RAC-WAM: REGION-WISE ADVERSARIAL CONSISTENCY FOR VISUAL ROBUSTNESS IN WORLD-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models (WAMs) jointly predict robot actions and future visual states, with future prediction providing supervision beyond action labels. Yet their performance still degrades notably under visual distribution shifts such as changes in camera viewpoint or scene appearance. Adversarial consistency training offers an effective way to improve robustness, but existing methods typically use action-only consistency and perform global perturbation search over the full observation. These choices are not fully aligned with WAMs: action-only consistency may miss vulnerabilities exposed by the future-video branch, while global search may repeatedly exploit the most sensitive evidence, leaving other semantic regions underexplored. To address these issues, we propose RAC-WAM, a region-wise adversarial consistency framework for robust WAMs. RAC-WAM jointly leverages action and future-video discrepancies to guide adversarial search. To broaden perturbation coverage across camera views and semantic regions, RAC-WAM performs hierarchical region-wise search by sampling a camera scope and constructing separate perturbations for robot, object, and background regions. To account for regional sensitivity differences under a conserved perturbation budget, perturbation magnitudes are allocated across regions according to local sensitivity. Experiments on LIBERO-Plus, RoboTwin 2.0, and a Franka robot demonstrate the effectiveness of RAC-WAM: it improves zero-shot success on LIBERO-Plus from 72.2% to 82.8% over a same-backbone baseline while preserving in-distribution performance, and delivers consistent robustness gains in bimanual and real-world manipulation.

1 INTRODUCTION

Vision-language-action (VLA) models directly map visual observations and language instructions to robot actions, and have become a prominent paradigm for generalist robot policies (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2025b). World action models (WAMs) further require the policy to predict future visual states alongside robot actions, providing additional supervision beyond action labels through temporal dynamics, object motion, and interaction processes (Kim et al., 2026; Ye et al., 2026; Li et al., 2026a). Existing work has shown that the benefits of such joint training largely stem from the representations shaped during training, while omitting future-video generation at inference incurs little performance loss (Yuan et al., 2026).

Although representations shaped by future visual prediction provide WAMs with a certain degree of visual robustness, existing WAMs still suffer substantial performance degradation when camera viewpoints or scene appearance change (Zhang et al., 2026). To further improve the robustness of WAMs, adversarial consistency training provides an effective approach: it uses the model's own gradients to search for worst-case perturbations that maximally change its predictions, and then constrains the predictions under perturbed inputs to remain consistent with those under clean inputs (Miyato et al., 2019; Xie et al., 2020). Rather than enumerating real-world visual variations or hand-specifying perturbation forms, adversarial training adapts perturbations to the directions along which the model is most vulnerable.

However, as illustrated in Figure 1(a), existing adversarial methods typically measure consistency only on action predictions and perform global perturbation search over the full observation. These

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

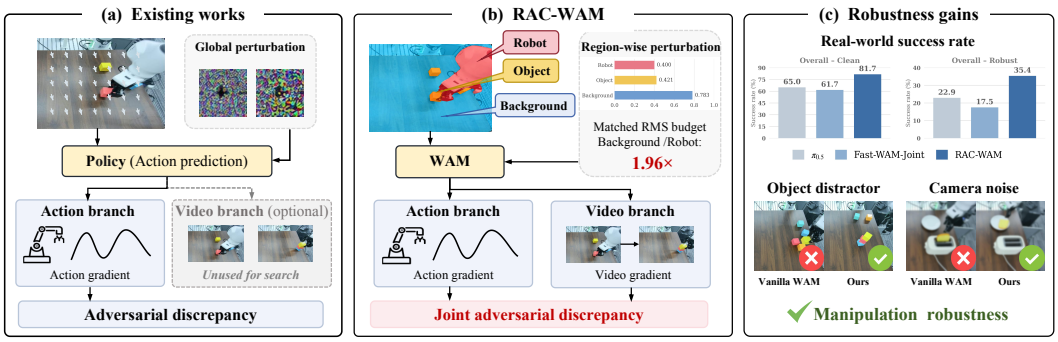


Figure 1: **Existing adversarial training vs. RAC-WAM.** Existing adversarial training typically relies on global perturbation search driven by action predictions. RAC-WAM instead combines joint action–video discrepancy with region-wise adversarial search, improving real-world robustness under visual distribution shifts.

choices are not fully aligned with WAMs. First, the action and future-video branches share a visual representation but may respond differently to the same perturbation; action-only consistency may therefore miss vulnerabilities exposed by the future-video branch. Second, our perturbation analysis reveals substantial sensitivity differences across semantic regions. As illustrated in Figure 1(b), under a matched perturbation budget, background perturbations induce much larger changes in predicted actions than robot- or object-region perturbations; across LIBERO suites, this difference reaches 1.6–2.3×. This suggests that the model may over-rely on background information that should be only weakly related to task decisions. A global adversarial search may therefore repeatedly exploit the most sensitive visual evidence, leaving other semantic regions insufficiently explored. Robustness training for WAMs should therefore exploit both action and future-video responses while explicitly accounting for regional differences in sensitivity.

To address these issues, we propose RAC-WAM, a region-wise adversarial consistency framework for robust WAMs. First, RAC-WAM jointly leverages action and future-video discrepancies under perturbations to the shared visual representation to guide adversarial search. Second, RAC-WAM performs hierarchical region-wise search by first sampling a camera scope and then constructing separate perturbations for robot, object, and background regions within that scope. A conserved perturbation budget is allocated across active regions according to local sensitivity. For robot-region perturbations, only future-video consistency is retained. Together, these designs tailor adversarial training to both the dual-branch prediction structure and the observed variation in regional sensitivity, providing a more targeted robustness objective under visual distribution shifts. All robustness-specific components operate only during training, with no change to the inference pipeline.

Our contributions are summarized as follows:

- We introduce a dual-branch adversarial consistency objective for WAMs that jointly exploits action and future-video discrepancies to guide adversarial search.
- We develop region-wise adversarial consistency that broadens perturbation coverage across camera scopes and semantic regions, allocates perturbation magnitudes according to local sensitivity, and adapts the action–video consistency objective to region semantics.
- RAC-WAM improves zero-shot success on LIBERO-Plus from 72.2% to 82.8% over a same-backbone WAM baseline while preserving in-distribution performance, and consistently improves robustness in bimanual and real-world manipulation settings (Figure 1(c)).

2 Related Work

World Action Models. Latent world models learn predictive dynamics for control through imagined rollouts or online planning (Hafner et al., 2020; Hansen et al., 2022). WAMs extend this predictive formulation to imitation learning by jointly modeling robot actions and future visual states (Wu et al., 2024; Cen et al., 2025). Recent work develops this formulation through more efficient future prediction (Yuan et al., 2026), task-progress modeling and cross-modal interaction (Chen et al., 2026a), and explicit structural priors. Such priors include geometric or semantic supervision

(Ma et al., 2026; Lou et al., 2026; Yu et al., 2026), object-centric representations (Liu et al., 2026), and 3D geometric grounding (Zhao et al., 2026).

Robust Policy Learning. Visual distribution shifts can substantially degrade the generalization of VLA and WAM policies (Fei et al., 2026; Zhang et al., 2026). Data augmentation and domain randomization improve robustness through predefined visual transformations, while clean-observation targets can stabilize learning under augmented inputs (Laskin et al., 2020; Kostrikov et al., 2021; Tobin et al., 2017; Hansen et al., 2021). Adversarial training provides a complementary approach by optimizing perturbations along directions that maximally disrupt model predictions (Miyato et al., 2019; Xie et al., 2020). Recent work adapts this paradigm to embodied policies through feature filtering (Fu et al., 2026), consistency regularization across perturbation types (Guo et al., 2026), multimodal alignment (Luo et al., 2026), and decoupled robustness objectives (Xie et al., 2026). Related studies further reveal robustness failures in WAMs under adversarial inputs (Li et al., 2026b; Yan et al., 2025). Existing adversarial defenses are largely action-centric and apply global visual perturbations, whereas RAC-WAM jointly regularizes action and video predictions with region-specific adversarial search.

