensitivity of the Baseline from 0.407 to 0.146 and from 0.486 to 0.189, and PICA-WAM further lowers it to 0.057 and 0.075. These reductions over +Aug are of similar size on the two groups, so the effect of the alignment is not confined to the perturbation types contained in T.

Lower sensitivity carries over to executed actions. The drift in Table 1 follows the same pattern. +Aug reduces the drift of the Baseline from 16.14 to 9.41 mm on seen families but only from 16.42 to 11.24 mm on unseen families, i.e., by 42% versus 32%. PICA-WAM reduces it to 5.49 and 6.21 mm, by 66% and 62%, respectively. Appendix B provides further analyses and visualizations, including the executed trajectories, and video prediction fidelity.

Table 1: **Paired Consistency under LIBERO-Plus Visual Perturbations.** Sensitivity is measured at layer 8 of each stream, and drift is the end-effector distance after executing the 32-step action chunks predicted from the paired views; lower is better for all metrics. Metrics and seen/unseen families are defined in Section 4.2. Bold and underline denote the best and second-best values.

	Video Sensitivity ↓		Action Sensitivity ↓		Drift (mm) ↓	
Method	**Seen**	**Unseen**	**Seen**	**Unseen**	**Seen**	**Unseen**
Baseline	0.610	0.755	0.407	0.486	16.14	16.42
+Aug	<u>0.530</u>	<u>0.655</u>	<u>0.146</u>	<u>0.189</u>	<u>9.41</u>	<u>11.24</u>
PICA-WAM (Ours)	**0.369**	**0.455**	**0.057**	**0.075**	**5.49**	**6.21**

Table 2: **Evaluation Results on RoboTwin 2.0 Clean-to-Random.** † denotes a method built on the LingBot-VA backbone. Results of other methods are taken from the RoboTwin 2.0 leaderboard (RoboTwin Team, 2026) or the corresponding papers. Bold and underline mark the best and second-best values within each group. PICA-WAM is ranked within the WAM group.

Method	**Clean**	**Random**	**Avg.**
	VLA		
StarVLA (Ye et al., 2026a)	46.5	3.2	24.8
GR00T-N1.7 (Bjorck et al., 2025)	43.6	20.7	32.2
GalaxeaVLA (Jiang et al., 2025b)	62.7	12.7	37.7
Xiaomi-Robotics-0 (Cai et al., 2026c)	62.9	18.2	40.6
EventVLA (Yang et al., 2026a)	65.6	15.7	40.7
ABot-M0 (Yang et al., 2026c)	57.4	30.4	43.9
X-VLA (Zheng et al., 2026)	68.0	20.9	44.5
Spatial Forcing (Li et al., 2026a)	**77.2**	26.7	52.0
$\pi_{0.5}$ (Intelligence et al., 2025)	<u>70.7</u>	<u>46.0</u>	<u>58.4</u>
GigaBrain-0.7 (Team et al., 2026)	66.8	**67.9**	**67.4**
	WAM		
AHA-WAM (Cai et al., 2026a)	64.3	3.2	33.8
Fast-WAM (Yuan et al., 2026b)	77.8	1.9	39.9
X-WAM (Guo et al., 2026)	70.0	25.8	47.9
LingBot-VA (Li et al., 2026b)	81.2	29.8	55.5
Robust-WAM† (Yan et al., 2026)	81.2	34.4	57.8
OLA-Geo (RoboTwin Team, 2026)	83.0	33.6	58.3
4D-WAM† (Yang et al., 2026b)	81.5	41.8	61.7
WorldScape Policy 2.0 (Su et al., 2026)	<u>84.9</u>	38.5	61.7
CSWAM (Liu et al., 2026a)	84.5	45.2	64.8
OpenWAM-α (Wang et al., 2026b)	**89.4**	<u>48.7</u>	<u>69.0</u>
PICA-WAM (Ours)†	83.1	**57.0**	**70.1**

4.3 Simulation Benchmarks

RoboTwin 2.0 Clean-to-Random. As shown in Table 2. Based on LingBot-VA, PICA-WAM raises the success rate from 29.8% to 57.0% in the randomized setting and from 81.2% to 83.1% in the clean setting. PICA-WAM outperforms Robust-WAM and 4D-WAM, which are built on the same backbone and align to external encoders or 3D representations. It achieves the highest randomized success rate among WAMs and the highest average success rate among all compared methods. In the randomized setting, only GigaBrain-0.7 performs better.