3 METHOD

3.1 PRELIMINARIES: WAMS

A language-conditioned visuomotor policy maps visual observations, proprioceptive states, and task instructions to a distribution over action chunks. Specifically, at control step k, let o_k denote the current visual observation, p_k the proprioceptive state, ℓ the language instruction, and A_k an H-step action chunk. The policy models $p_\theta(A_k \mid o_k, p_k, \ell)$. WAMs augment this formulation with future visual prediction, jointly learning how the robot should act and how the observed scene may evolve under interaction (Wu et al., 2024; Cen et al., 2025; Yuan et al., 2026).

For concreteness, we formulate WAM training using Fast-WAM-Joint, in which future-video and action tokens attend to each other and are denoised jointly within a shared model (Yuan et al., 2026). Our robustness formulation only assumes that the two branches condition on a shared visual representation and is otherwise independent of these specific architectural components.

Let z_k^o denote the clean latent representation of the current observation and z_k^+ the latent sequence of subsequent visual observations. The action and future-video targets are

$$y_k^{\mathrm{act}} = A_k, \qquad y_k^{\mathrm{vid}} = z_k^+. \tag{1}$$

Both branches are trained with flow matching. For each $q \in \mathcal{Q} = \{\mathrm{act}, \mathrm{vid}\}$, we independently sample a flow time $t^q \sim \mathcal{U}(0,1)$ and Gaussian noise $\epsilon^q \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The flow-matching interpolant and its target velocity are given by

$$y_{k,t^q}^q = (1-t^q) y_k^q + t^q \epsilon^q, \qquad u_k^q = \epsilon^q - y_k^q. \tag{2}$$

For compactness, we collect the non-visual conditioning variables as $\zeta_k = \Big(\ell, p_k, \{y_{k,t^q}^q, t^q\}_{q\in\mathcal{Q}}\Big)$. Given the shared visual context z_k^o, branch q predicts the flow velocity $\hat{u}_{\theta,k}^q = f_\theta^q(z_k^o, \zeta_k)$. The standard WAM training objective is

$$\mathcal{L}_{\mathrm{joint}} = \sum_{q\in\mathcal{Q}} \mathbb{E}_{k,\,\{t^q,\epsilon^q\}_{q\in\mathcal{Q}}} \left[\omega^q(t^q) \left\| \hat{u}_{\theta,k}^q - u_k^q \right\|_{\Omega^q}^2 \right], \tag{3}$$

where ω^q is the flow-schedule weight and $\|x\|_\Omega = \big(\sum_i \Omega_i x_i^2 / \sum_i \Omega_i\big)^{1/2}$ denotes the masked root-mean-squared magnitude over valid entries. Here, Ω^{act} and Ω^{vid} are validity masks that exclude padded action and future-video latent entries, respectively.

3.2 RAC-WAM

We propose RAC-WAM, a region-wise adversarial consistency framework for improving the visual robustness of WAMs. The overview of RAC-WAM is illustrated in Figure 2. We first measure perturbation-induced discrepancies in the action and future-video branches relative to their clean

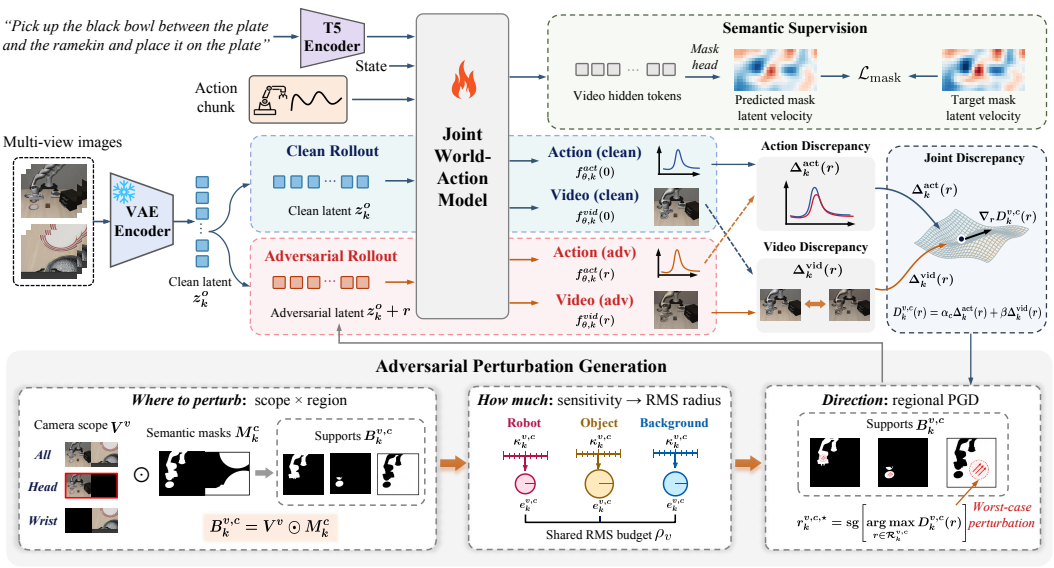


Figure 2: **Overview of RAC-WAM.** RAC-WAM jointly leverages action and future-video discrepancies to guide hierarchical region-wise adversarial search across camera scopes and semantic regions, with perturbation magnitudes allocated according to local sensitivity and region-specific perturbations optimized by PGD.

predictions under matched conditioning. We then construct perturbations hierarchically by sampling a camera scope and defining separate robot, object, and background regions within it using offline semantic masks. A conserved perturbation budget is allocated across active regions according to their sensitivity, and a separate adversarial perturbation is optimized for each region. For the robot region, we retain only future-video consistency. All robustness-specific components are used during training, leaving the inference pipeline unchanged.

3.2.1 Dual-Branch Perturbation Discrepancies

Visual perturbations can propagate differently through the action and future-video branches of a WAM. Although the two branches share a visual representation, their velocity predictions are defined over different target spaces: the action-chunk space and the future-visual-latent space. We therefore characterize the perturbation response within each branch by comparing its perturbed prediction with its own clean prediction. Let r denote a perturbation applied to the clean observation latent z_k^o. For each branch $q \in \mathcal{Q}$, we define

$$f_{\theta,k}^q(r) := f_\theta^q(z_k^o + r, \zeta_k), \qquad f_{\theta,k}^q(0) = f_\theta^q(z_k^o, \zeta_k) = \hat{u}_{\theta,k}^q. \tag{4}$$

The clean and perturbed evaluations use the same conditioning variables ζ_k, including the language instruction, proprioceptive state, noised branch targets, and flow times. Consequently, their only difference is the observation latent supplied as visual context.

We measure the response of branch q by

$$\Delta_k^q(r) = \left\| f_{\theta,k}^q(r) - \text{sg}\left[f_{\theta,k}^q(0)\right] \right\|_{\Omega^q}^2, \qquad q \in \mathcal{Q}, \tag{5}$$

where $\text{sg}[\cdot]$ denotes stop-gradient, so that the clean prediction serves as a fixed reference. Under the matched conditioning above, $\Delta_k^q(r)$ isolates the change in branch q's predicted flow velocity induced by the visual perturbation r. We use $\Delta_k^{\text{act}}(r)$ as the primary control-related signal, while $\Delta_k^{\text{vid}}(r)$ provides a complementary signal from the shared visual representation.

3.2.2 Region-Wise Adversarial Consistency

Hierarchical perturbation coverage. A single adversarial search over the full observation may repeatedly exploit the visual evidence that most readily increases the prediction discrepancy, leaving

other camera views or semantic regions insufficiently explored. We broaden perturbation coverage hierarchically: at each training instance, we first sample a camera scope and then construct a separate perturbation for every non-empty semantic region within that scope.