LIBERO-Plus. Table 3 reports the results. Without embodied pretraining, PICA-WAM reaches an average success rate of 85.2% on LIBERO-Plus, exceeding all compared VLAs and WAMs with large-scale embodied pretraining, including $\pi_{0.5}$ (84.4%) and ABot-M0.5 (83.4%), while retaining 98.2% on LIBERO. On the visual perturbation dimensions, it achieves the highest success rate under sensor noise among all methods (94.6%) and the highest under camera-viewpoint changes among WAMs without embodied pretraining (64.7%).

Table 3: **Evaluation Results on LIBERO and LIBERO-Plus.** WAM groups distinguish models with and without large-scale embodied pretraining before LIBERO adaptation. † denotes a method built on the Fast-WAM backbone. Bold denotes the best value and underline the second best within each group. PICA-WAM is ranked within the WAM without Emb. PT. group.

		LIBERO-Plus							
Method	**LIBERO**	**Robot**	**Language**	**Light**	**BG**	**Noise**	**Layout**	**Camera**	**Avg.**
VLA									
π_0 (Black et al., 2024)	94.4	6.0	58.8	85.0	81.4	79.0	68.9	13.8	53.6
OpenVLA-OFT (Kim et al., 2025)	97.1	31.9	79.5	88.7	93.3	75.8	74.2	56.4	69.6
X-VLA (Zheng et al., 2026)	**98.1**	**89.7**	75.7	88.2	**96.0**	62.7	71.8	23.4	71.4
$\pi_{0.5}$ (Intelligence et al., 2025)	96.9	73.6	**80.8**	**96.2**	94.1	**89.0**	**84.5**	**78.4**	**84.4**
WAM with Emb. PT.									
OpenWAM-α (Wang et al., 2026b)	99.3	76.1	88.0	**97.0**	87.1	39.8	77.5	33.8	69.2
LaWAM (Chen et al., 2026a)	98.6	**96.5**	85.2	93.6	**94.0**	68.3	83.5	33.8	77.7
LingBot-VA (Li et al., 2026b)	98.5	93.9	84.1	92.3	80.1	**75.6**	79.3	62.9	80.7
ABot-M0.5 (Chen et al., 2026b)	**99.4**	87.4	**88.6**	94.0	89.7	75.5	**85.2**	**70.5**	**83.4**
WAM without Emb. PT.									
Fast-WAM (Yuan et al., 2026b)	97.6	44.5	68.9	78.2	53.7	37.7	60.7	16.4	51.5
Robust-WAM† (Yan et al., 2026)	97.9	43.4	73.6	90.2	62.6	56.8	67.8	28.7	58.9
4D-WAM† (Yang et al., 2026b)	98.6	64.3	**90.6**	94.3	57.7	69.1	79.2	45.2	71.0
GaussianWAM† (Zhang et al., 2026b)	97.6	61.6	75.4	90.8	58.2	83.0	76.7	54.7	71.3
ST-WAM (Wang et al., 2026a)	**98.7**	60.1	79.3	93.0	74.2	79.5	74.3	55.4	72.8
Self-Guided WAM (Zhao et al., 2026)	98.5	48.9	81.4	89.8	86.1	80.7	74.2	58.6	73.0
SG-WAM (He et al., 2026)	**98.7**	78.3	81.7	**96.5**	**88.9**	90.7	**85.3**	54.8	81.3
PICA-WAM (Ours)†	98.2	**95.4**	86.5	94.1	79.3	**94.6**	82.6	**64.7**	**85.2**

Table 4: **Ablation Results on LIBERO-Plus.** We report success rates (%) on the five dimensions that perturb the visual scene; robot initial state and language are excluded because they perturb the starting pose and the instruction instead. Avg. is the unweighted mean over these five dimensions. +Aug is the augmentation-only variant of Section 4.2; + SRA adds self-alignment with the noise-level and layer gaps but without the appearance gap; the PICA-WAM rows further add the appearance gap and differ in the aligned stream.

Method	**Light**	**BG**	**Noise**	**Layout**	**Camera**	**Avg.**
+ Aug	93.17	60.59	67.21	81.38	45.72	69.61
+ SRA (Jiang et al., 2025a)	94.13	60.59	70.08	82.49	50.78	71.61
+ PICA-WAM$_{\text{align video}}$	93.96	79.00	92.75	82.16	60.04	81.58
+ PICA-WAM$_{\text{align both}}$	**95.36**	76.30	93.57	79.80	58.66	80.74
+ PICA-WAM$_{\text{align action}}$ (Ours)	94.13	**79.28**	**94.63**	**82.56**	**64.67**	**83.05**

4.4 Ablation Studies

Table 4 ablates the alignment on LIBERO-Plus, starting from the +Aug variant of Section 4.2. SRA-style self-alignment, which keeps the noise-level and layer gaps but removes the appearance gap ($z' = z$), raises the average success rate from 69.6% to 71.6%. Adding the appearance gap raises it to between 80.7% and 83.1%, with the largest gains on background, sensor noise, and camera viewpoint. The three PICA-WAM variants differ only in the stream to which Eq. equation 3 is applied. Aligning the action stream performs best (83.1%), ahead of aligning the video stream (81.6%) or both streams (80.7%). This is consistent with Section 3.2: aligning the video stream also constrains the features from which the future frames are predicted, whereas aligning only the action stream leaves the video stream free to encode the appearance that video prediction requires.

4.5 Real-World Experiments

We evaluate on an ARX-Lift2 dual-arm robot with one head camera and two wrist cameras. The eight tasks require the left arm to pick up a banana, a bowl, a cucumber, a computer mouse, an

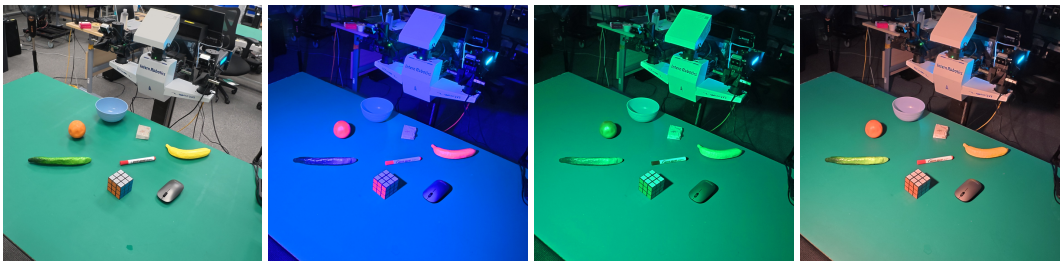

Figure 3: **Real-world evaluation scene under the four lighting conditions.** The photos show the eight task objects on the evaluation tabletop, which differs from the one used for the demonstrations and the bowl, the mouse, and the pen are new instances. The demonstrations are recorded under natural lighting, whereas the blue, green, and orange lights shift the apparent colors of the tabletop and the objects; for example, the banana appears pink under blue light.

Table 5: **Real-World Success Rates per Task** (%). Each task is evaluated in 20 trials, five under each lighting condition; Avg. is over all 160 trials. * marks objects replaced with new instances at evaluation. † denotes a method built on the OpenWAM-α backbone.

Method	Banana	Bowl*	Cucumber	Mouse*	Orange	Pen*	Cube	Tissue	Avg.
$\pi_{0.5}$	85.0	**100.0**	85.0	0.0	0.0	**100.0**	0.0	**75.0**	55.6
OpenWAM-α	5.0	95.0	5.0	25.0	0.0	70.0	0.0	65.0	33.1
PICA-WAM (Ours)†	**95.0**	**100.0**	**90.0**	**30.0**	**75.0**	**100.0**	**10.0**	40.0	**67.5**

orange, a pen, a Rubik's cube, or a tissue, with 25 demonstrations per task. Every trial uses visual perturbations absent from the demonstrations: a new tabletop texture and new instances of the bowl, the mouse, and the pen with different appearances. We evaluate under four lighting conditions: the natural lighting of the demonstrations and three colored lights (blue, green, and orange), as shown in Figure 3. For each lighting condition, we run five trials per task with a different object layout in each trial, and a trial succeeds if the robot picks up the object. We compare PICA-WAM based on OpenWAM-α with OpenWAM-α and $\pi_{0.5}$, all fine-tuned on the same demonstrations.