Specifically, the camera scope is indexed by $v \in \mathcal{V} = \{\text{all}, \text{head}, \text{wrist}\}$, corresponding respectively to the full multi-view observation, the head-camera view, and the wrist-camera view. At each training instance, we sample one scope v from $\mathcal{V}$. We denote the associated binary mask on the observation-latent grid by V^v, with $V^{\text{all}} = \mathbf{1}$ and V^{head} and V^{wrist} selecting the corresponding camera views. We additionally partition the observation according to its semantic content. Robot and object masks are obtained from simulator annotations in simulation and generated offline with Qwen3-VL (Bai et al., 2025) and SAM 3 (Carion et al., 2026) for physical-robot data. Further details of the mask construction pipeline are provided in Appendix A.2. Together with a background mask covering the remaining visual content, they define three semantic regions on the observation-latent grid, denoted by M_k^c for $c \in \mathcal{C} = \{\text{rob}, \text{obj}, \text{bg}\}$. These masks are used only during training. Combining the camera scope with the semantic partition, we define the mask of region c within scope v as

$$B_k^{v,c} = V^v \odot M_k^c, \tag{6}$$

where $\odot$ denotes element-wise multiplication. Because a semantic region may be absent from a particular camera scope, we define the set of active regions as

$$\mathcal{C}_k^v = \left\{ c \in \mathcal{C} : |B_k^{v,c}|_1 > 0 \right\}. \tag{7}$$

For each sampled camera scope v, we construct a separate perturbation for every $c \in \mathcal{C}_k^v$.

Joint action–video consistency. The preceding construction determines the camera scope and semantic region in which each perturbation is applied. We next define how the action and future-video responses are combined for each such perturbation. For every active region $c \in \mathcal{C}_k^v$, we define

$$D_k^{v,c}(r) = \alpha_c \Delta_k^{\text{act}}(r) + \beta \Delta_k^{\text{vid}}(r), \tag{8}$$

where β is the fixed weight of the future-video discrepancy and α_c determines whether action consistency is imposed in region c. Specifically, we set

$$\alpha_c = \begin{cases} 0, & c = \text{rob}, \\ 1, & c \in \{\text{obj}, \text{bg}\}. \end{cases} \tag{9}$$

Perturbations concentrated on the robot region may interfere with visual evidence of the robot configuration, making an unchanged action prediction an inappropriate consistency target. We therefore remove the action discrepancy from the robot-region objective while retaining the future-video discrepancy. In contrast, the object and background objectives use both branch-wise discrepancies. Importantly, $\alpha_{\text{rob}} = 0$ does not exclude the robot region from adversarial training: a separate robot-region perturbation is still constructed and optimized through $\Delta_k^{\text{vid}}(r)$.

Sensitivity-guided magnitude allocation. We next allocate perturbation magnitudes according to the local sensitivity of each region-conditioned consistency objective. For every active region $c \in \mathcal{C}_k^v$, we evaluate the objective at a non-zero random initialization $r_k^{v,c,(0)}$ restricted to $B_k^{v,c}$, thereby avoiding the degenerate zero gradient at $r = \mathbf{0}$. We define the sensitivity score as

$$\kappa_k^{v,c} = \left\| B_k^{v,c} \odot S\left(\nabla_r D_k^{v,c}\left(r_k^{v,c,(0)} \right) \right) \right\|_{B_k^{v,c}}, \tag{10}$$

where S is a fixed spatial low-pass filter.

Using $\kappa_k^{v,c}$,we assign each active region a root-mean-square (RMS) perturbation radius $e_k^{v,c}$ using an area-weighted normalization. Let $a_k^{v,c} = N^{-1} \sum_{i=1}^N B_{k,i}^{v,c}$ denote the fraction of the full observation-latent grid occupied by the region, where N is the total number of spatial positions. Given a perturbation scale ρ_v associated with camera scope v, we set

$$e_k^{v,c} = \rho_v \frac{(\kappa_k^{v,c})^\psi}{\left[\sum_{c' \in \mathcal{C}_k^v} a_k^{v,c'} \left(\kappa_k^{v,c'} \right)^{2\psi} \right]^{1/2}}, \tag{11}$$

where $\psi \geq 0$ controls how strongly the allocation favors sensitive regions. This normalization gives $\sum_{c \in \mathcal{C}_k^v} a_k^{v,c} (e_k^{v,c})^2 = \rho_v^2$.

Region-wise adversarial search. Using the allocated RMS radius $e_k^{v,c}$, we define the feasible perturbation set for each active region as

$$\mathcal{R}_k^{v,c} = \left\{ r \;\middle|\; (\mathbf{1} - B_k^{v,c}) \odot r = \mathbf{0}, \quad \|r\|_{B_k^{v,c}} \le e_k^{v,c} \right\}, \tag{12}$$

where $B_k^{v,c}$ is broadcast over the latent channels. We then separately search each active region for the perturbation that maximizes its consistency objective:

$$r_k^{v,c,\star} = \mathrm{sg}\left[\arg\max_{r \in \mathcal{R}_k^{v,c}} D_k^{v,c}(r) \right]. \tag{13}$$

We approximate the inner maximization using projected gradient ascent (PGD) (Madry et al., 2018) under the region-specific constraint. Although their magnitudes are allocated jointly, the regional perturbations are optimized separately and evaluated in distinct perturbed forward passes rather than combined into a single multi-region perturbation.

We aggregate the resulting regional discrepancies into the adversarial consistency loss

$$\mathcal{L}_{\mathrm{adv}} = \mathbb{E}_{k,\, \{t^q, \epsilon^q\}_{q \in \mathcal{Q}},\, v} \left[\frac{1}{|\mathcal{C}_k^v|} \sum_{c \in \mathcal{C}_k^v} D_k^{v,c}\left(r_k^{v,c,\star}\right) \right]. \tag{14}$$

Thus, camera scopes are varied across training instances, while every active semantic region within the sampled scope contributes its own adversarial consistency term.

3.3 Training Objective and Deployment

The adversarial objective above augments the standard WAM training objective. Since the semantic masks M_k^c are already available during training to define the perturbation regions, we further reuse them as a lightweight auxiliary target that encourages the shared representation to retain the future robot–object layout. Specifically, we form a three-channel future-mask sequence containing the object, robot–object union, and robot regions, and encode it with the same frozen variational autoencoder (VAE) as the future video to obtain a latent z_k^{mask} aligned with z_k^+. A lightweight head reads the video tokens from the clean-observation forward pass and predicts the corresponding flow velocity. Reusing the flow time, noise, weighting, and validity mask of the future-video branch, we define

$$u_k^{\mathrm{mask}} = \epsilon^{\mathrm{vid}} - z_k^{\mathrm{mask}}, \qquad \mathcal{L}_{\mathrm{mask}} = \mathbb{E}_{k,,t^{\mathrm{vid}},,\epsilon^{\mathrm{vid}}} \left[\omega^{\mathrm{vid}}(t^{\mathrm{vid}}) \left| \hat{u}_{\theta,\phi,k}^{\mathrm{mask}} - u_k^{\mathrm{mask}} \right|_{\Omega^{\mathrm{vid}}}^2 \right], \tag{15}$$

where $\hat{u}_{\theta,\phi,k}^{\mathrm{mask}}$ denotes the mask-head prediction. The auxiliary head does not enter the adversarial discrepancy $D_k^{v,c}$.

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{joint}} + \lambda_{\mathrm{mask}} \mathcal{L}_{\mathrm{mask}} + \lambda_{\mathrm{adv}} \mathcal{L}_{\mathrm{adv}}, \tag{16}$$

where $\mathcal{L}_{\mathrm{joint}}$ is the standard WAM objective in Eq. 3. We first optimize $\mathcal{L}_{\mathrm{joint}}$ and $\mathcal{L}_{\mathrm{mask}}$, and subsequently activate $\mathcal{L}_{\mathrm{adv}}$ with a linear warm-up of λ_{adv}. This training procedure requires no additional robot demonstrations. All additional components are confined to training. At deployment, the semantic masks, mask-prediction head, and adversarial search are removed. The policy follows the unmodified inference path of the base model, and future-video latents are never decoded; RAC-WAM therefore incurs no additional inference cost relative to the base model.