As shown in Table 5. PICA-WAM raises the success rate of OpenWAM-α from 33.1% to 67.5% and exceeds $\pi_{0.5}$ (55.6%). It improves over OpenWAM-α on seven of the eight tasks, most on the banana, cucumber, and orange tasks, where OpenWAM-α succeeds in at most 5% of the trials and PICA-WAM in at least 75%. These objects are the same as in the demonstrations, so these gains are obtained under the scene-level perturbations, namely the new tabletop and the lighting. On the three objects replaced with new instances, PICA-WAM succeeds in 76.7% of the trials, compared with 63.3% for OpenWAM-α and 66.7% for $\pi_{0.5}$. $\pi_{0.5}$ either succeeds in most trials of a task or fails in all of them, and it fails in every trial of the mouse, orange, and Rubik's cube tasks, whereas PICA-WAM succeeds on every task. It is also the only method that succeeds on the orange and Rubik's cube tasks.

5 Conclusion

We introduced PICA-WAM, a self-supervised training framework that casts perturbation invariance as an auxiliary cross-appearance self-alignment task for WAMs. An EMA copy of the model computes a target from the nominal observation at a lower noise level and a later layer, and the action representation of a perturbed observation is aligned to this target. The teacher and the projection head are discarded after training. Across Fast-WAM, LingBot-VA, and OpenWAM-α, PICA-WAM improves success rates under visual perturbations on LIBERO-Plus, RoboTwin 2.0 Clean-to-Random, and a real robot. Paired diagnostics show that it reduces the appearance sensitivity of both streams and the resulting trajectory drift, including on perturbation families unseen during training.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing and for retrieval and discovery, such as identifying related work. Specifically, large language models were used to edit the text for readability, clarity, and consistency, and to search for related work. We have not used generative AI tools for the other tasks with required disclosure, such as proposing hypotheses, designing experiments, implementing methods, or interpreting results. We have reviewed all AI-assisted work and take responsibility for the final content of this work.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, et al. Aha-wam: Asynchronous horizon-adaptive world-action modeling with observation-guided context routing. *arXiv preprint arXiv:2606.09811*, 2026a.

Jisong Cai, Yao Mu, Ganlin Yang, Zhe Cao, Zhangzheng Tu, Xing Gao, Kailin Li, Xinyu Zhan, Lixin Yang, Yangkun Zhu, Haoxiang Ma, Ming Zhou, Qiaojun Yu, Yufei Xue, Liqun He, Yifei Yao, Yifan Zhu, Long Ling, Bingqi Jiang, Haoyu Guo, Xueyue Zhu, Bowen Zhou, Bin Zhao, Tianfan Xue, Chunhua Shen, and Weinan Zhang. Internw0: A foundational physical world model for efficient real-world interactions, 2026b. URL https://arxiv.org/abs/2609.27656.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026c.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026a.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, et al. Abot-m0. 5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026b.

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.

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

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

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

Junjie He, Junfeng Li, Zhide Zhong, Haodong Yan, Ruixin Li, Yangyang Zheng, Jiaguan Zhu, Tianran Zhang, Yuqiao Du, Wen Chen, et al. Sg-wam: Text-grounded and spatial-aware semantic guidance for world-action models. *arXiv preprint arXiv:2608.08839*, 2026.

Jihoon Hong, Julian Skifstad, Qiyue Dai, Alice Chan, and Glen Chou. Steering robustness into world action models via mechanistic interpretability and optimal control. *arXiv preprint arXiv:2607.14943*, 2026.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

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

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025a.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025b.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. *URL https://arxiv. org/abs/2502.19645*, 2025.

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

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18262–18272. IEEE, 2025.

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. In *International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026a.

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

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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.