4 Experiments

4.1 Experimental Setup

Benchmarks and baselines. Our evaluation spans LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2026), RoboTwin 2.0 (Chen et al., 2026b), and a physical Franka robot. LIBERO measures in-distribution performance across four suites with 100 trials per task, while LIBERO-Plus evaluates zero-shot robustness under seven visual shifts. On RoboTwin 2.0, we follow the six-task MaskWAM protocol (Yu et al., 2026) with 500 randomized episodes per task. Real-robot experiments cover

Table 1: **LIBERO and LIBERO-Plus results.** Success rate (%); LIBERO-Plus is evaluated zero-shot after LIBERO training. **PT** denotes embodied pretraining beyond LIBERO demonstrations. Best and second-best results are **bolded** and underlined.

Method	PT	LIBERO					LIBERO-Plus							
		Spatial	Object	Goal	Long	Avg.	Cam.	Rob.	Lang.	Light	BG	Noise	Layout	Avg.
Vision-Language-Action Models														
UniVLA (Wang et al., 2026)	✓	96.5	96.8	95.6	92.0	95.2	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
OpenVLA-OFT (Kim et al., 2025)	✓	97.6	98.4	97.9	94.5	97.1	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
RIPT-VLA (Tan et al., 2025)	✓	99.0	98.6	**98.6**	93.8	97.5	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
X-VLA (Zheng et al., 2026)	✓	98.2	98.6	97.8	97.6	98.1	23.4	**89.7**	75.7	88.2	**96.0**	62.7	71.8	71.4
AVA-VLA (Xiao et al., 2026)	✓	99.2	99.6	97.9	96.2	98.2	**69.4**	34.9	81.5	97.5	94.1	79.1	78.3	74.7
π_0 (Black et al., 2025b)	✓	96.8	98.8	95.8	85.2	94.2	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
HoloBrain-0-GD (Lin et al., 2026)	✓	97.8	98.2	95.2	95.6	96.7	65.5	58.2	78.7	88.1	90.3	66.9	79.5	74.0
World-Action / World-Model-Augmented Models														
Mantis (Yang et al., 2026)	✓	98.8	99.2	94.4	94.2	96.7	15.7	41.8	45.9	45.1	28.9	39.2	62.5	39.8
GE-Act (Liao et al., 2026)	✓	98.2	97.6	95.8	94.4	96.5	60.7	77.0	77.4	95.8	86.0	**90.9**	80.2	80.3
VLA-JEPA (Sun et al., 2026)	✓	96.2	99.6	97.2	95.8	97.2	63.3	67.1	85.4	95.6	93.6	66.3	**85.1**	79.5
WorldVLA (Cen et al., 2025)	✗	87.6	96.2	83.4	60.0	81.8	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
Fast-WAM (Yuan et al., 2026)	✗	98.2	**100.0**	97.0	95.2	97.6	16.4	44.5	68.9	78.2	53.7	37.7	60.7	50.0
SG-WAM (Zhao et al., 2026)	✗	**99.4**	99.8	**98.6**	96.2	**98.5**	58.6	48.9	81.4	89.8	86.1	80.7	74.2	73.0
Fast-WAM-Joint (Yuan et al., 2026)	✗	98.0	99.4	97.6	95.6	97.7	49.3	73.7	92.6	95.5	56.5	65.5	75.1	72.2
RAC-WAM (Ours)	✗	97.8	99.6	97.8	**98.0**	98.3	67.2	75.8	**95.6**	**97.6**	83.7	86.0	78.2	**82.8**

Toaster to Plate, Pour Water, and Block Stacking under clean and four perturbed visual conditions, with 20 trials per task–condition pair. See more detail in Appendix A.1. We compare RAC-WAM against representative VLA policies, including OpenVLA-OFT, π_0, $\pi_{0.5}$(Black et al., 2025b;a), and X-VLA, as well as WAMs such as Fast-WAM, Fast-WAM-Joint, and MaskWAM.

Implementation details. We build RAC-WAM on the pretrained Wan2.2-5B backbone following the Fast-WAM-Joint architecture (Yuan et al., 2026), and use Fast-WAM-Joint as the direct same-backbone baseline. We train with AdamW using a learning rate of 1×10^{-4}. For region-wise adversarial training, we sample the camera scope $v \in \mathrm{all}, \mathrm{head}, \mathrm{wrist}$ with probabilities $(0.5, 0.25, 0.25)$. We set $\beta = 0.2$, $\alpha_{\mathrm{rob}} = 0$, and $\alpha_{\mathrm{obj}} = \alpha_{\mathrm{bg}} = 1$. Adversarial perturbations are optimized with $K = 3$ projected-gradient ascent steps using a base latent RMS budget $\rho = 0.22$ and sensitivity exponent $\psi = 0.5$. See Appendix A.6 for more details.

4.2 Main Results

Performance on LIBERO and LIBERO-Plus. Table 1 compares in-distribution performance on LIBERO and zero-shot robustness on LIBERO-Plus. On standard LIBERO, RAC-WAM achieves 98.3% average success, compared with 97.7% for the direct Fast-WAM-Joint baseline, showing that the proposed robustness training preserves in-distribution performance. On LIBERO-Plus, RAC-WAM improves the average success rate from 72.2% to 82.8%, yielding a 10.6-point gain over Fast-WAM-Joint. It also achieves the highest overall average among the compared methods, surpassing the strongest prior result of 80.3% by 2.5 points. The largest gains over Fast-WAM-Joint occur under camera-viewpoint, background, and sensor-noise shifts, with improvements of 17.9, 27.2, and 20.5 points, respectively. These results show that RAC-WAM substantially improves zero-shot visual robustness while preserving strong in-distribution performance.

Performance on RoboTwin 2.0. We further evaluate RAC-WAM on six randomized RoboTwin 2.0 tasks to assess transfer to bimanual manipulation under diverse environment variations. As shown in Table 2, RAC-WAM achieves an average success rate

Table 2: **RoboTwin 2.0 results.** Success rate (%) on six bimanual tasks; best and second-best results are **bolded** and underlined.

Method	Ham.	Bell	Card	Bur.	Stand	Shoe	Avg.
π_0 (Black et al., 2025b)	68	72	81	79	63	74	72.8
Fast-WAM (Yuan et al., 2026)	83	87	92	94	80	90	87.7
MaskWAM (Yu et al., 2026)	88	93	95	97	85	95	92.2
Fast-WAM-Joint (Yuan et al., 2026)	89	**100**	**99**	**99**	91	89	94.5
RAC-WAM (Ours)	**98**	**100**	**99**	98	**94**	**98**	**97.8**

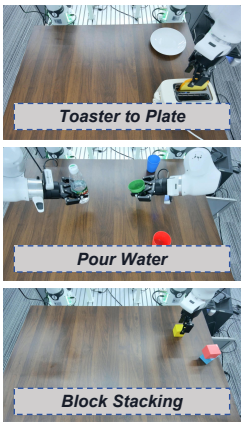


Figure 3: Real-world evaluation setup on Franka.

Table 3: **Real-world robustness evaluation on Franka.** Success rate (%) over 20 trials per task–condition pair. **Robust Avg.** is averaged over the four visual perturbations. Best results are **bolded**.