Tianbin Liu, Jian Zhu, Taiyi Su, Jianjun Zhang, Chong Ma, Zitai Huang, and Yi Xu. Cswam: Better causal semantic representations for out-of-distribution generalization in world action models. *arXiv preprint arXiv:2609.18462*, 2026a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. Oa-wam: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026b.

Fulong Ma, Daojie Peng, Wenjun Yue, Jiahang Cao, Bintao Wang, Qiang Zhang, and Jun Ma. Geosem-wam: Geometry-and semantic-aware world action models. *arXiv preprint arXiv:2606.03188*, 2026.

Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision*, pp. 264–273. Springer, 2024.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660. IEEE, 2025.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models. *arXiv preprint arXiv:2606.12217*, 2026.

RoboTwin Team. RoboTwin 2.0 Leaderboard. `https://robotwin-platform.github.io/leaderboard`, 2026. Accessed: 2026-09-26.

Haisheng Su, Zongdai Liu, Xin Jin, Haoxuan Dou, Chengming Hu, Baorun Li, Zhanwang Liu, Ruiyan Xu, Jianjie Fang, Xin Zhang, et al. Worldscape policy 2.0: Empowering steerable world action modeling with reasoning-augmented memory. *arXiv preprint arXiv:2607.18840*, 2026.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

GigaBrain Team, Angen Ye, Axiang Sun, Can Jin, Chenxi Cheng, Chong Shi, Dengke Shang, Dingqian Zhang, Guan Huang, Guangqiang Wang, et al. Gigabrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture. *arXiv preprint arXiv:2608.15875*, 2026.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Mingxin Wang, Bin Hu, Bin Qian, Kaitao Jiang, Haoning Wu, Feng Yan, Bowen Jing, Ruiyang Hao, Enyi Wang, Kangning Niu, et al. St-wam: Semantic-temporal world action model for robust manipulation under visual distribution shifts. *arXiv preprint arXiv:2607.28993*, 2026a.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, et al. Openwam: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026b.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Haodong Yan, Junfeng Li, Junjie He, Zhide Zhong, MingMing Yu, Wenxuan Song, Jiaguan Zhu, Yangyang Zheng, Yuqiao Du, Jiadi You, et al. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models. *arXiv preprint arXiv:2608.05903*, 2026.

Ganlin Yang, Zhangzheng Tu, Yuqiang Yang, Sitong Mao, Junyi Dong, Tianxing Chen, Jiaqi Peng, Jing Xiong, Jiafei Cao, Jifeng Dai, et al. Eventvla: Event-driven visual evidence memory for long-horizon vision-language-action policies. *arXiv preprint arXiv:2606.20092*, 2026a.

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

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026c.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-$alpha$: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026a.

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

Jewon Yeom, Hanseul Kim, Jeongjae Park, Sungmok Jung, Jaejin Lee, and Taesup Kim. What makes video world model latents action-relevant: Prediction over reconstruction. *arXiv preprint arXiv:2606.07687*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a.

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

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. *Advances in Neural Information Processing Systems*, 38:122647–122676, 2026a.

Zijian Zhang, Yuqing Jiang, Weitao Zhou, Minglei Li, Jinhao Zhang, Yao Mu, Xiaofan Li, Hao Zhao, and Haibao Yu. Gaussianwam: Distilling geometry and semantics from 3d gaussian fields into world-action models. *arXiv preprint arXiv:2608.24714*, 2026b.

Ruiteng Zhao, Zhengshen Zhang, Yue Su, Wenshuo Wang, Jiahui Li, Zhiyuan Yang, Francis EH Tay, Marcelo H Ang Jr, and Haiyue Zhu. Sg-wam: Self-guided world modeling in geometry-aware policy space. *arXiv preprint arXiv:2608.01397*, 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

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. *arXiv preprint arXiv:2504.02792*, 2025.

A Implementation Details

Perturbation distribution. T is applied to each camera independently, and its parameters are sampled once per clip and shared by all frames of that camera. For each camera, T applies photometric transforms only, additive noise only, or both, with probabilities $1/4$, $1/4$, and $1/2$. The photometric transforms scale brightness and contrast by factors in $[0.7, 1.3]$ and saturation by a factor in $[0.75, 1.25]$, shift hue by up to 0.04, apply a gamma in $[0.8, 1.25]$, and change exposure by up to ± 0.25 stops. The additive noise is Gaussian with standard deviation 0.015 for pixel values in $[0, 1]$, and the same noise field is applied to all frames. With probability 0.25, T also crops the clip to a side-length fraction in $[0.92, 1.0]$ and rotates it by up to $\pm 8°$. On LIBERO, with probability 0.5, T further applies one optical corruption drawn from isotropic blur ($\sigma \in [0.4, 5]$ pixels), directional blur (length in $[3, 21]$ pixels), zoom blur (maximum zoom in $[1.02, 1.2]$), and fog (strength in $[0.3, 2.5]$). The teacher always reads the unperturbed clip.

Alignment. Table 6 lists the alignment hyperparameters, which are shared by the three backbones. The projection head g_ψ is a two-layer MLP with a GELU activation that maps the action-stream width to a hidden width of 512 and back.

Table 6: **Alignment hyperparameters** shared by Fast-WAM, LingBot-VA, and OpenWAM-α. Layers are indexed from 0.

Hyperparameter	Value
Aligned stream	action
Student layer m / teacher layer n	8 / 20
Loss weight λ	0.05
Noise-level gap δ	$\mathcal{U}[0, 0.2)$
Projection head	Linear–GELU–Linear, hidden width 512
EMA decay	0.9999 (fp32)

Training. *LIBERO.* Fast-WAM builds on Wan2.2-TI2V-5B with 30 MoT layers and reads a third-person and a wrist camera at 224×224. The three variants are trained for 10 epochs on the four LIBERO suites with a learning rate of 10^{-4}, a cosine schedule, and a global batch size of 128, and are evaluated at the same checkpoint. They differ only in T and λ. *RoboTwin 2.0.* LingBot-VA is trained on the 50 clean demonstrations of each task for 30,000 steps with a global batch size of 16 episodes. *Real robot.* OpenWAM-α and PICA-WAM are fine-tuned from the pretrained OpenWAM-α checkpoint for 2,000 steps with a global batch size of 128, a learning rate of 10^{-5}, AdamW ($\beta = (0.9, 0.95)$, weight decay 0.01), and a cosine schedule with 5% warmup, on 33-frame clips with a stride of 4 from the three cameras. $\pi_{0.5}$ is fine-tuned from the $\pi_{0.5}$ base checkpoint for 2,000 steps with a batch size of 128, a peak learning rate of 5×10^{-5} with cosine decay, and an action horizon of 32, using the image augmentation of its official implementation.

Inference. On LIBERO and LIBERO-Plus, Fast-WAM runs in its joint mode: the video stream takes a single Euler step from $\sigma = 0.9998$ to $\sigma = 0.953$, and the action stream then runs 10 denoising steps while attending to the video features at that noise level. The diagnostics in Section 4.2 use the same procedure. On RoboTwin 2.0 and the real robot, we use the default inference settings of each backbone.

B Diagnostic Protocol

Paired views. We replay LIBERO demonstrations in the LIBERO-Plus base environment and record the full simulator state at five control steps per episode. For the lighting, background, and camera families, we load the corresponding LIBERO-Plus scene, which differs from the base scene only in its scene description or camera pose, and set it to the recorded state; we verify that the resulting simulator state and end-effector pose match the nominal ones. We use one variant per LIBERO-Plus difficulty level for each of these families. For the sensor-noise families, we apply the

Table 7: **Video prediction fidelity** on 96 LIBERO training clips, averaged over the two cameras, with 95% bootstrap confidence intervals. The perturbed view is constructed with the training distribution T.

	Nominal view		Perturbed view	
Method	**PSNR ↑**	**SSIM ↑**	**PSNR ↑**	**SSIM ↑**
Baseline	24.62	0.864	19.82	0.734
+Aug	22.38	0.829	24.14	0.813
PICA-WAM	22.44	0.828	24.12	0.813

LIBERO-Plus corruption functions to the nominal render at five severities. All views of a state share the same noise seed.