Task	Method	Clean	Visual Perturbations				Robust Avg.
			Noise	Background	Light	Distractor	
Toaster to Plate	$\pi_{0.5}$	85.0	**50.0**	55.0	5.0	**90.0**	50.0
	Fast-WAM-Joint	**95.0**	20.0	85.0	0.0	50.0	38.8
	RAC-WAM	**95.0**	40.0	**90.0**	**15.0**	70.0	**53.8**
Pour Water	$\pi_{0.5}$	65.0	0.0	0.0	10.0	**35.0**	11.3
	Fast-WAM-Joint	50.0	10.0	20.0	0.0	10.0	10.0
	RAC-WAM	**75.0**	**35.0**	**40.0**	**20.0**	25.0	**30.0**
Block Stacking	$\pi_{0.5}$	45.0	0.0	15.0	5.0	10.0	7.5
	Fast-WAM-Joint	40.0	0.0	0.0	10.0	5.0	3.8
	RAC-WAM	**75.0**	**15.0**	**35.0**	**25.0**	**15.0**	**22.5**
Overall	$\pi_{0.5}$	65.0	16.7	23.3	6.7	**45.0**	22.9
	Fast-WAM-Joint	61.7	10.0	35.0	3.3	21.7	17.5
	RAC-WAM	**81.7**	**30.0**	**55.0**	**20.0**	36.7	**35.4**

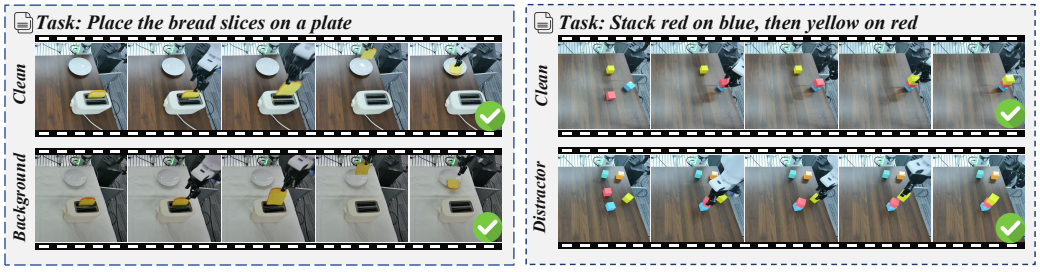


Figure 4: **Examples of real-robot rollouts under visual distribution shifts.**

of 97.8%, compared with 94.5% for Fast-WAM-Joint and 92.2% for MaskWAM. RAC-WAM achieves the strongest performance across nearly all tasks. These results show that the proposed robustness training remains effective in a substantially different bimanual setting with domain-randomized visual conditions.

Real-world evaluation. To assess whether the robustness gains transfer beyond simulation, we evaluate RAC-WAM on three Franka manipulation tasks under four controlled visual shifts (Figure 3, Table 3). Across all tasks and perturbations, RAC-WAM achieves a robust average success rate of 35.4%, compared with 22.9% for $\pi_{0.5}$ and 17.5% for Fast-WAM-Joint. The gains are particularly clear under sensor noise, background changes, and lighting variation. While $\pi_{0.5}$ remains stronger under distractor perturbations on some tasks, RAC-WAM achieves the highest overall robust success rate, demonstrating that the robustness gains transfer to real-world manipulation.

Qualitative real-robot results. Figure 4 shows real-robot rollouts for toaster-to-plate transfer and sequential cube stacking under clean and shifted visual conditions. Despite background changes and visual distractors, the policy maintains the task-relevant manipulation sequence. Additional qualitative visualizations are provided in Appendix A.5.

4.3 Ablation Studies

Contribution of individual components. Table 4 progressively introduces the major components of RAC-WAM under the same WAM backbone. Mask supervision improves the LIBERO-Plus average from 72.2% to 77.5%. Adding joint adversarial consistency further raises the average to 81.1%. Hierarchical region coverage and sensitivity-guided magnitude allocation provide additional gains, reaching 81.9% and 82.8%, respectively. Overall, the complete method improves the controlled WAM baseline by 10.6 percentage points. These results show that the proposed components contribute complementary improvements to robustness.

Table 4: **Component ablation on LIBERO-Plus.** *Mask sup.*, *Joint cons.*, *Region cov.*, and *Sens. alloc.* denote mask supervision, joint adversarial consistency, hierarchical region coverage, and sensitivity-guided allocation, respectively.

Training components				LIBERO-Plus success rate (%)							
Mask sup.	Joint cons.	Region cov.	Sens. alloc.	Cam.	Robot	Lang.	Light	BG	Noise	Layout	Avg.
✗	✗	—	—	49.3	73.7	92.6	95.5	56.5	65.5	75.1	72.2
✓	✗	—	—	59.6	79.9	92.4	95.7	61.3	74.8	79.7	77.5
✓	✓	✗	✗	62.8	72.1	93.5	99.0	84.6	85.4	76.8	81.1
✓	✓	✓	✗	69.0	75.4	92.2	97.3	77.0	87.1	77.8	81.9
✓	✓	✓	✓	67.2	75.8	95.6	97.6	83.7	86.0	78.2	82.8

Table 5: **Action–video discrepancy and region-wise consistency ablation on LIBERO-Plus.** α_{rob} controls action consistency in the robot region. Δ Avg. is relative to the mask-supervision baseline.

Setting	Discrepancy		LIBERO-Plus success rate (%)								Δ Avg.
	Act.	Vid.	Cam.	Robot	Lang.	Light	BG	Noise	Layout	Avg.	
Mask-supervision baseline	✗	✗	59.6	79.9	92.4	95.7	61.3	74.8	79.7	77.5	–
Action only	✓	✗	63.7	74.7	93.0	96.9	80.6	87.9	76.7	81.3	+3.8
Video only	✗	✓	56.6	75.0	92.8	93.7	71.0	80.1	73.8	77.1	-0.4
Joint, $\alpha_{\mathrm{rob}} = 1$	✓	✓	61.3	79.1	94.9	94.0	79.1	86.7	74.0	80.8	+3.3
RAC-WAM, $\alpha_{\mathrm{rob}} = 0$	✓	✓	67.2	75.8	95.6	97.6	83.7	86.0	78.2	82.8	+5.3

Action–video discrepancy and region-wise consistency. Table 5 isolates the roles of the action and future-video discrepancies under the same perturbation configuration. Action-only consistency improves the LIBERO-Plus average from 77.5% to 81.3%, establishing the action branch as the primary robustness signal. Although video-only consistency does not improve the overall average, it yields notable gains under background and sensor-noise shifts, indicating that the future-video branch provides useful complementary supervision. Naively combining both discrepancies across all regions reaches only 80.8%, suggesting that the two signals should not be applied uniformly. With region-dependent consistency, however, their combination reaches 82.8%, outperforming action-only consistency by 1.5 points. These results show that jointly modeling action and future-video discrepancies with region-dependent consistency provides the strongest robustness gains.

Effect of adversarial perturbation search. To verify whether the robustness gains come from model-guided adversarial search rather than perturbation injection alone, Table 6 compares perturbation-generation strategies on LIBERO-Plus. Under matched training settings, random latent perturbations achieve 77.5% average success, whereas PGD-based perturbations reach 82.8%; the WAM baseline and predefined image augmentation achieve 72.2% and 78.1%, respectively. The improvement over random perturbations indicates that searching along model-sensitive directions contributes to robustness beyond perturbation injection alone. See Appendix A.4 for details.

Table 6: **Perturbation-search ablation on LIBERO-Plus.** Average success rate (%).

Training perturbation	Search	Avg.	Δ
WAM baseline	–	72.2	–
Image augmentation	Predefined	78.1	+5.9
Random latent	Random	77.5	+5.3
RAC-WAM (Ours)	PGD (K=3)	**82.8**	**+10.6**

5 CONCLUSION

This work studies adversarial consistency training for improving the visual robustness of WAMs. Our results show that future visual prediction alone does not eliminate sensitivity to visual distribution shifts, while structured regional training can better exploit action consistency with complementary future-video signals. Ablations indicate that semantic supervision, region-wise perturbation coverage, and sensitivity-guided allocation contribute complementary gains. Across simulation, bimanual manipulation, and physical-robot evaluation, RAC-WAM improves robustness without additional inference-time components. Robustness nevertheless remains limited under challenging real-world distractors, highlighting the need for more effective treatment of semantic visual shifts.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We have not used generative AI tools for other tasks requiring disclosure, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to create or edit software code and to edit the paper to improve readability. We have reviewed all AI-assisted work. AI-assisted code edits were manually inspected and tested by the authors, while AI-assisted translations and language edits were checked for technical correctness and revised as needed. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. arXiv preprint, 2025.

Kevin Black, Noah Brown, James Darpinian, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *CoRL*, 2025a.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. In *RSS*, 2025b.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, et al. SAM 3: Segment anything with concepts. In *ICLR*, 2026.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. arXiv preprint, 2025.

Jintao Chen, Peidong Jia, Qingpo Wuwu, et al. MV-WAM: Manifold-aware world action model with value augmentation. arXiv preprint, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *ICML*, 2026b.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *CVPR*, 2026.

Yiyang Fu, Chubin Zhang, Shukai Gong, Yufan Deng, Kaiwei Sun, Qiyang Min, Qibin Hou, Yansong Tang, Jianan Wang, and Daquan Zhou. Stablevla: Towards robust vision-language-action models without extra data. In *ICML*, 2026.

Jianing Guo, Zhenhong Wu, Chang Tu, et al. On robustness of vision-language-action model against multi-modal perturbations. In *ICLR*, 2026.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. In *NeurIPS*, 2021.

Nicklas A. Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *ICML*, 2022.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In *CoRL*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *RSS*, 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, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *ICLR*, 2026.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *ICLR*, 2021.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *NeurIPS*, 2020.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *RSS*, 2026a.

Qi Li, Xingyi Yang, and Xinchao Wang. Badwam: When world-action models dream right but act wrong. arXiv preprint, 2026b.

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. In *ICLR*, 2026.

Xuewu Lin, Tianwei Lin, Yun Du, et al. HoloBrain-0 technical report. arXiv preprint, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023.

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, 2026.

Yunfan Lou, Xiaowei Chi, Xiaojie Zhang, Zezhong Qian, Chengxuan Li, Rongyu Zhang, Yaoxu Lyu, Guoyu Song, Chuyao Fu, Haoxuan Xu, Pengwei Wang, and Shanghang Zhang. Mask world model: Predicting what matters for robust robot policy learning. In *ICML*, 2026.

Jingzhou Luo, Yifan Wen, Yongjie Bai, Xinshuai Song, Yang Liu, and Liang Lin. RoVLA: Multi-consistency constraints for robust vision-language-action models. arXiv preprint, 2026.

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, 2026.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *TPAMI*, 2019.

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. In *ECCV*, 2026.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. arXiv preprint, 2025.

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 *IROS*, 2017.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *ICLR*, 2026.

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 *ICLR*, 2024.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: Improving vision-language-action models with active visual attention. In *CVPR*, 2026.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. In *CVPR*, 2020.

Yuhan Xie, Yuping Yan, Yunqi Zhao, Handing Wang, and Yaochu Jin. STRONG-VLA: Decoupled robustness learning for vision-language-action models under multimodal perturbations. 2026.

Yuping Yan, Yuhan Xie, Yixin Zhang, Lingjuan Lyu, Handing Wang, and Yaochu Jin. When alignment fails: Multimodal adversarial attacks on vision-language-action models. arXiv preprint, 2025.

Yi Yang, Xueqi Li, Yiyang Chen, Jin Song, Yihan Wang, Zipeng Xiao, Jiadi Su, You Qiaoben, Pengfei Liu, and Zhijie Deng. Mantis: A versatile vision-language-action model with disentangled visual foresight. In *CVPR*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, et al. World action models are zero-shot policies. arXiv preprint, 2026.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. MaskWAM: Unifying mask prompting and prediction for world-action models. arXiv preprint, 2026.

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

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, 2026.

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

Jinliang Zheng, Jianxiong Li, Zhihao Wang, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *ICLR*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

A Appendix

A.1 Real-World Robot Deployment

Robot platform. We conduct real-world experiments on a dual-arm Franka manipulation platform, as illustrated in Figure 5. The system consists of two 7-DoF Franka manipulators mounted around a shared tabletop workspace. Each arm is equipped with a parallel gripper and an Intel RealSense D435 RGB-D camera mounted near the wrist, providing close-range observations of the gripper, manipulated objects, and local contact interactions. In addition, a fixed Orbbec Femto Bolt RGB-D camera is positioned in front of the workspace to provide a third-person view of the complete scene. The policy therefore receives three complementary visual observations: the left-wrist view, the right-wrist view, and a fixed third-person view. The two wrist cameras capture fine-grained robot–object interactions, while the third-person camera provides global workspace context and task-progress information.

For robot control, each Franka arm is represented by seven joint commands together with one gripper command. We therefore use a unified 16-dimensional bimanual action space,

$$\mathbf{a}_t = [\mathbf{q}_t^L, g_t^L, \mathbf{q}_t^R, g_t^R] \in \mathbb{R}^{16}, \tag{17}$$

where $\mathbf{q}_t^L, \mathbf{q}_t^R \in \mathbb{R}^7$ denote the joint commands of the left and right arms, respectively, and g_t^L and g_t^R denote their gripper commands. For predominantly single-arm tasks, the non-acting arm remains in a predefined safe configuration, whereas both arms are jointly controlled for bimanual manipulation.



Figure 5: **Real-world dual-arm Franka platform.** The setup consists of two 7-DoF Franka manipulators, two wrist-mounted Intel RealSense D435 RGB-D cameras, and one fixed Orbbec Femto Bolt RGB-D camera providing a third-person view of the workspace.

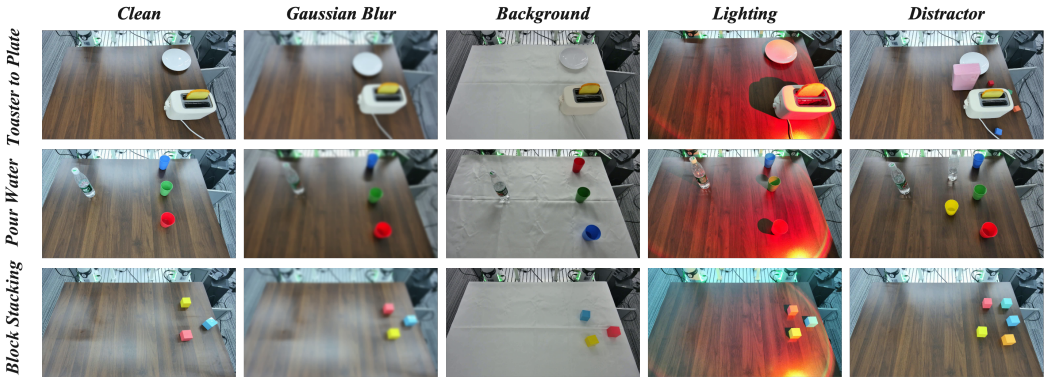


Figure 6: **Real-world tasks and visual perturbation settings.** We evaluate three physical manipulation tasks under the clean condition and four controlled visual perturbations: Gaussian blur, background change, lighting variation, and distractor objects. Each row corresponds to one task—Toaster to Plate, Pour Water, and Block Stacking—and each column corresponds to one visual condition. The task objective and manipulation setup remain unchanged across perturbations.

Real-world tasks. We design three real-world manipulation tasks to evaluate different aspects of robotic control. For each task, we collect 100 teleoperated demonstrations for fine-tuning, resulting in 300 demonstrations in total.