Feature sensitivity. For each layer, we compute the relative ℓ_2 distance between the perturbed and the nominal features, and divide it by the same distance between the nominal features of the state and those of a different state in the same episode, at least eight control steps apart. The video features are the hidden states from which each layer computes its video keys and values, restricted to the observed-frame tokens. The action features are the action-token inputs of each layer at the first denoising step, where the action input is the same noise for both views. We pool distances over views and report 95% bootstrap confidence intervals over states.

Layer-band intervention. We partition the 30 video layers into the bands 0–4, 5–9, 10–14, 15–20, 21–25, and 26–29. For each band, we replace the video hidden states of the nominal view with those of the perturbed view, recompute the keys and values of these layers, and denoise the action again with the same noise. We report the resulting action deviation from the nominal action, normalized by that of the Baseline on the same pair of views, and compare PICA-WAM with +Aug by a paired Wilcoxon test.

Trajectory drift. Each predicted 32-step action chunk is executed open-loop from the recorded state in the base environment, which the perturbations do not change physically. The drift is the distance between the end-effector positions reached from the nominal and the perturbed view. PICA-WAM reduces the drift of +Aug on every family ($p < 10^{-8}$, paired Wilcoxon test). Figure 4 visualizes the executed trajectories for five states.

Video prediction fidelity. We draw 96 training clips, construct their nominal and perturbed views with T, run the full video denoising schedule, and compare the predicted future frames with the ground truth of the same view. Table 7 shows that PICA-WAM and +Aug have the same fidelity on both views ($p > 0.1$). Both are below the Baseline on nominal views and above it on perturbed views.

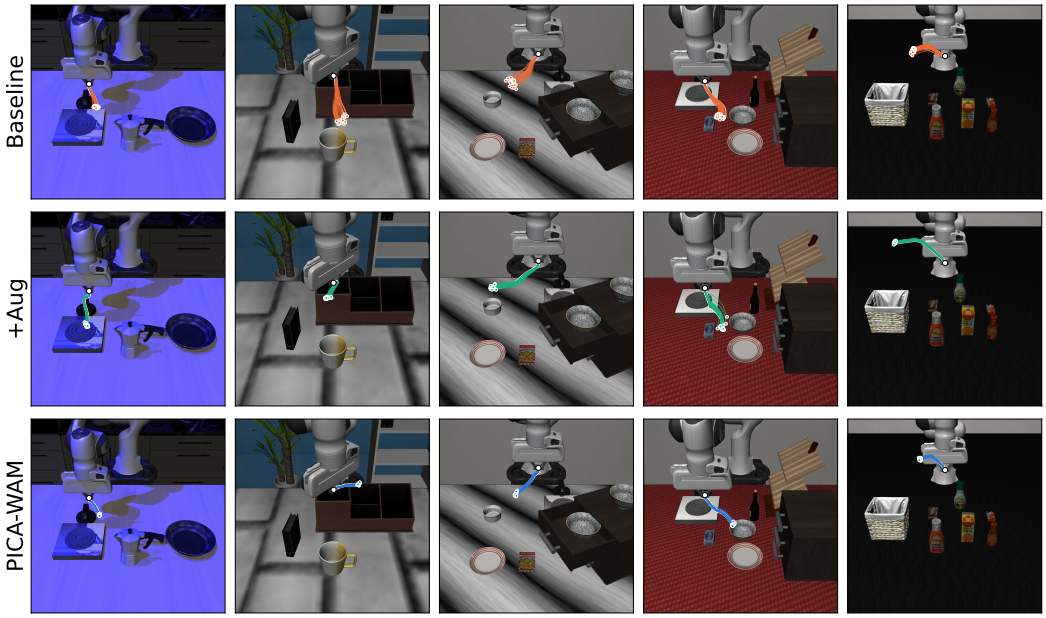

Figure 4: **Executed trajectories under visual perturbations.** Each column shows one LIBERO-Plus state under a perturbed view. Each curve is the end-effector path obtained by executing the 32-step action chunk predicted from a different perturbed view of the same state with the same noise seed; the white dot marks the start position. A tighter fan indicates more consistent actions across appearances. The first three columns are the states with the largest baseline drift; the last two are drawn at random.