1. **Toaster to Plate:** A single-arm task requiring the robot to grasp a bread slice from a toaster and place it onto a target plate. This task evaluates basic grasping and precise object placement.
2. **Pour Water:** A bimanual task in which one arm manipulates the source cup while the other holds the target cup. This task evaluates dual-arm coordination and precise manipulation during water pouring.
3. **Block Stacking:** A single-arm sequential task requiring the robot to grasp and stack multiple blocks. This task evaluates spatial alignment and long-horizon sequential manipulation.

For robustness evaluation, each task is tested under the clean setting and four controlled visual distribution shifts: sensor noise, background changes, lighting variation, and distractor objects. Each task–condition pair is evaluated over 20 trials.

Visual perturbations. We evaluate each real-world task under the clean setting and four controlled visual perturbations, as illustrated in Fig. 6. **Gaussian Blur** simulates sensor noise by applying Gaussian blur to the visual observations, with the blur parameter set to 10. **Background Change**

Table 7: **Regional sensitivity under matched energy.** Region-only $K=3$ perturbations share the same full-canvas latent RMS $\rho_0=0.22$. We report $s^c = \Delta^{\mathrm{act}}(r_c^\star)$ and the corresponding background-to-region ratios.

LIBERO suite	N	Robot s^{rob}	Object s^{obj}	Background s^{bg}	BG/Robot ratio	BG/Object ratio
Spatial	250	0.477	0.485	0.920	1.929	1.898
Object	250	0.377	0.327	0.755	2.006	2.313
Goal	250	0.377	0.431	0.731	1.941	1.697
Long	250	0.368	0.443	0.724	1.966	1.635
Overall	1000	0.400	0.421	0.783	1.959	1.858

replaces the original scene background with a white background. **Lighting Variation** introduces an external lighting disturbance to alter the illumination and appearance of the workspace. **Distractor Objects** adds task-irrelevant objects to the workspace while keeping the task objective unchanged. These perturbations are applied only at evaluation time and are not included in the teleoperated training demonstrations.

A.2 Task-Grounded Mask Construction

We build an offline mask extraction pipeline based on Qwen3-VL and SAM 3. For each episode and camera view, Qwen3-VL jointly processes the task instruction and the first observation frame to identify the manipulated object and the robot, producing short, visually grounded text prompts and object aliases. These prompts are then passed separately to SAM 3 to initialize the object and robot masks. On the prompt frame, we remove masks with implausibly small or large areas and retain the highest-confidence target instance for the manipulated object, while merging valid robot instances to cover the full manipulator. SAM 3 subsequently propagates the selected instance identities across the entire episode, yielding temporally consistent mask trajectories. For views in which the target is difficult to observe in the first frame, the pipeline optionally re-prompts SAM 3 at periodic frames and merges the resulting tracking trajectories. Finally, we resolve overlaps by assigning overlapping pixels to the robot, define the background as the complement of the robot and object regions, and resize all masks to the training resolution. The resulting mutually exclusive robot, object, and background masks are stored as binary trajectories for model training.

A.3 Regional Sensitivity Analysis

We probe the frozen WAM baseline on 1000 held-out observations, with 250 from each LIBERO suite. For each region $c \in \{\mathrm{rob}, \mathrm{obj}, \mathrm{bg}\}$, we restrict perturbations to that region and maximize Δ^{act} using $K=3$ projected-ascent steps. All searches use the same full-canvas RMS budget $\rho_0=0.22$ and share the instruction, proprioceptive state, flow time, and flow noise within each observation. As shown in Table 7, background perturbations are consistently the most influential. Overall, background sensitivity is $1.959\times$ that of the robot region and $1.858\times$ that of the object region. This consistent asymmetry motivates nonuniform regional allocation; its downstream benefit is evaluated by the ablations below.

A.4 Additional Ablations

Robustness of future prediction. Figure 7 shows that prediction fidelity decreases as input corruption grows, but RAC-WAM consistently maintains higher PSNR and SSIM and lower LPIPS than the WAM baseline and action-only training. The separation is most pronounced under Gaussian noise. Table 8 quantifiesthese trends. On clean inputs, RAC-WAM improves PSNR from 24.07 to 25.74 and SSIM from 0.8539 to 0.8781 over the baseline, while reducing LPIPS from 0.0665 to 0.0519. At $\sigma = 16$, the baseline, action-only, and RAC-WAM achieve PSNR values of 16.72, 19.82, and 20.82, respectively. Compared with action-only training, RAC-WAM further improves SSIM from 0.6742 to 0.7292 and reduces LPIPS from 0.2832 to 0.2117. These results show an additional world-branch robustness benefit in the full training configuration.

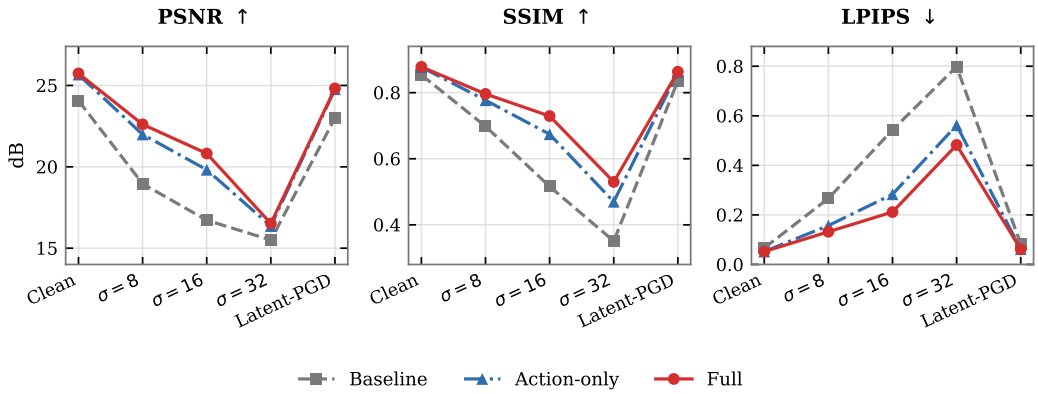


Figure 7: **Future-video prediction fidelity under clean and perturbed inputs.** PSNR, SSIM, and LPIPS are computed against ground-truth future clips over 400 LIBERO samples. We compare the WAM baseline, action-only training, and the full RAC-WAM objective with future consistency.

Table 8: **Future-video prediction fidelity across four LIBERO suites.** Results are averaged over 400 paired clips; bold denotes the best result.

Condition	PSNR (dB) ↑			SSIM ↑			LPIPS ↓		
	Base.	Action-only	Full	Base.	Action-only	Full	Base.	Action-only	Full
Clean	24.07	25.67	**25.74**	0.8539	0.8773	**0.8781**	0.0665	0.0522	**0.0519**
Gaussian $\sigma = 8$	18.96	21.99	**22.61**	0.6981	0.7770	**0.7962**	0.2665	0.1570	**0.1320**
Gaussian $\sigma = 16$	16.72	19.82	**20.82**	0.5161	0.6742	**0.7292**	0.5430	0.2832	**0.2117**
Gaussian $\sigma = 32$	15.50	16.36	**16.52**	0.3508	0.4698	**0.5299**	0.7984	0.5618	**0.4822**
Latent-PGD	23.00	24.77	**24.82**	0.8356	0.8624	**0.8633**	0.0822	0.0618	**0.0615**

Model-guided perturbation search. Table 9 compares different perturbation-generation strategies. Image augmentation improves the WAM baseline from 72.2% to 78.1%. More importantly, random latent perturbations and our method share the same structural supervision, structured supports, sensitivity-guided budget allocation, and perturbation magnitude, differing only in the perturbation direction. Replacing random directions with PGD-based model-guided search increases the average success rate from 77.5% to 82.8%, a gain of 5.3 points. The largest improvements occur under camera, background, and noise shifts, with gains of 12.8, 18.4, and 7.4 points, respectively. These results show that explicitly searching for model-sensitive directions is substantially more effective than sampling random perturbations under the same perturbation budget.

A.5 Additional Qualitative Results

Figure 8 presents six representative LIBERO-Plus tasks under background, robot initial-state, camera, sensor-noise, object-layout, and lighting perturbations. Each panel compares rollouts with the same task ID and perturbation configuration. The baseline rollout in the upper row fails, whereas our rollout in the lower row succeeds. Frames progress from left to right, showing how the two policies execute the same task under the same evaluation setting.

Figure 9 shows selected real-world rollouts under Gaussian blur, distractor objects, and lighting variation. The three tasks are placing bread slices on a plate, stacking colored blocks, and pouring water into a green cup, respectively. In each panel, the upper row shows one failed baseline rollout and the lower row shows one successful rollout of our method. These examples illustrate the challenge posed by visual perturbations through both failed and successful executions.

A.6 Implementation Details

Table 10 lists the architecture, sampling, and optimization hyperparameters used to train RAC-WAM, complementing Section 4.1.

Table 9: **Perturbation-generation ablation on LIBERO-Plus.**

Training perturbation	Search strategy	LIBERO-Plus success rate (%)								Δ Avg.
		Cam.	Robot	Lang.	Light	BG	Noise	Layout	Avg.	
WAM baseline	—	49.3	73.7	92.6	95.5	56.5	65.5	75.1	72.2	—
Image augmentation	Predefined	60.1	67.2	93.0	93.4	73.1	83.6	79.5	78.1	+5.9
Random latent	Random	54.4	78.0	94.1	94.7	65.3	78.6	79.0	77.5	+5.3
RAC-WAM (Ours)	PGD (K=3)	67.2	75.8	95.6	97.6	83.7	86.0	78.2	**82.8**	+10.6

Table 10: **Implementation hyperparameters for RAC-WAM.**

Hyperparameter	Value
Backbone and action expert	
Backbone	Wan2.2-TI2V-5B
Action-expert hidden dimension d_a	1024
Action horizon H	32
Latent structure	
Action steps per sample	32
RGB frames per sample	9, sampled from 33 observation steps at stride 4
Causal VAE temporal factor	4 (1 observed + 2 future latent slices)
Structural supervision	
Semantic regions	Robot, object, background
Structural supervision weight λ_{mask}	1.0
Regional adversarial consistency	
Camera-scope probabilities	$(0.50, 0.25, 0.25)$ for (all views, head view, wrist view)
Action coefficients $\alpha_{\mathrm{robot}}, \alpha_{\mathrm{object}}, \alpha_{\mathrm{bg}}$	$(0, 1, 1)$
Video coefficient β	0.2
Inner ascent steps K	3 per active region
Camera-scope RMS budget	0.22 for each of the three scopes
Regional sensitivity exponent	0.5
Random-start RMS ξ	10^{-3}
Low-frequency gradient smoothing	4×7 grid; mixing coefficient 0.5
Training schedule	
Final consistency weight λ_{adv}	0.5
Consistency warmup	0 for the first 1000 steps; linear rise over the next 3,000 steps
Flow time sampling	$u \sim \mathcal{U}(0, 1), t = 5u/(1 + 4u)$
Training duration	10 epochs
Optimization	
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.95$
Learning rate and weight decay	10^{-4} and 10^{-2}
Learning-rate schedule	Cosine decay with 5% warmup
Per-GPU batch size	2
Gradient clipping	1.0
Hardware	8 × NVIDIA A100 GPUs
Inference	
Denoising schedule	Deterministic shifted flow schedule, 1 step
Classifier-free guidance scale	1.0

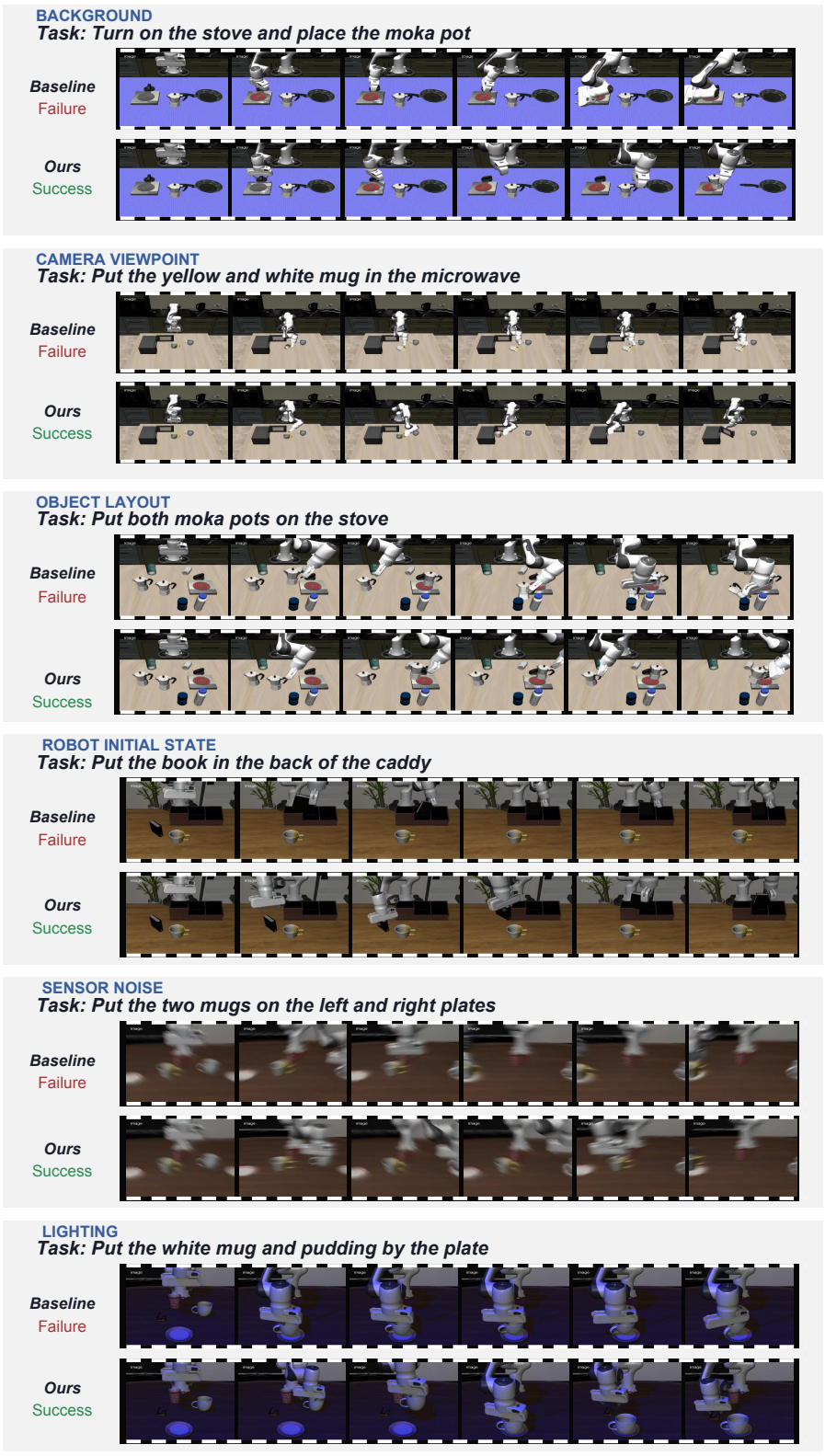


Figure 8: **Representative LIBERO-Plus rollouts under six perturbation types.** The panels show background texture, robot initial state, camera viewpoint, sensor noise, object layout, and lighting shifts. Within each panel, the baseline failure (top) and our successful rollout (bottom) use the same task ID and perturbation configuration. Key execution frames are ordered from left to right.



Figure 9: **Selected real-world rollouts under visual perturbations.** The panels show Gaussian blur, distractor objects, and lighting variation. Each panel contains one failed baseline rollout (top) and one successful rollout of our method (bottom). Frames progress from left to right. These selected examples illustrate execution outcomes rather than aggregate success rates.